Deep and generative modeling for protein-ligand interactions

David Koes

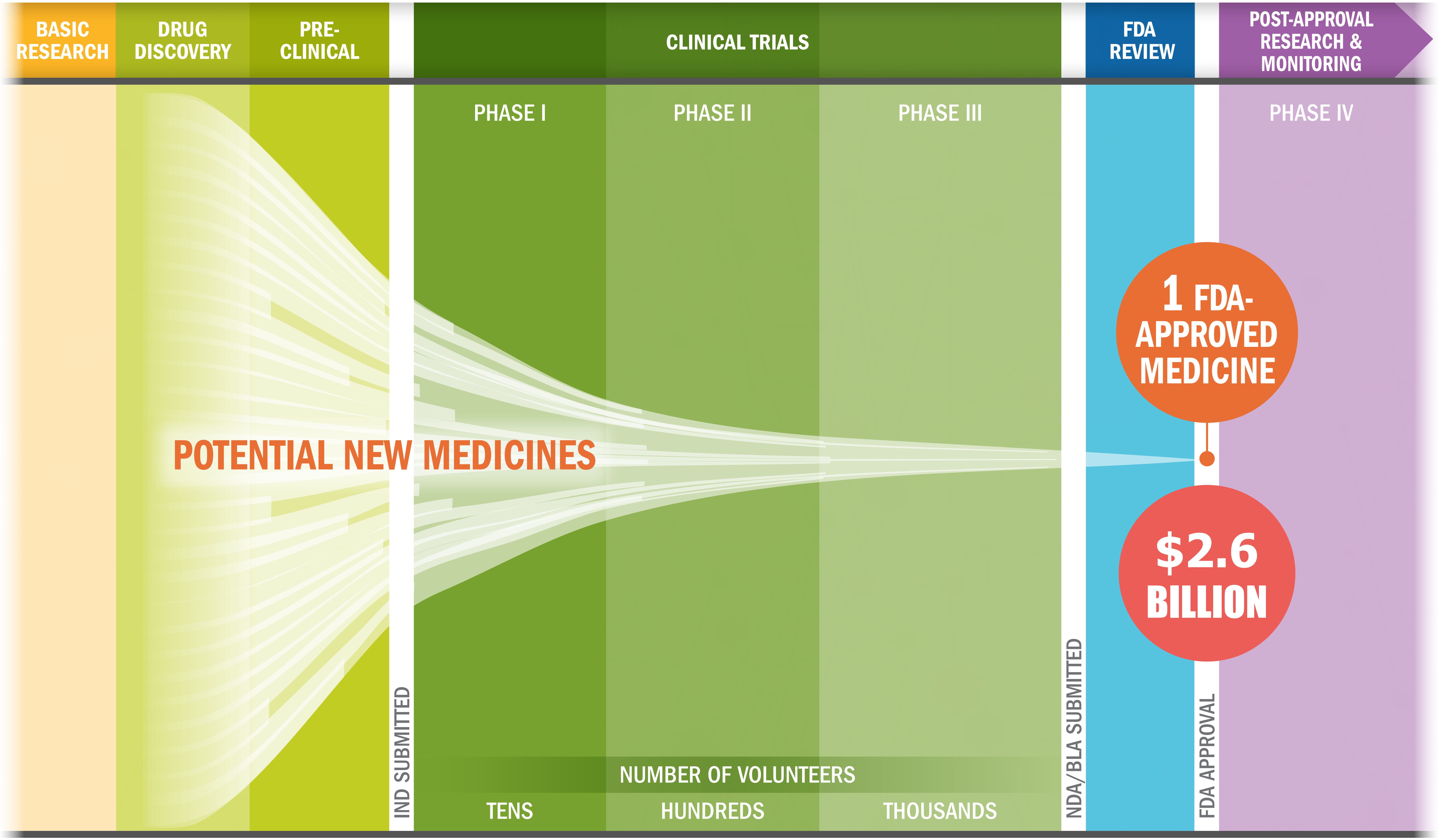
@david_koes

Critical Assessment of Techniques for Protein Structure Prediction

Cancun, Mexico

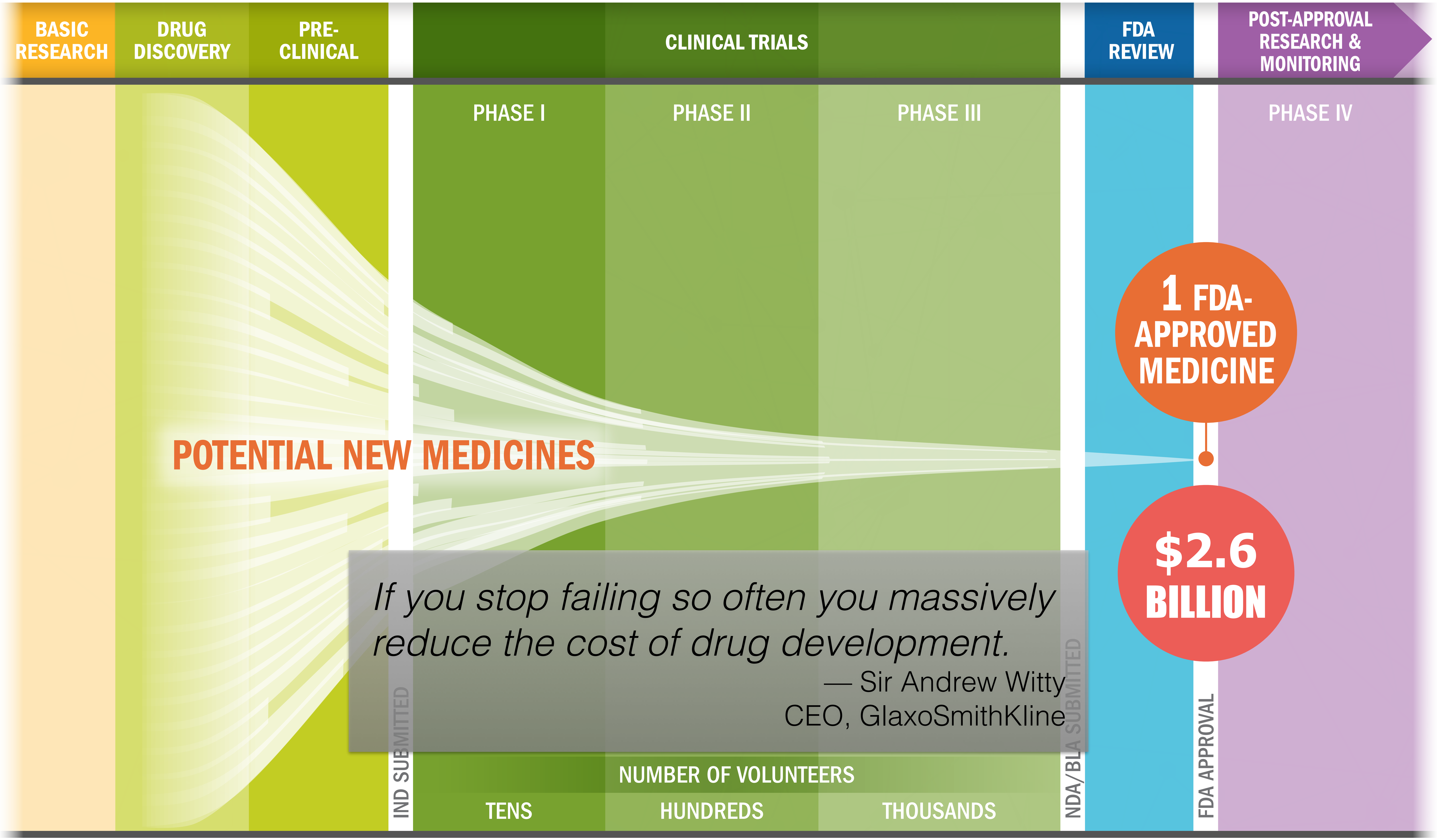
December 2, 2018

THE BIOPHARMACEUTICAL RESEARCH AND DEVELOPMENT PROCESS



Source: Pharmaceutical Research and Manufacturers of America (http://phrma.org)

THE BIOPHARMACEUTICAL RESEARCH AND DEVELOPMENT PROCESS

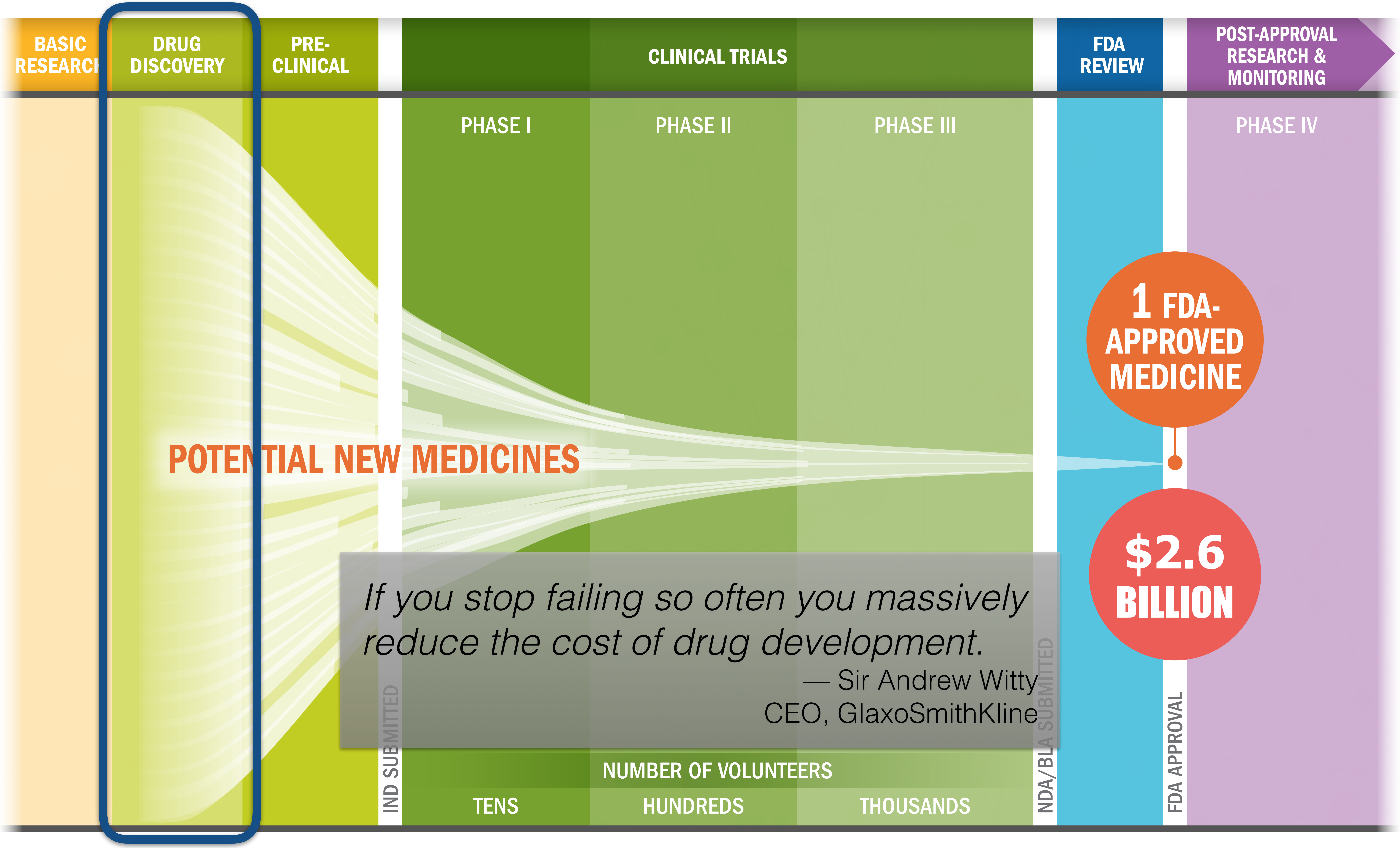


If you stop failing so often you massively reduce the cost of drug development.

— Sir Andrew Witty
CEO, GlaxoSmithKline

Source: Pharmaceutical Research and Manufacturers of America (http://phrma.org)

THE BIOPHARMACEUTICAL RESEARCH AND DEVELOPMENT PROCESS



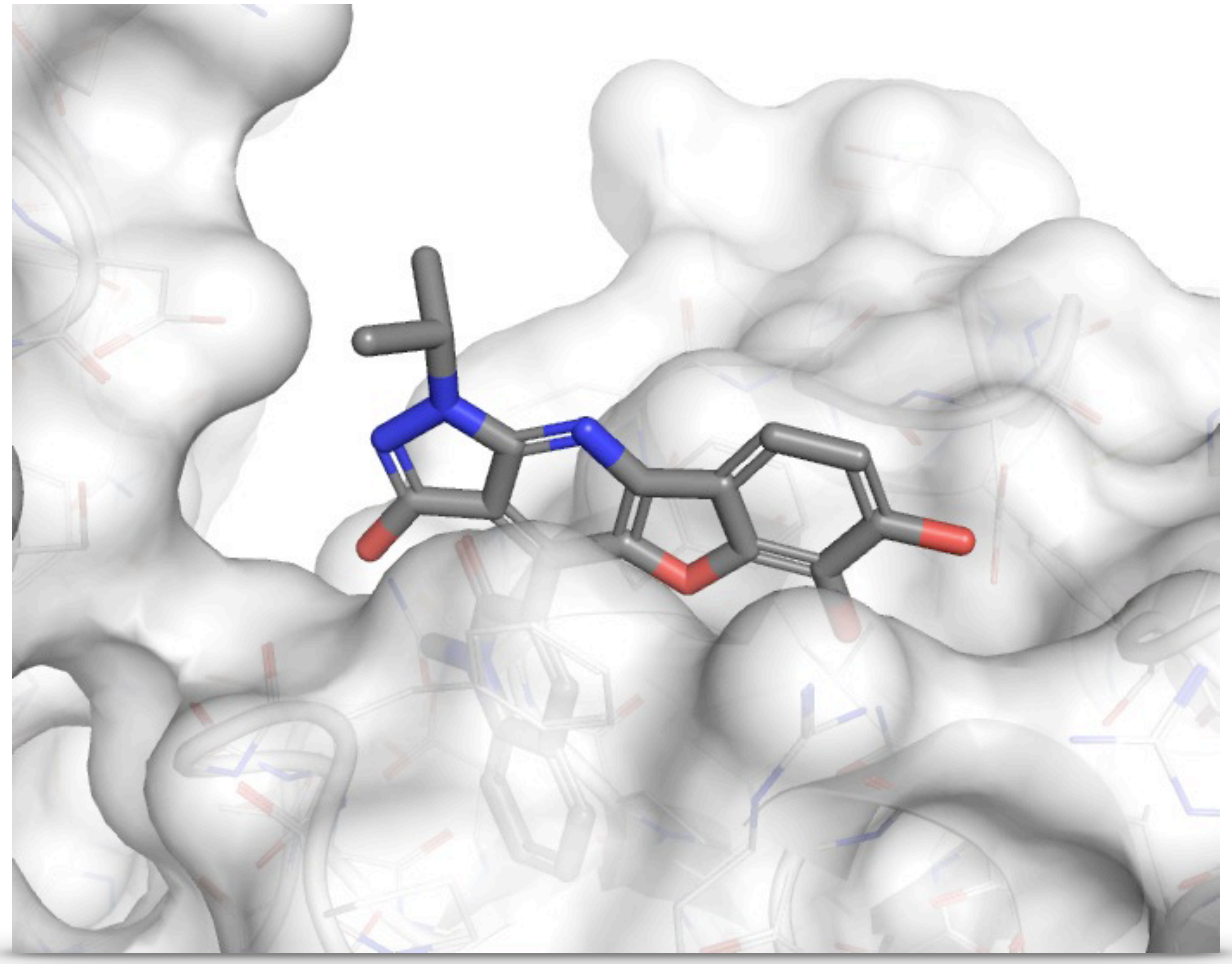
If you stop failing so often you massively reduce the cost of drug development.

— Sir Andrew Witty
CEO, GlaxoSmithKline

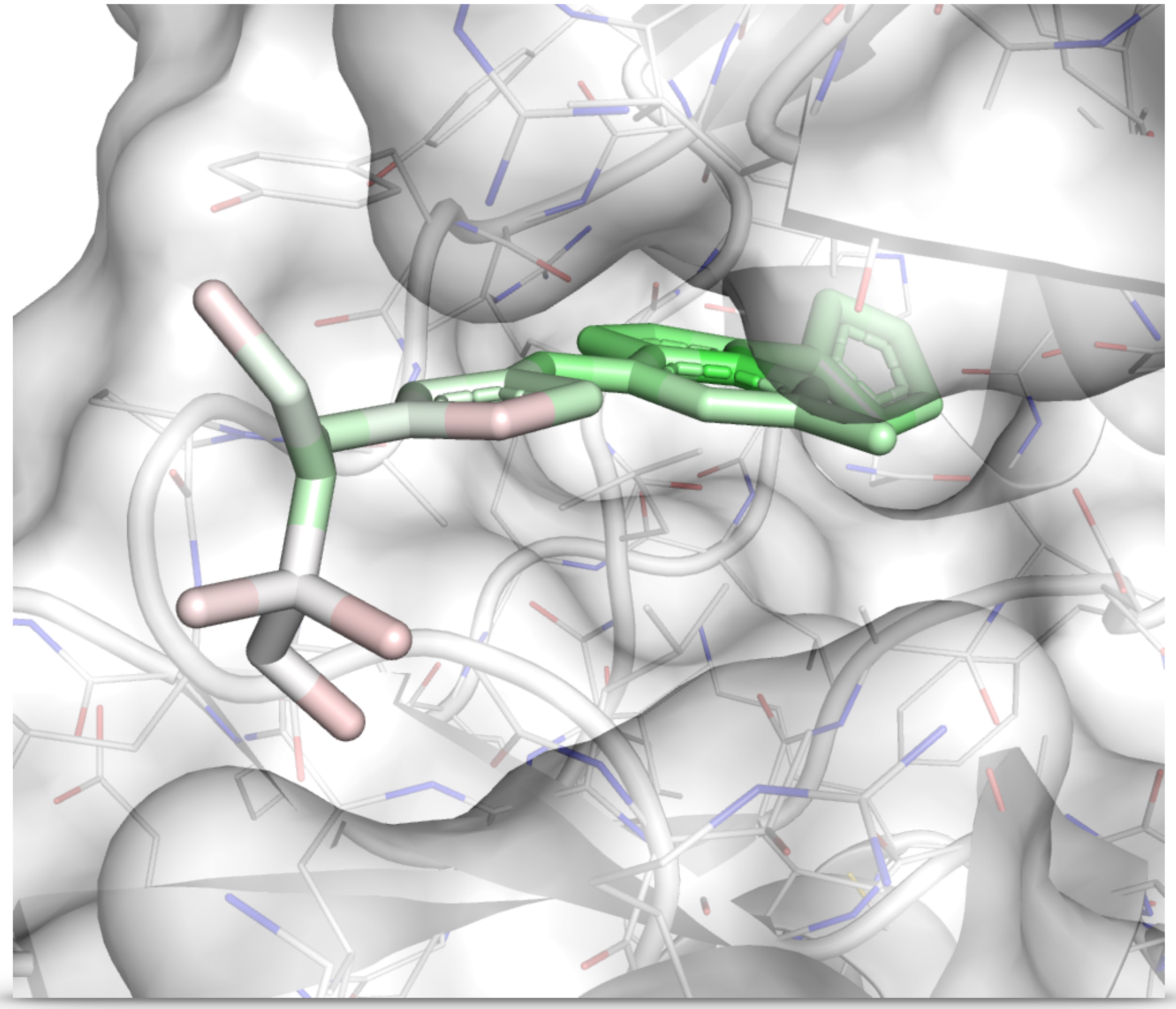
Source: Pharmaceutical Research and Manufacturers of America (http://phrma.org)

Structure Based Drug Design

Pose Prediction → Binding Discrimination → Affinity Prediction



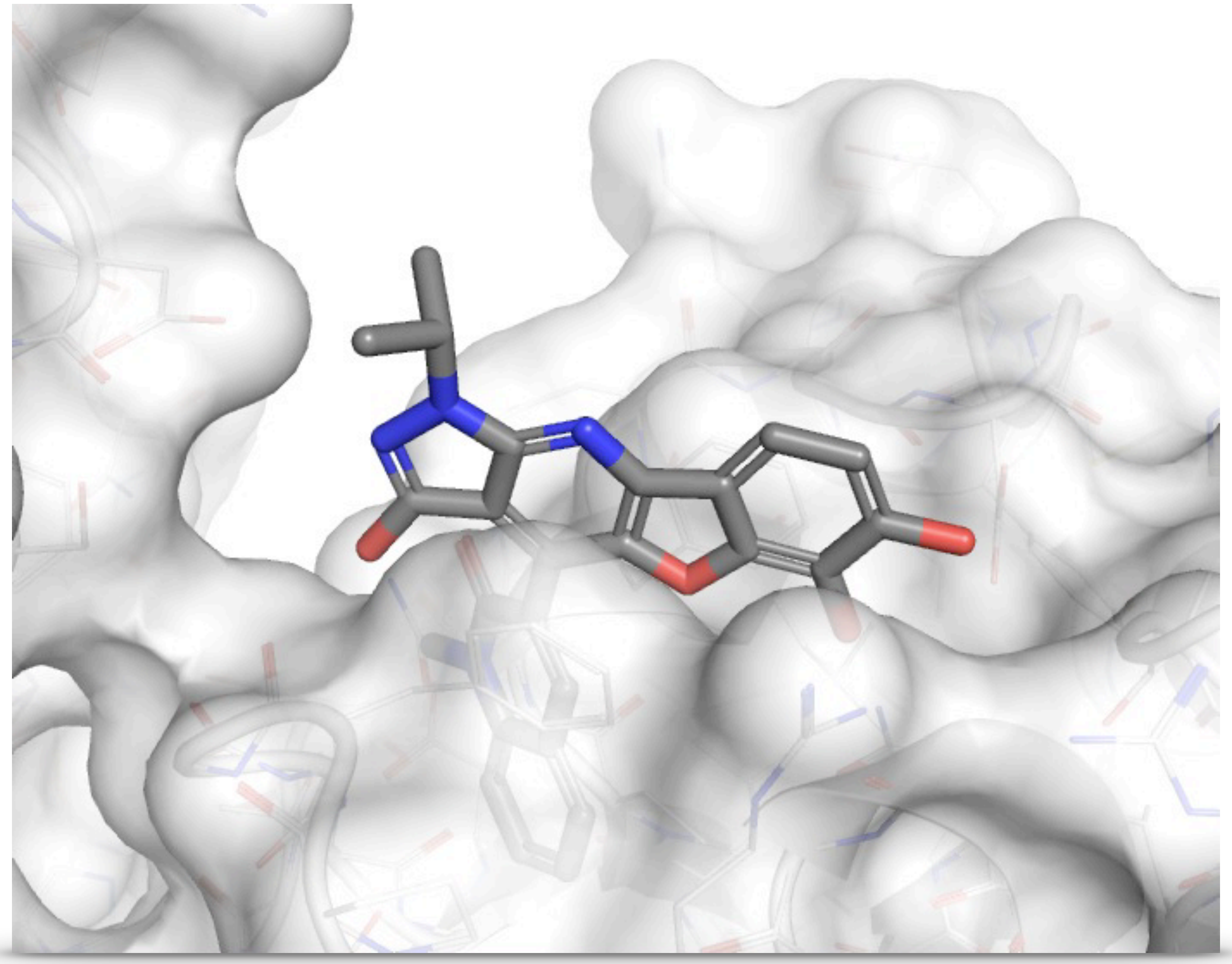
Virtual Screening



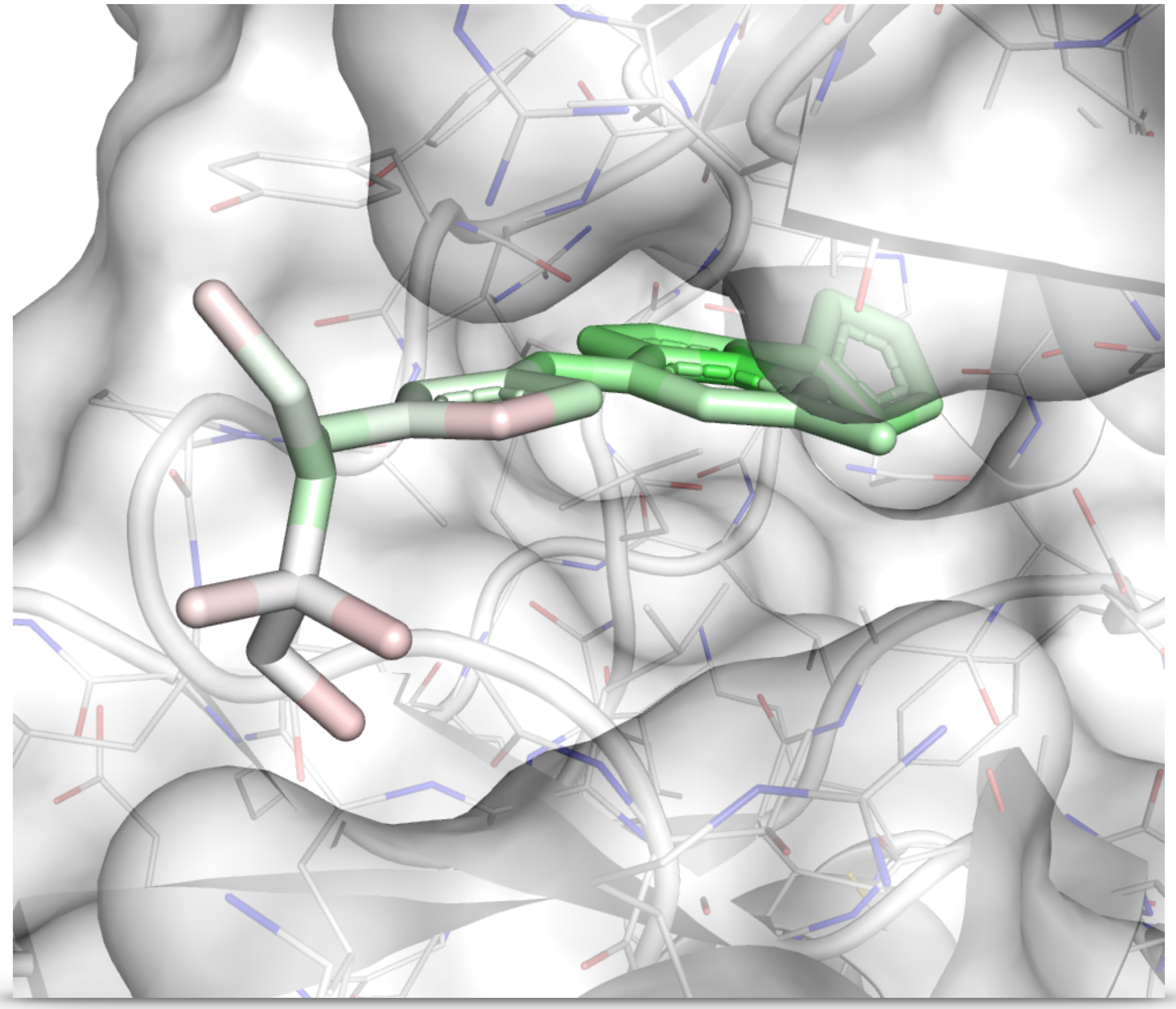
Lead Optimization

Structure Based Drug Design

Pose Prediction → Binding Discrimination → Affinity Prediction

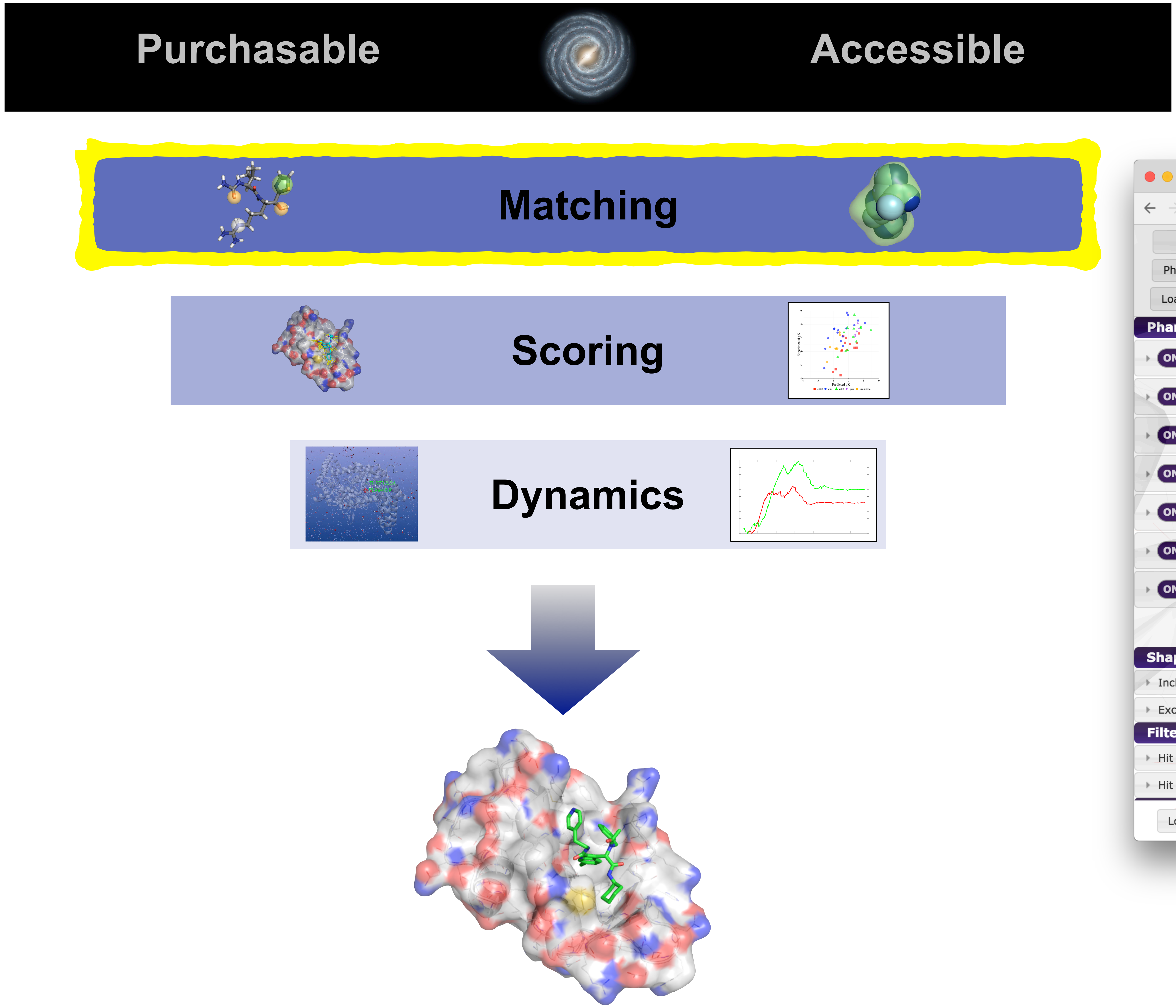


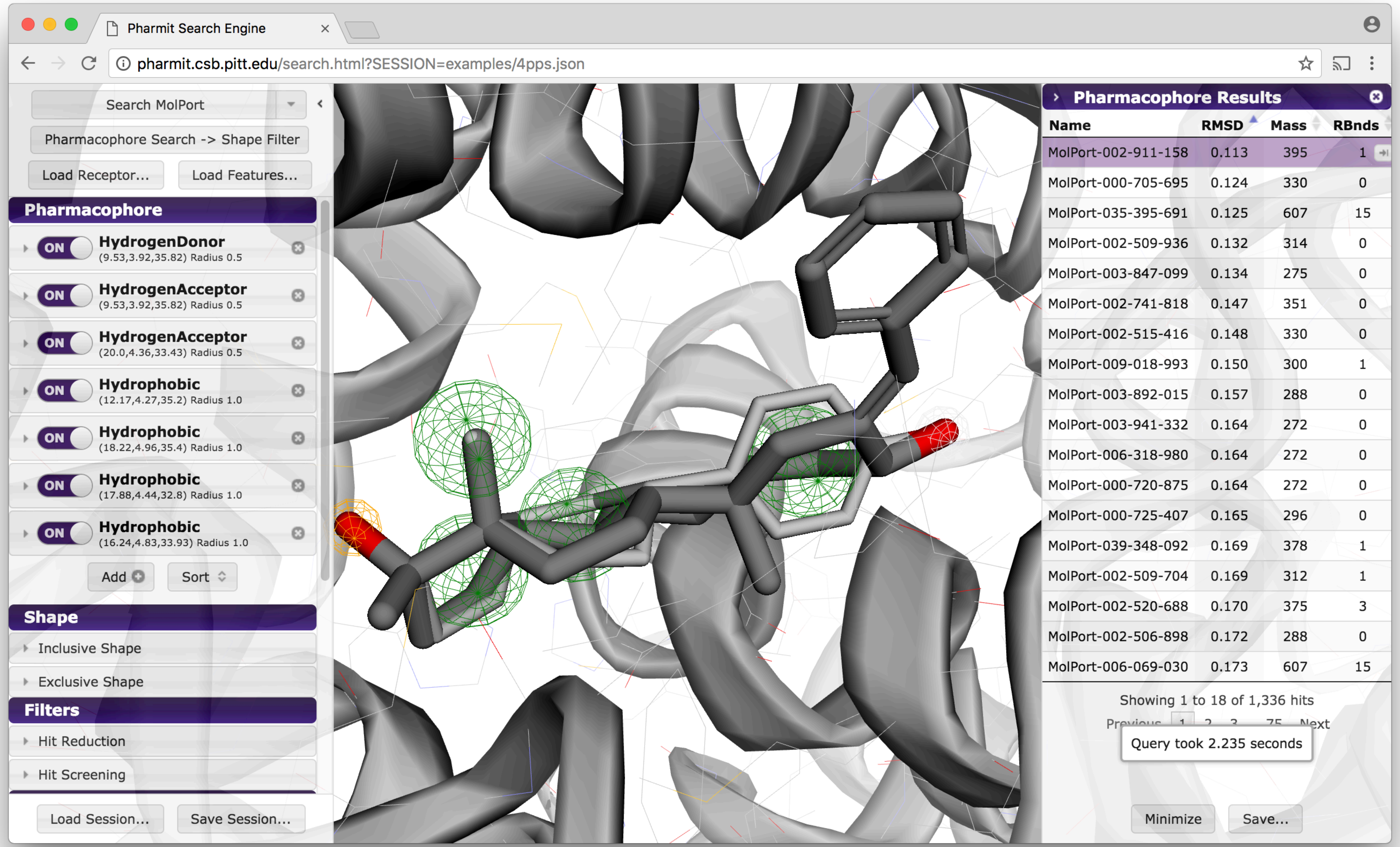
Virtual Screening



Lead Optimization

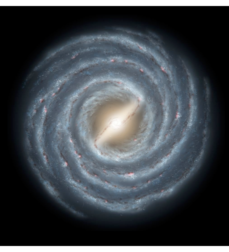
Drug Discovery Funnel

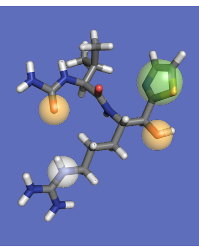
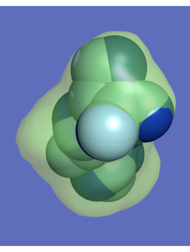


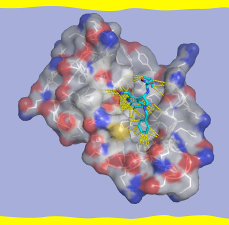
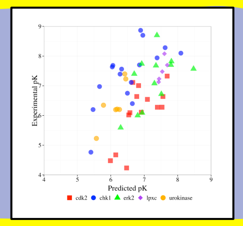


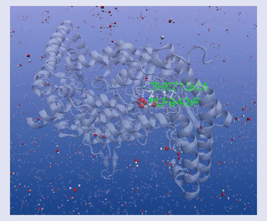
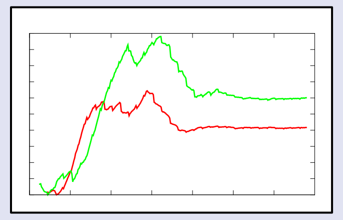
http://pharmit.csb.pitt.edu

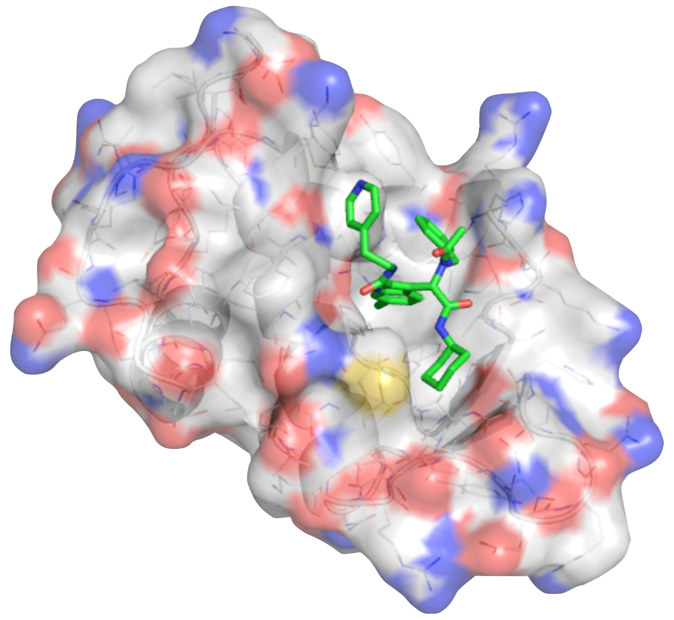
Drug Discovery Funnel

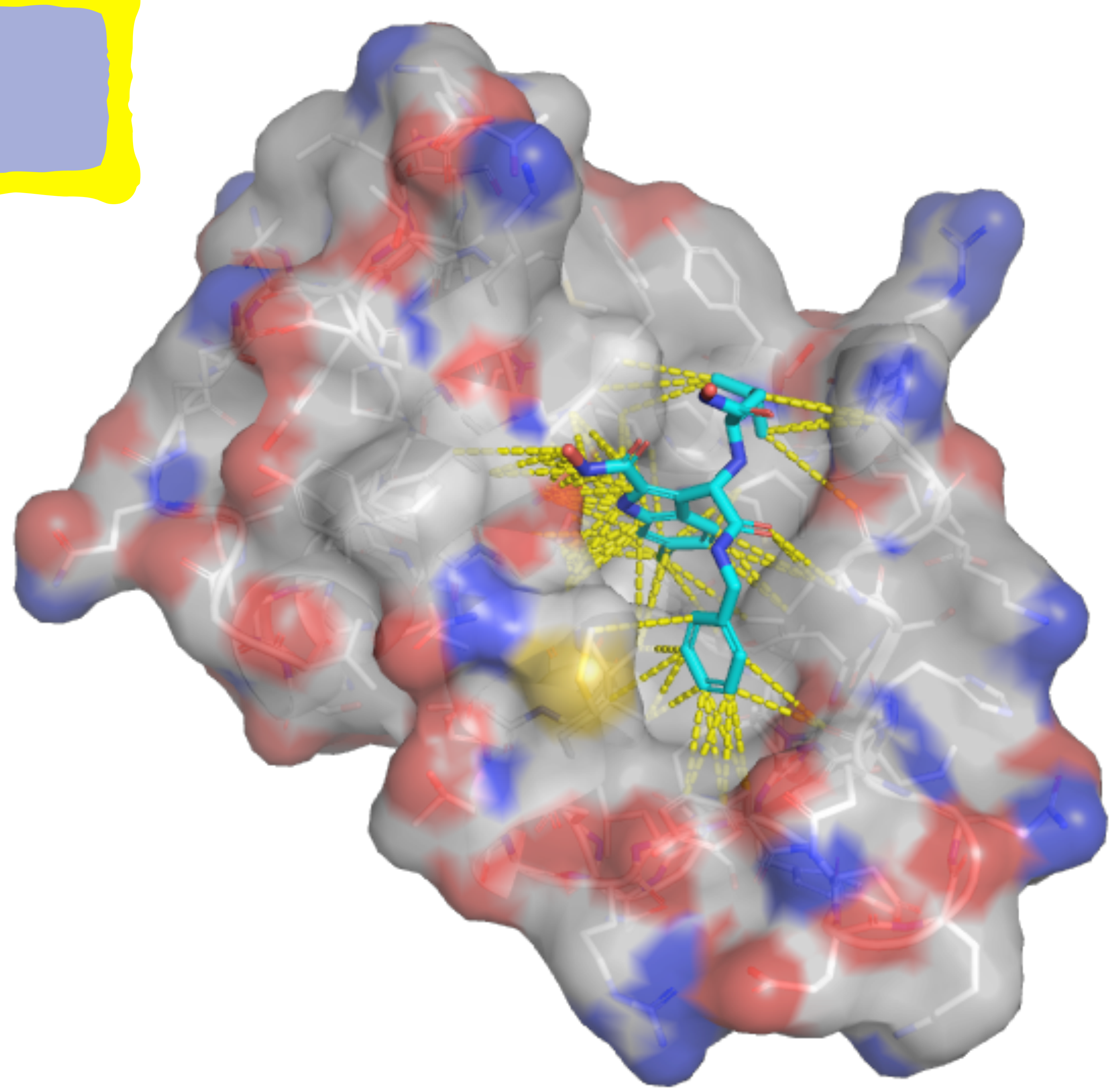
Purchasable  **Accessible**

 **Matching** 

 **Scoring** 

 **Dynamics** 

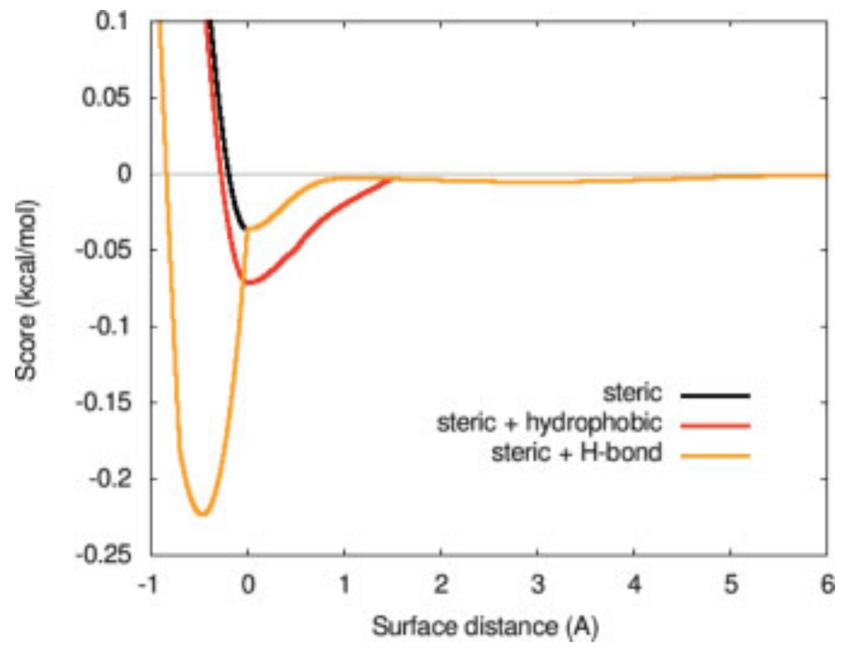




$$\begin{aligned} \mathrm{gauss}_1(d) &= w_{\mathrm{guass}_1} e^{-(d/0.5)^2} \\ \mathrm{gauss}_2(d) &= w_{\mathrm{guass}_2} e^{-((d-3)/2)^2} \\ \mathrm{repulsion}(d) &= \begin{cases} w_{\mathrm{repulsion}} d^2 & d < 0 \\ 0 & d \geq 0 \end{cases} \end{aligned}$$

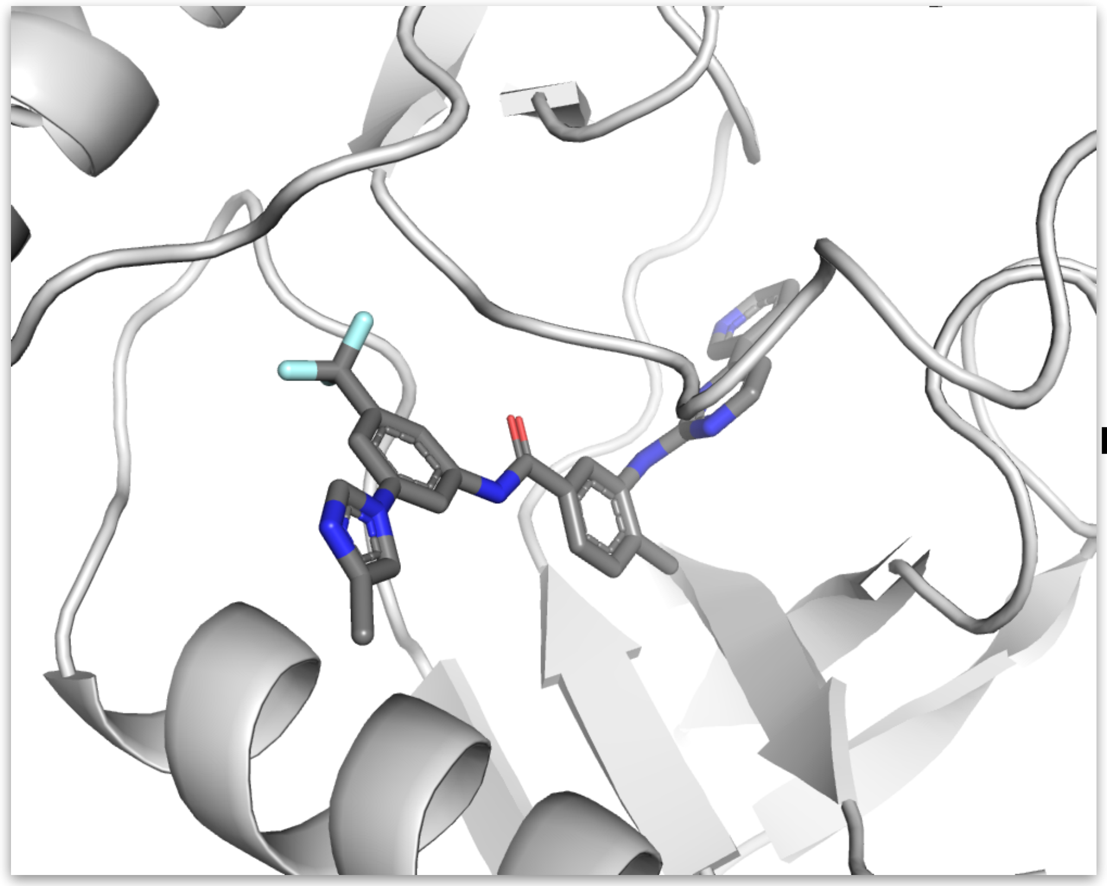
$$\mathrm{hydrophobic}(d) = \begin{cases} w_{\mathrm{hydrophobic}} & d < 0.5 \\ 0 & d > 1.5 \\ w_{\mathrm{hydrophobic}}(1.5 - d) & \mathit{otherwise} \end{cases}$$

$$\mathrm{hbond}(d) = \begin{cases} w_{\mathrm{hbond}} & d < -0.7 \\ 0 & d > 0 \\ w_{\mathrm{hbond}}(-\frac{10}{7} d) & \mathit{otherwise} \end{cases}$$



O. Trott, A. J. Olson, AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading, *Journal of Computational Chemistry* 31 (2010) 455-461

Protein-Ligand Scoring



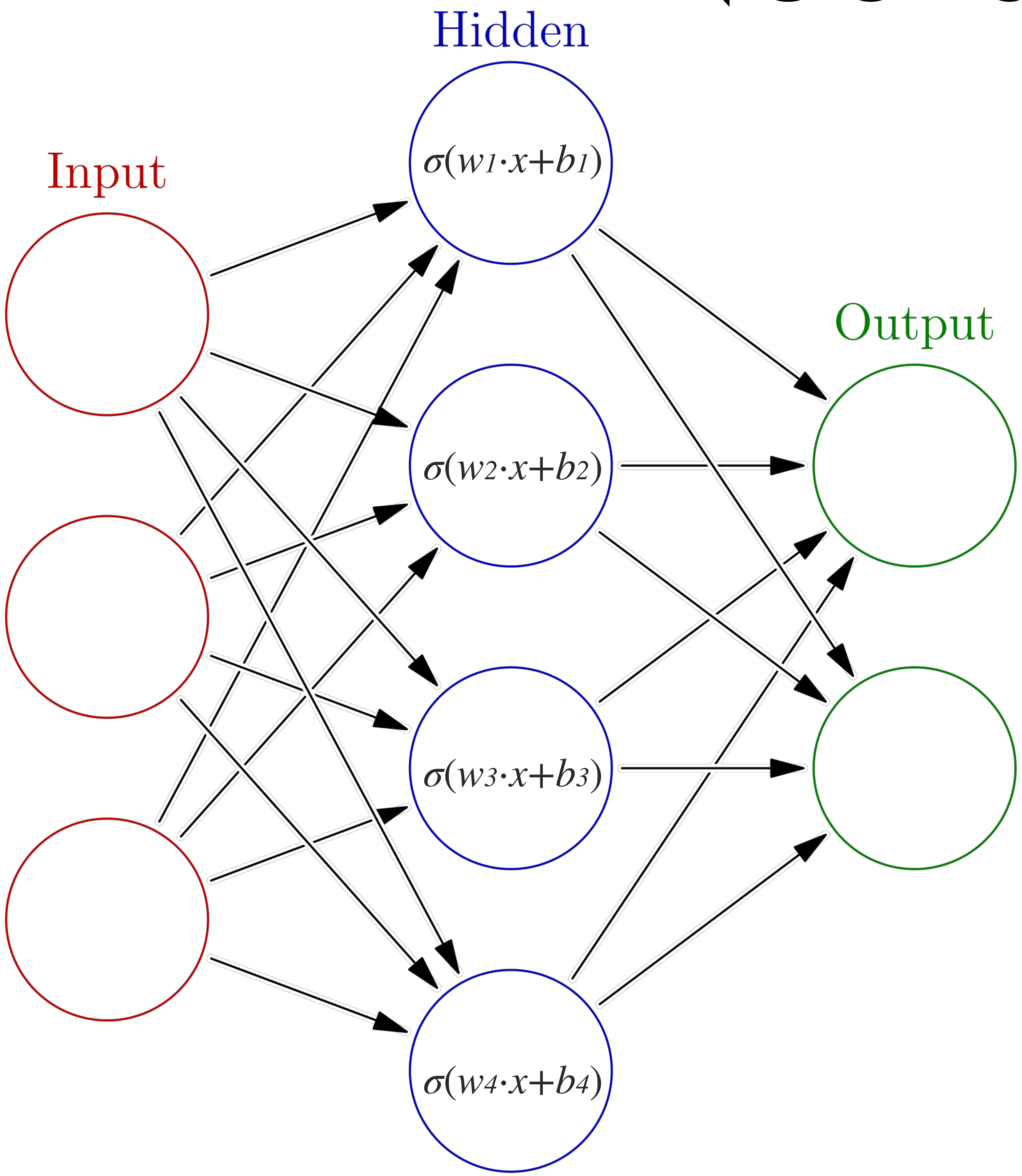
Model

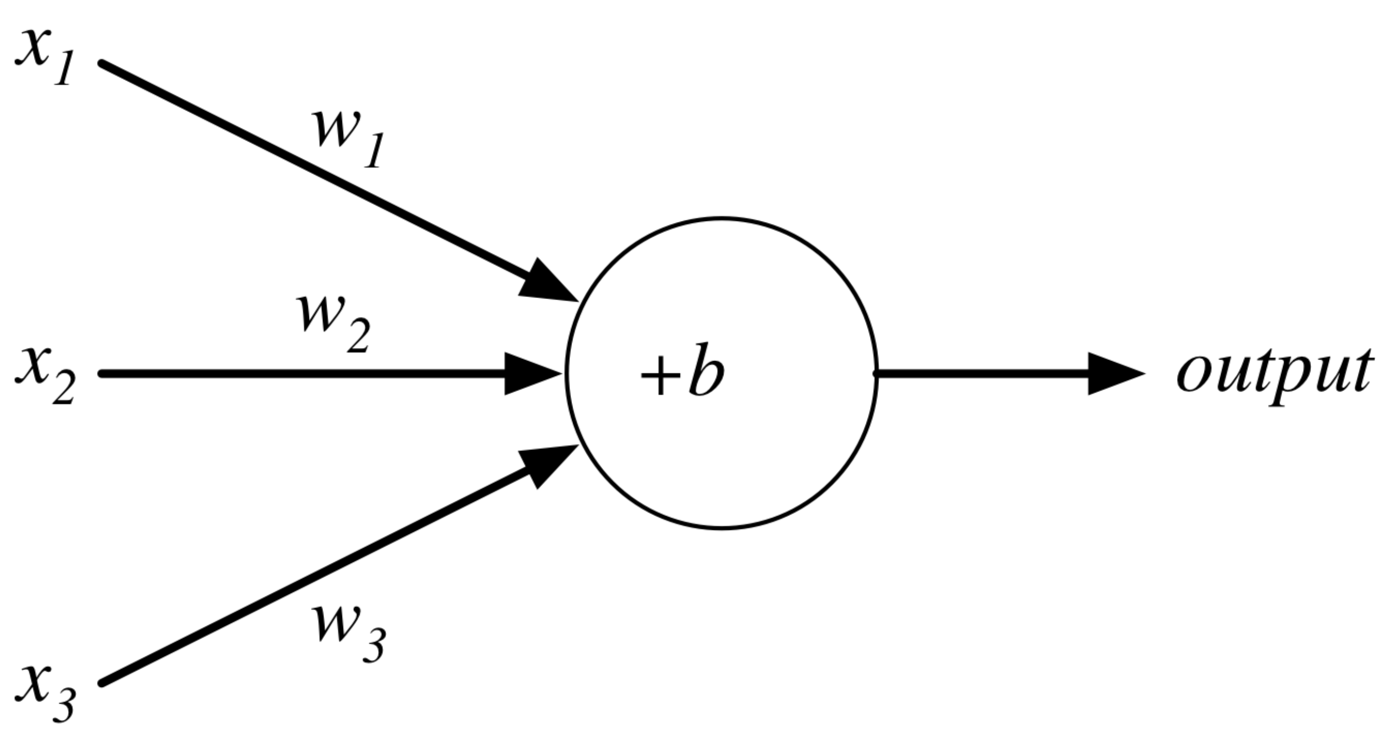
Pose Prediction

Binding Discrimination

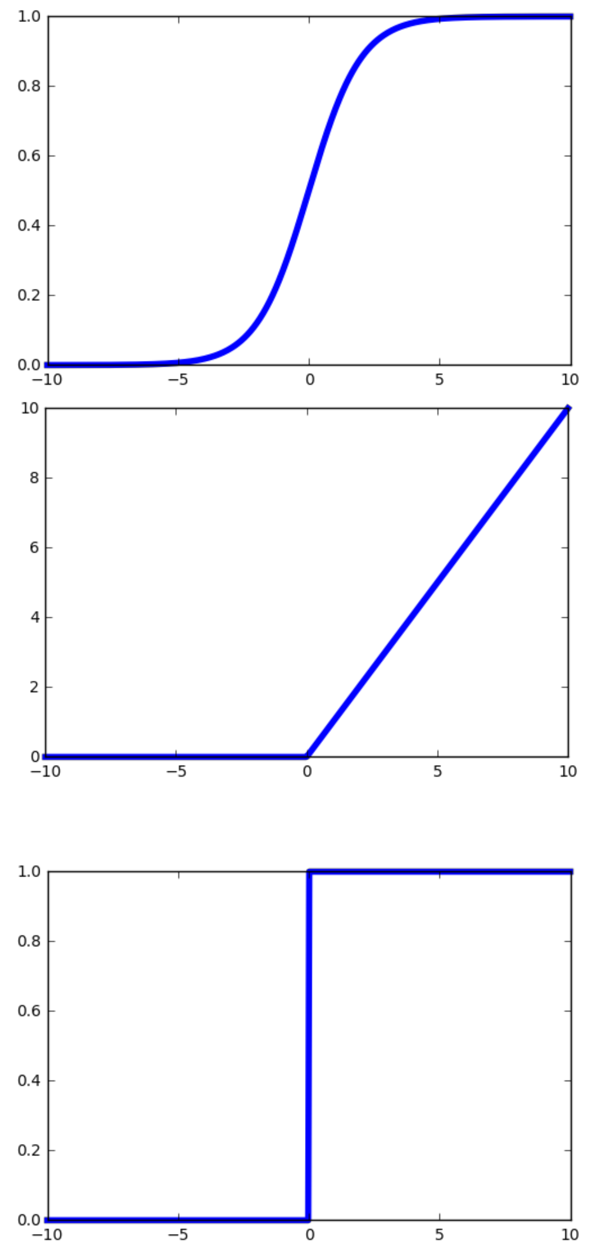
Affinity Prediction

Neural Networks



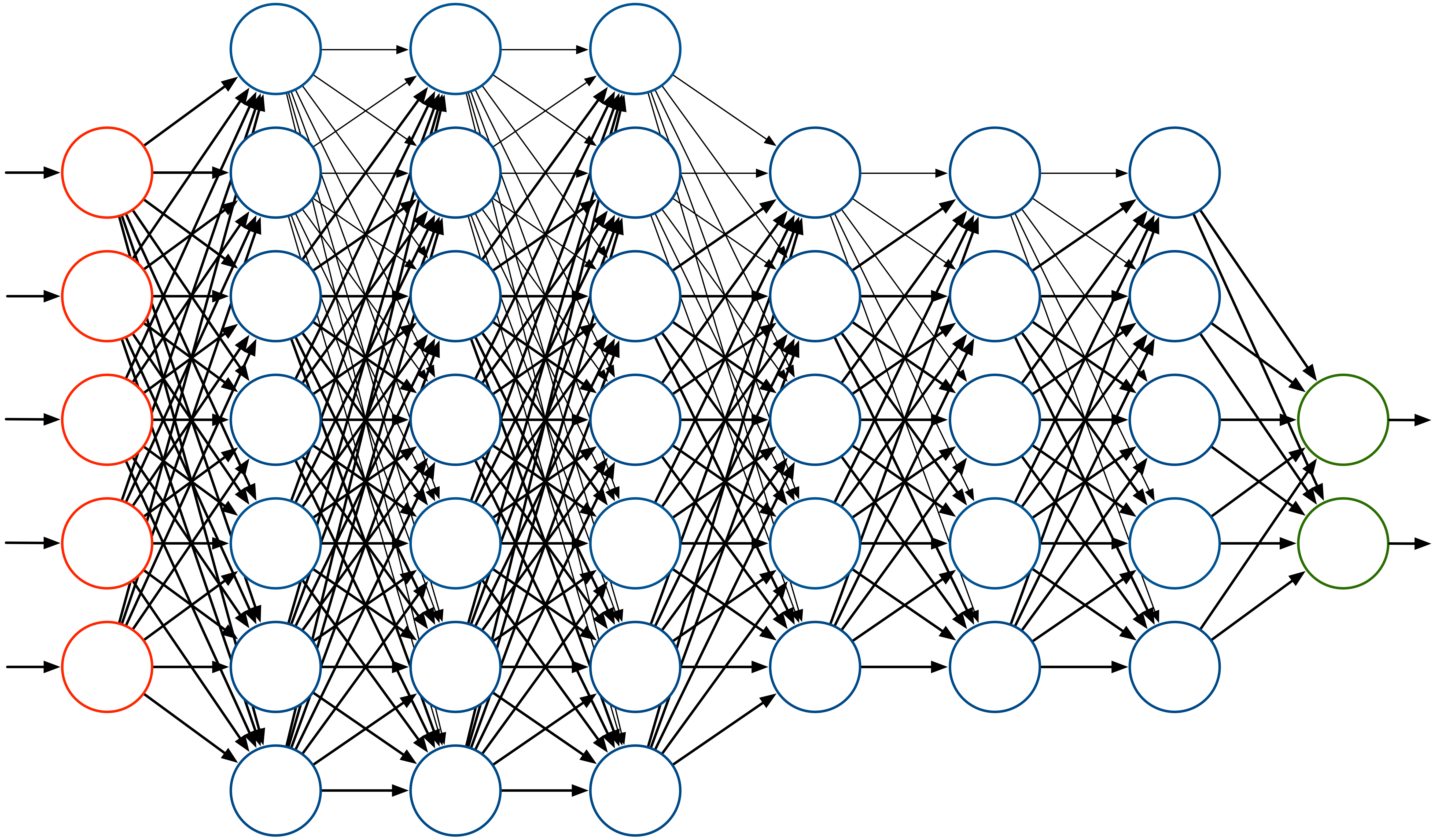


$$output = \sigma\left(\sum_i w_i x_i + b\right)$$

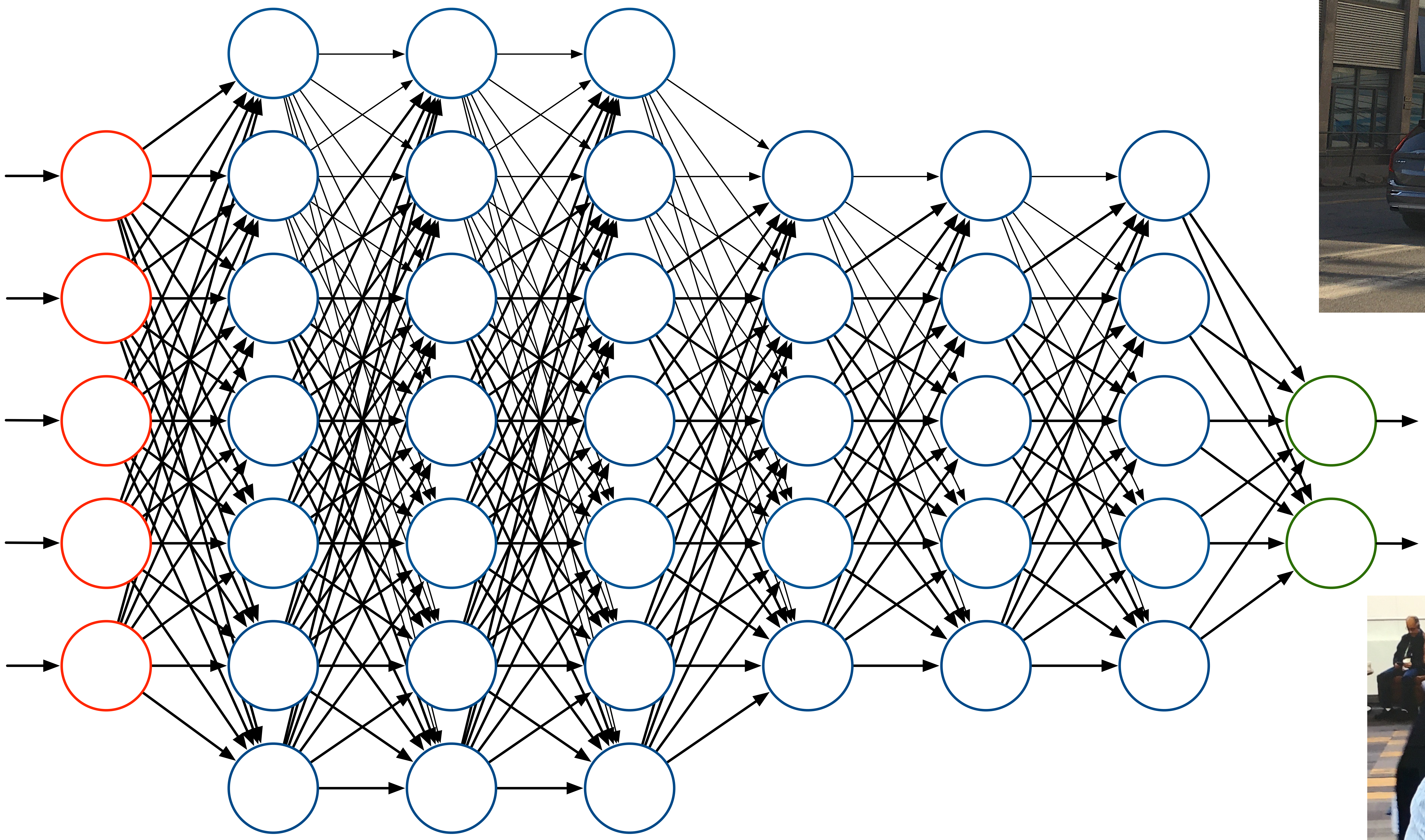


The **universal approximation theorem** states that, under reasonable assumptions, a feedforward **neural network** with a finite number of nodes **can approximate any continuous** function to within a given error over a bounded input domain.

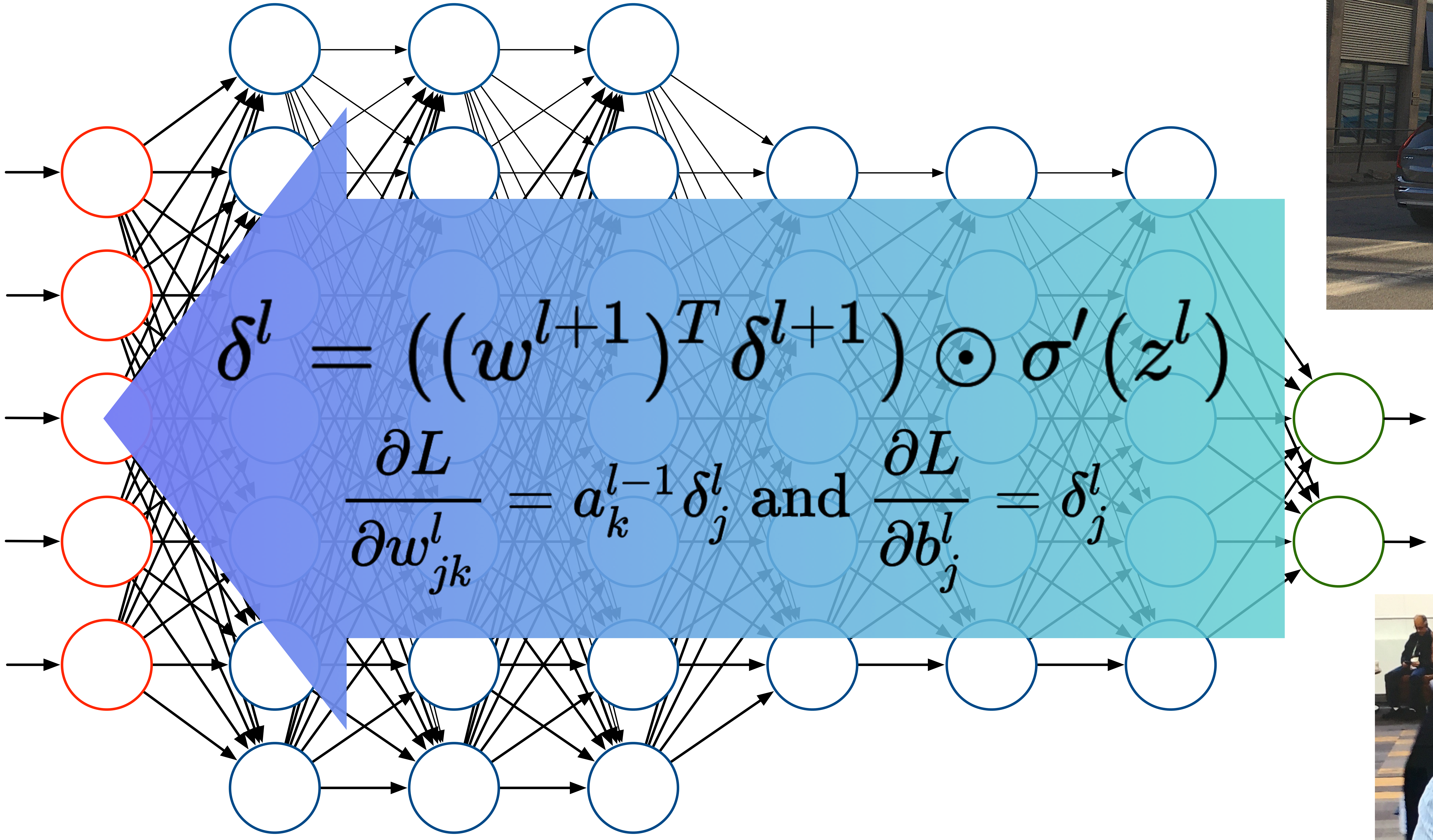
Deep Learning



Deep Learning



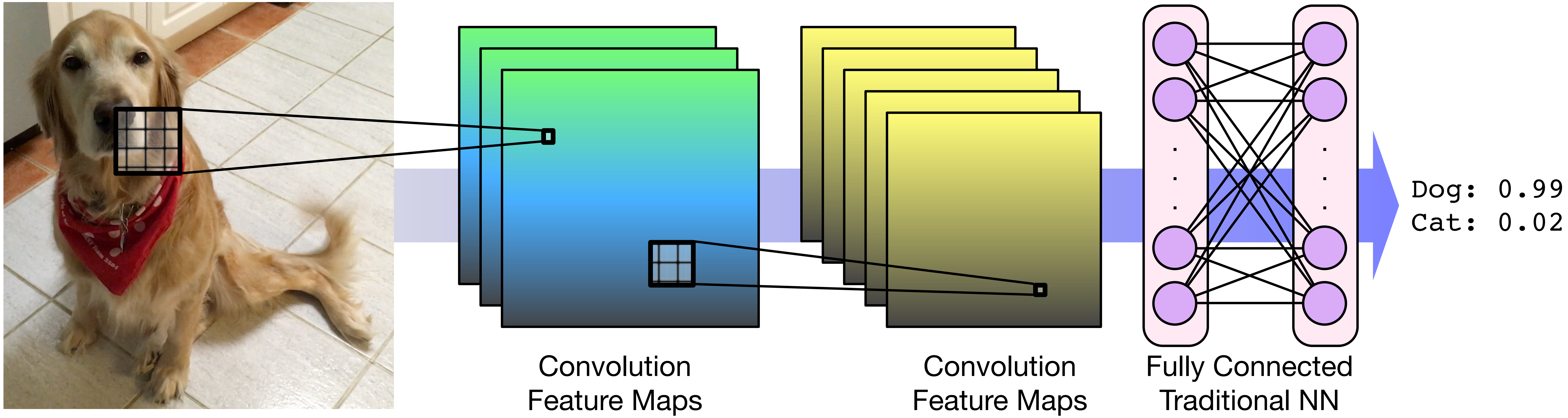
Deep Learning

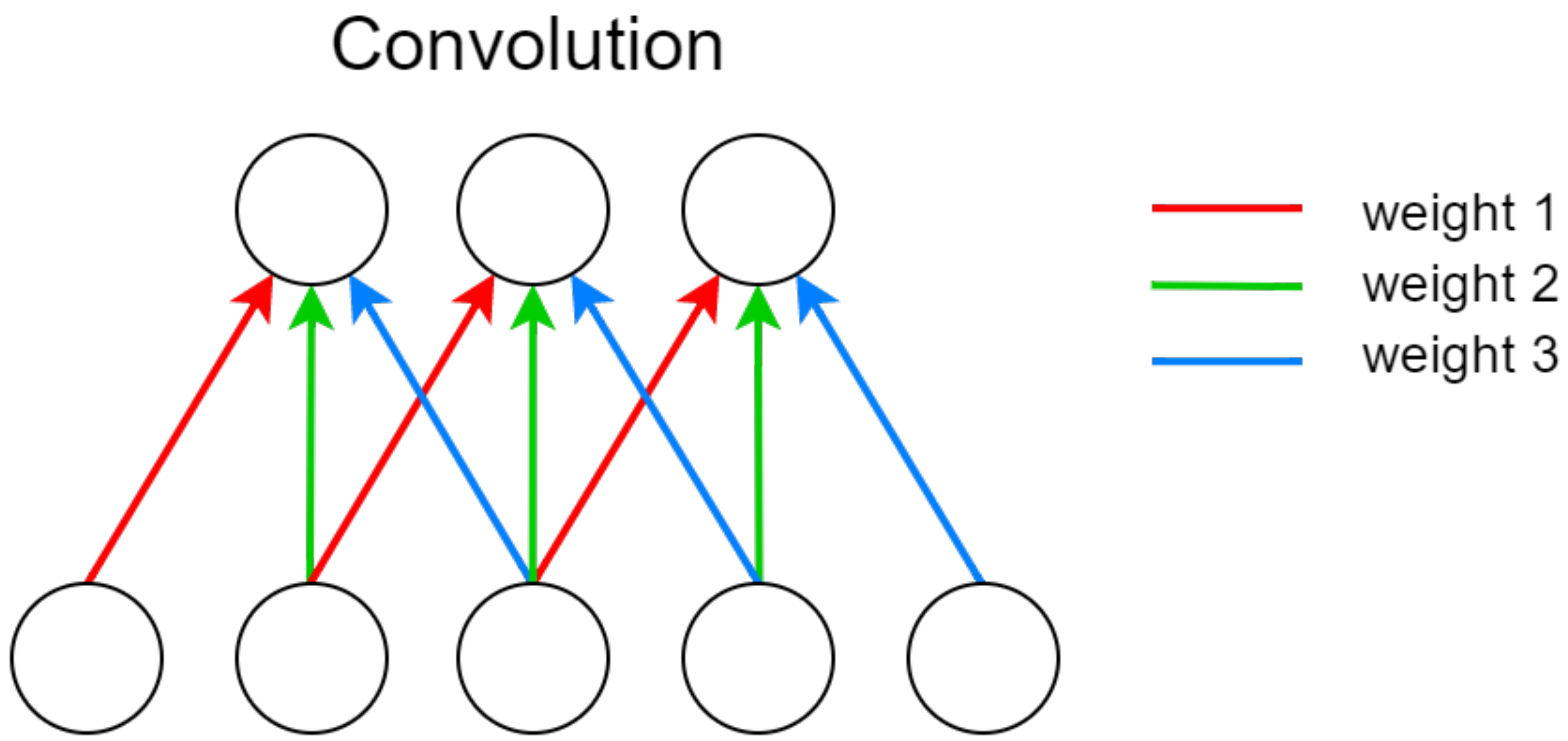


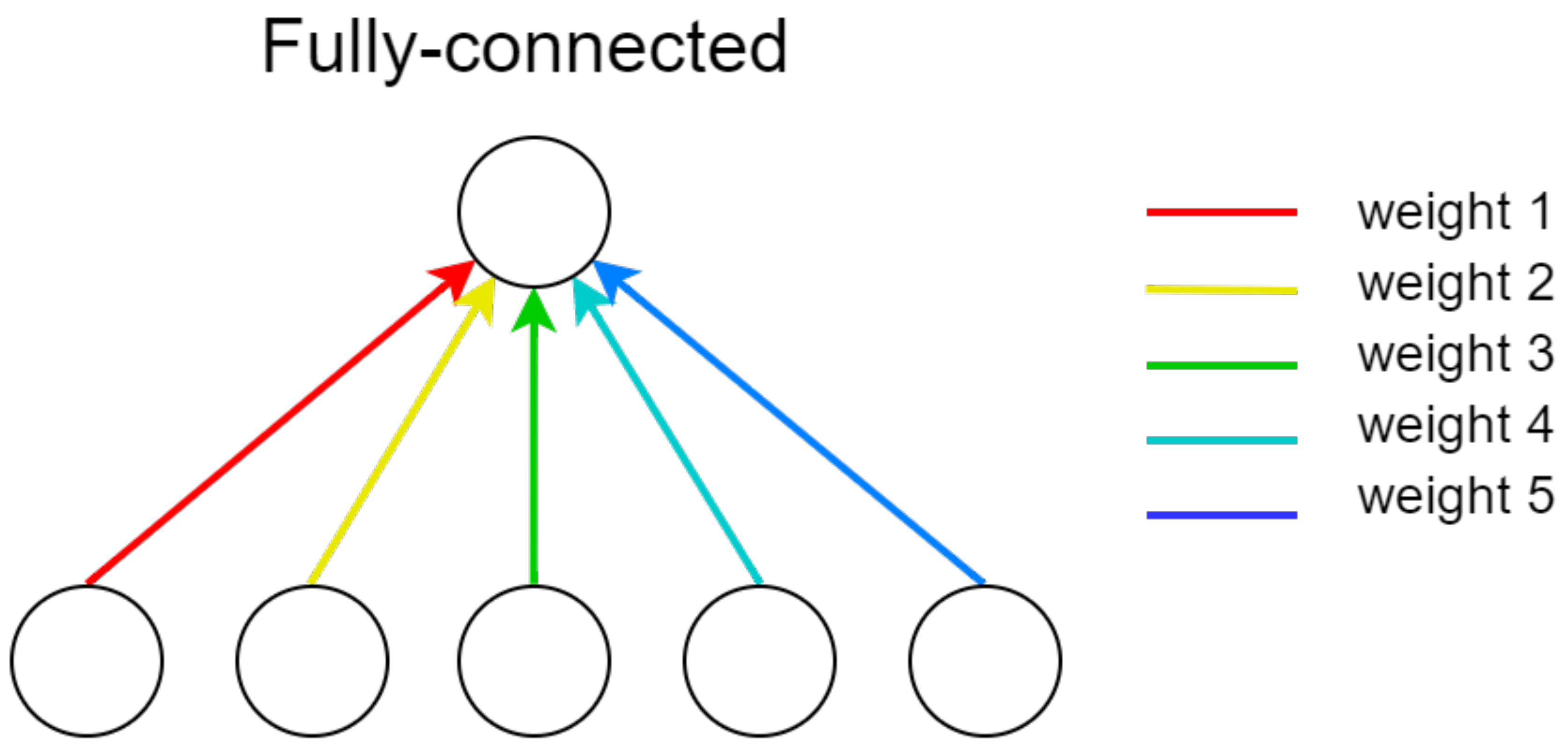
$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

$$\frac{\partial L}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \text{ and } \frac{\partial L}{\partial b^l_j} = \delta^l_j$$

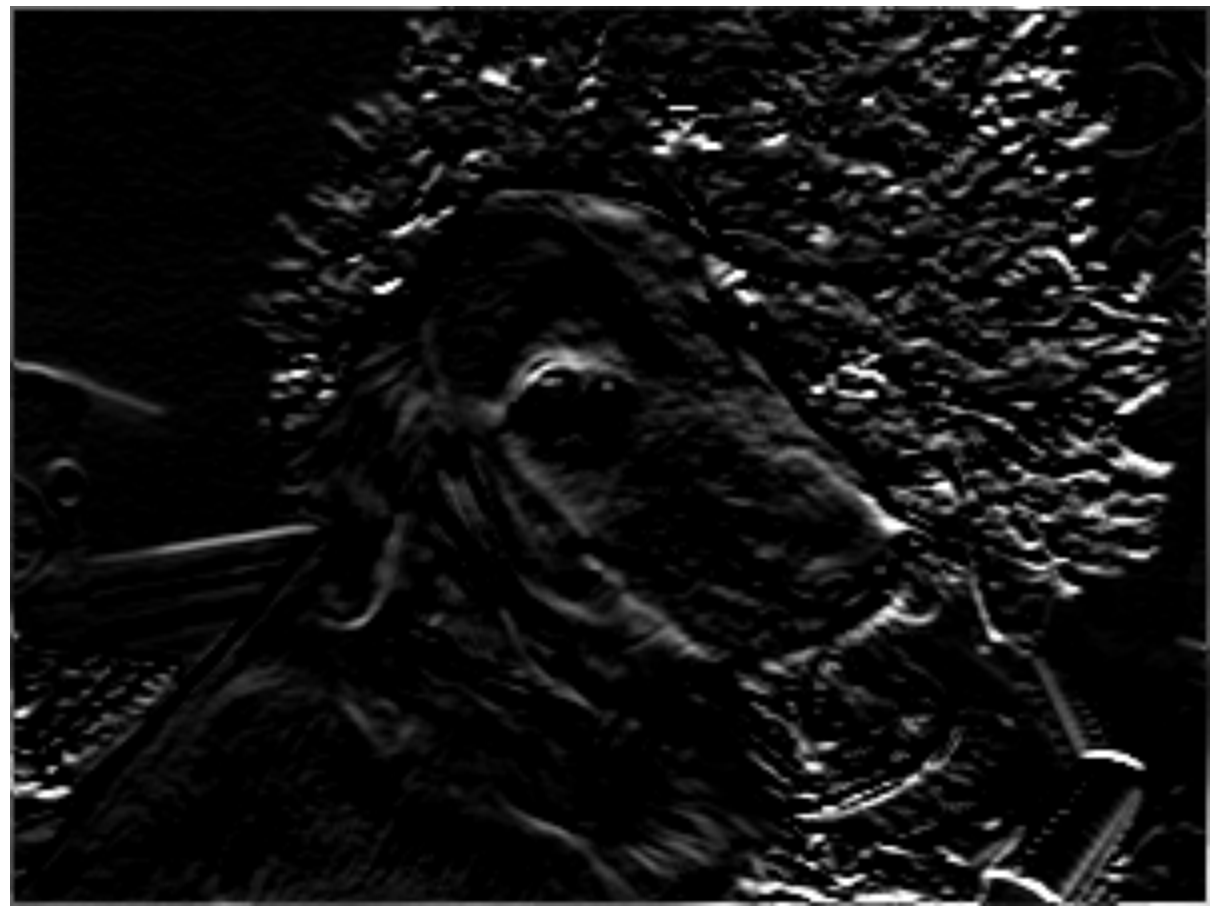
Convolutional Neural Networks







Convolutional Filters

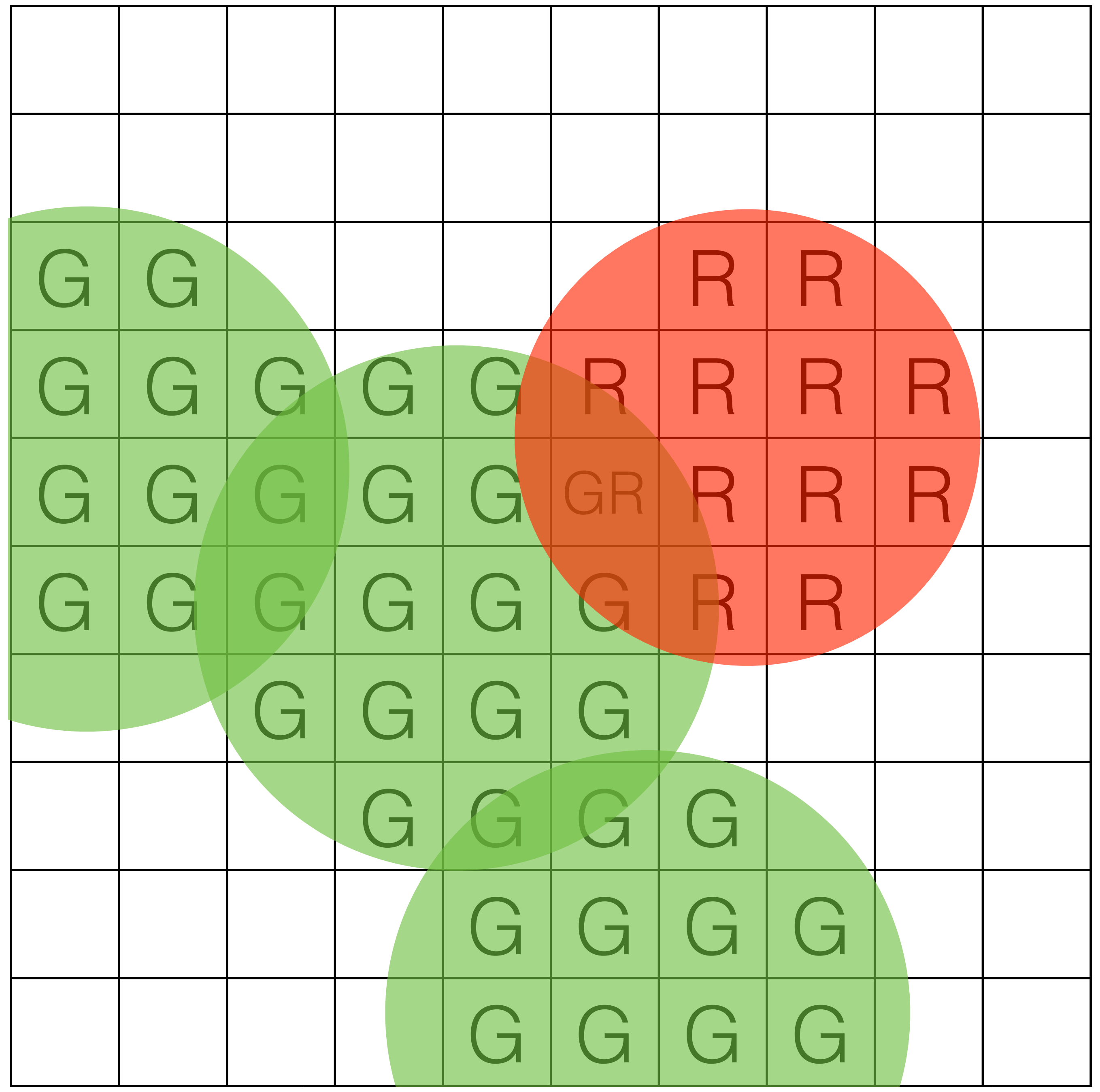


-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

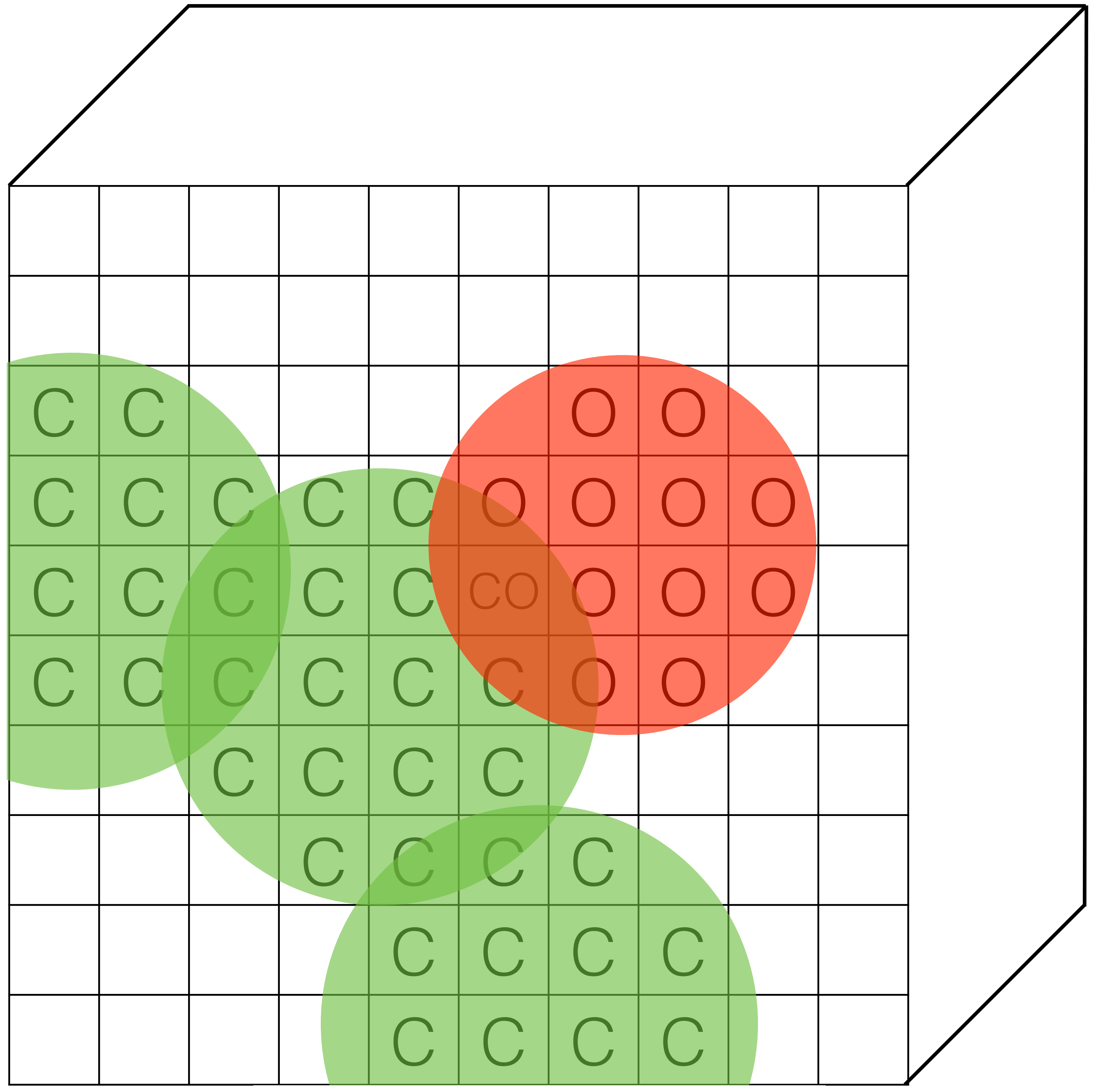
-1	-1	-1
-1	8	-1
-1	-1	-1

Protein-Ligand Representation



(R,G,B) pixel

Protein-Ligand Representation



(R,G,B) pixel →

(Carbon, Nitrogen, Oxygen,…) **voxel**

The only parameters for this representation are the choice of **grid resolution**, **atom density**, and **atom types**.

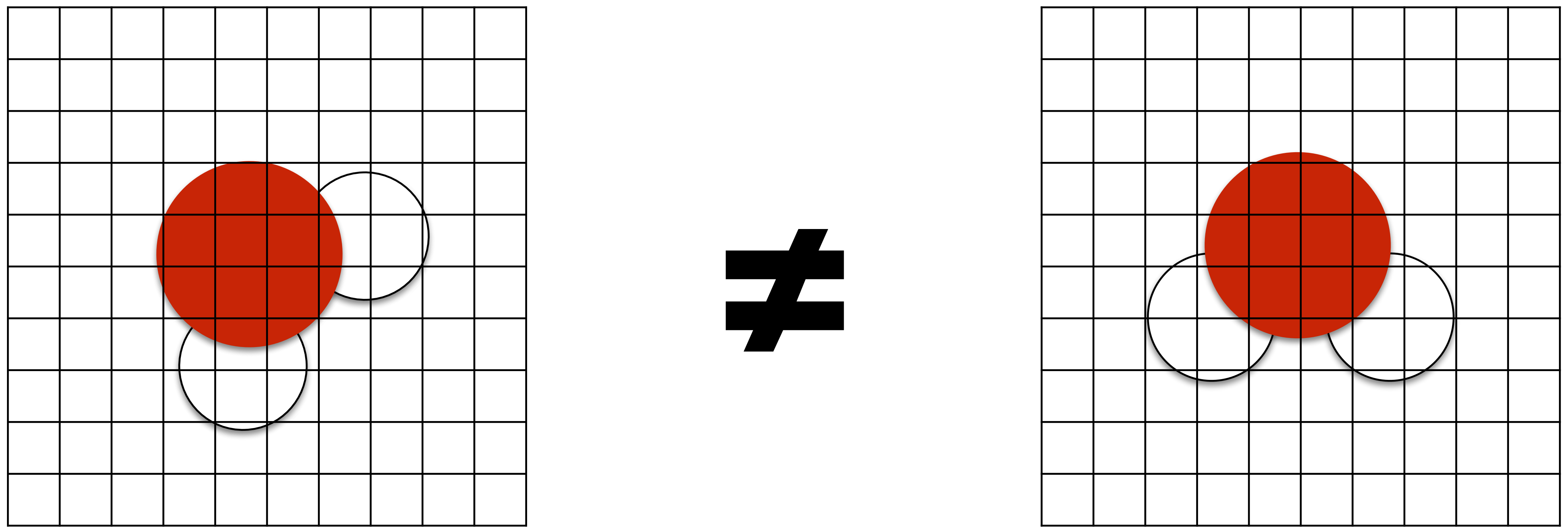
Why Grids?

Cons

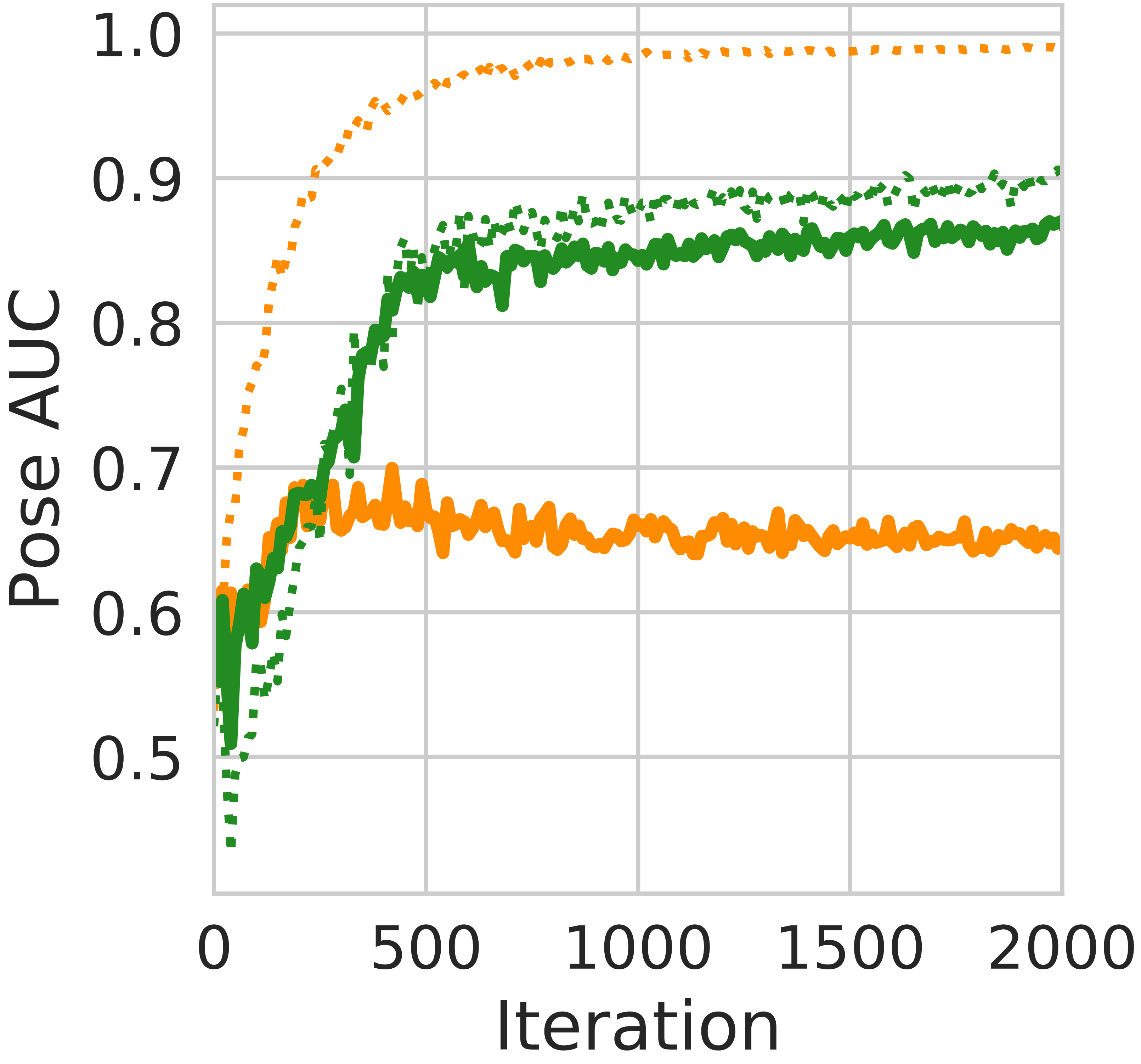
- *coordinate frame dependent*
- pairwise interactions not explicit

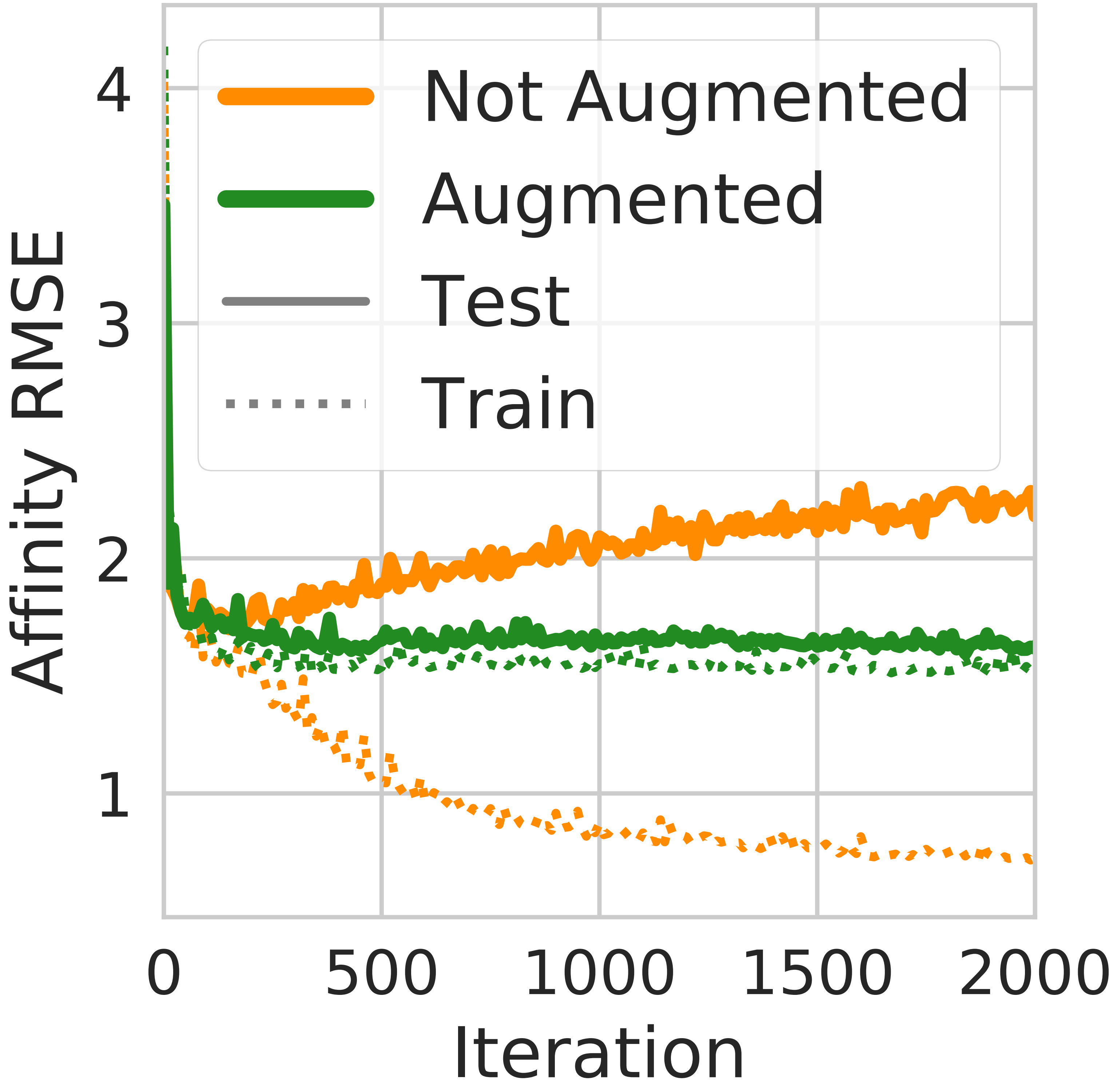
Pros

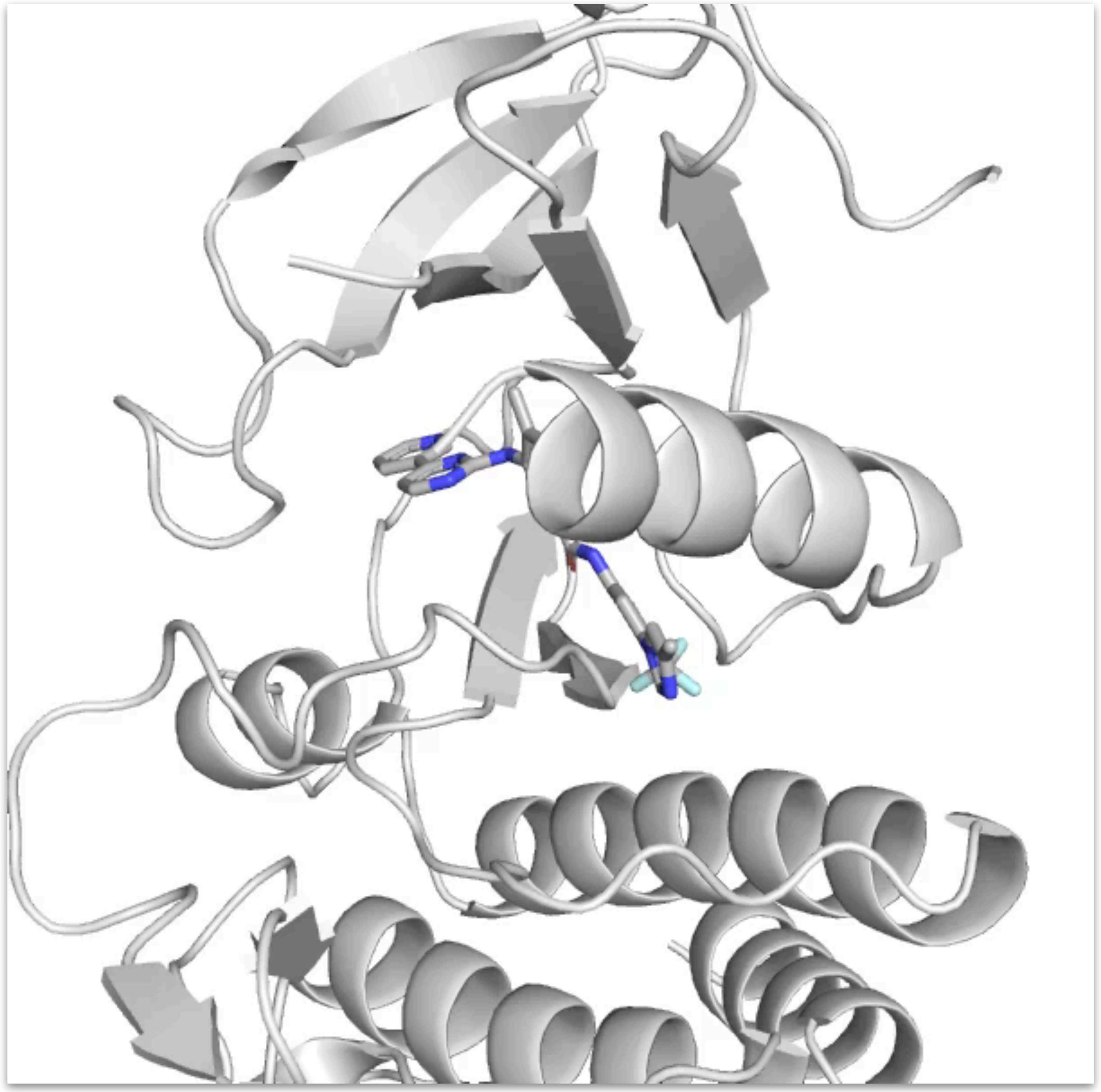
- clear spatial relationships
- amazingly parallel
- easy to interpret



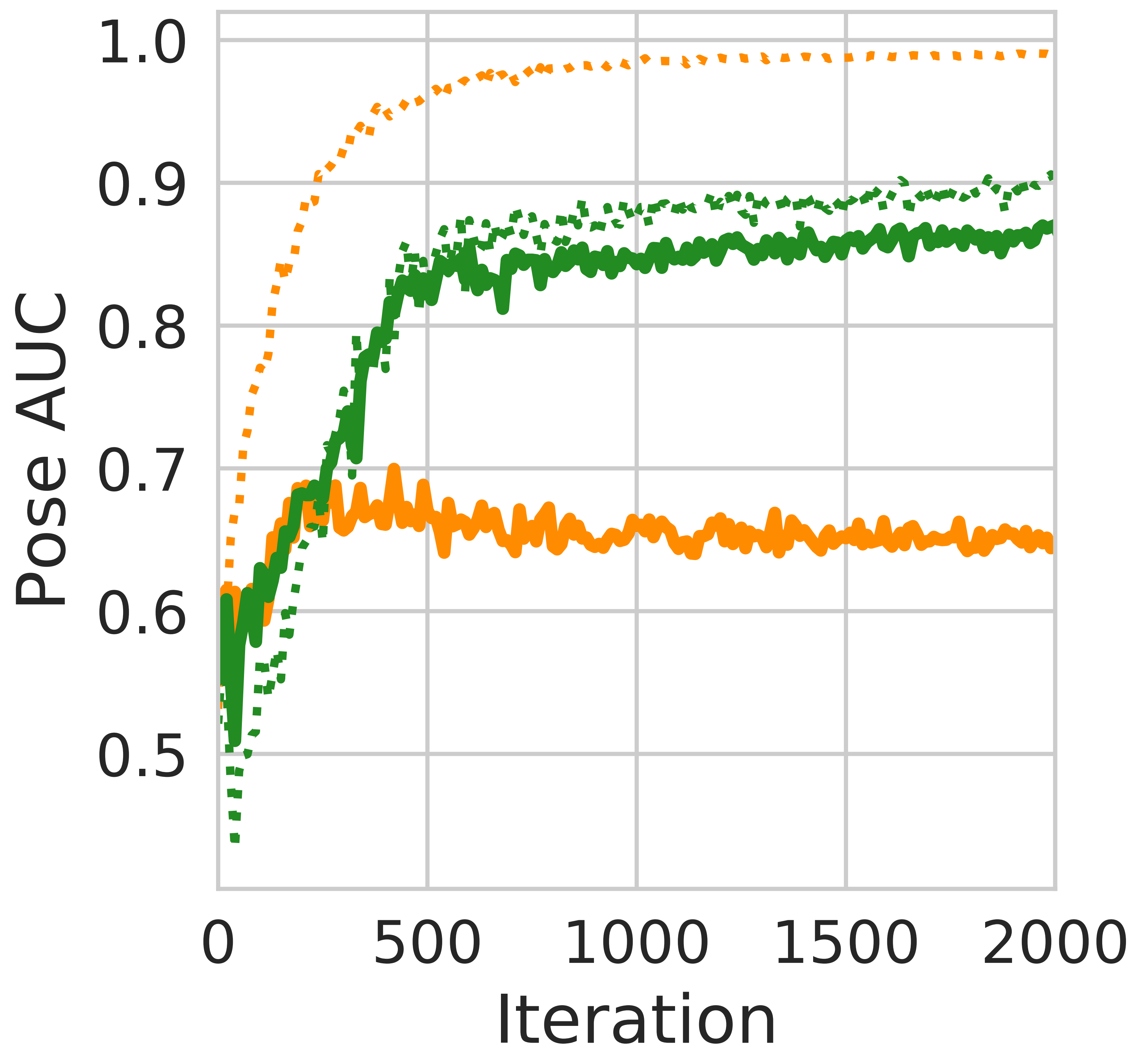
Data Augmentation

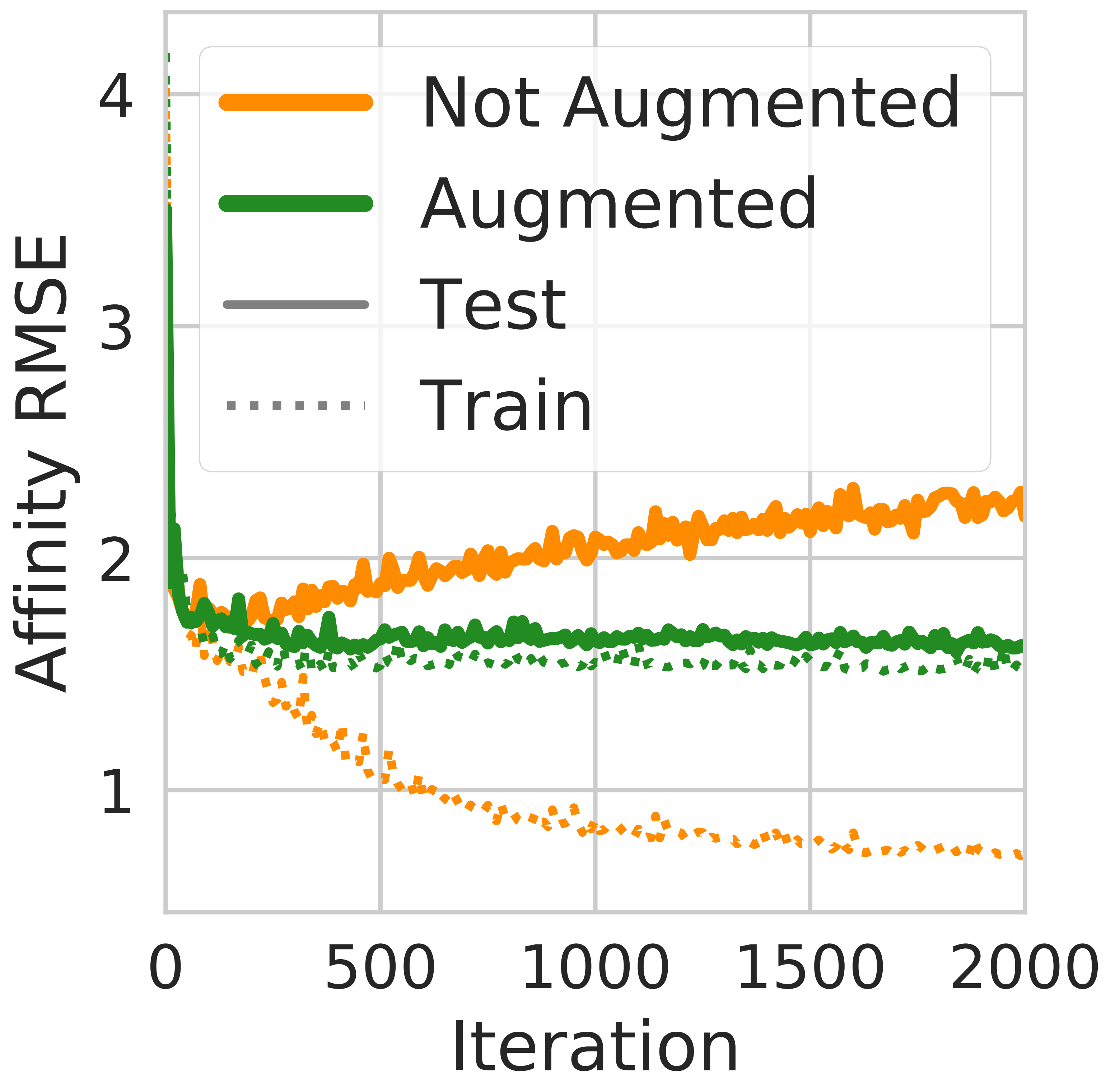


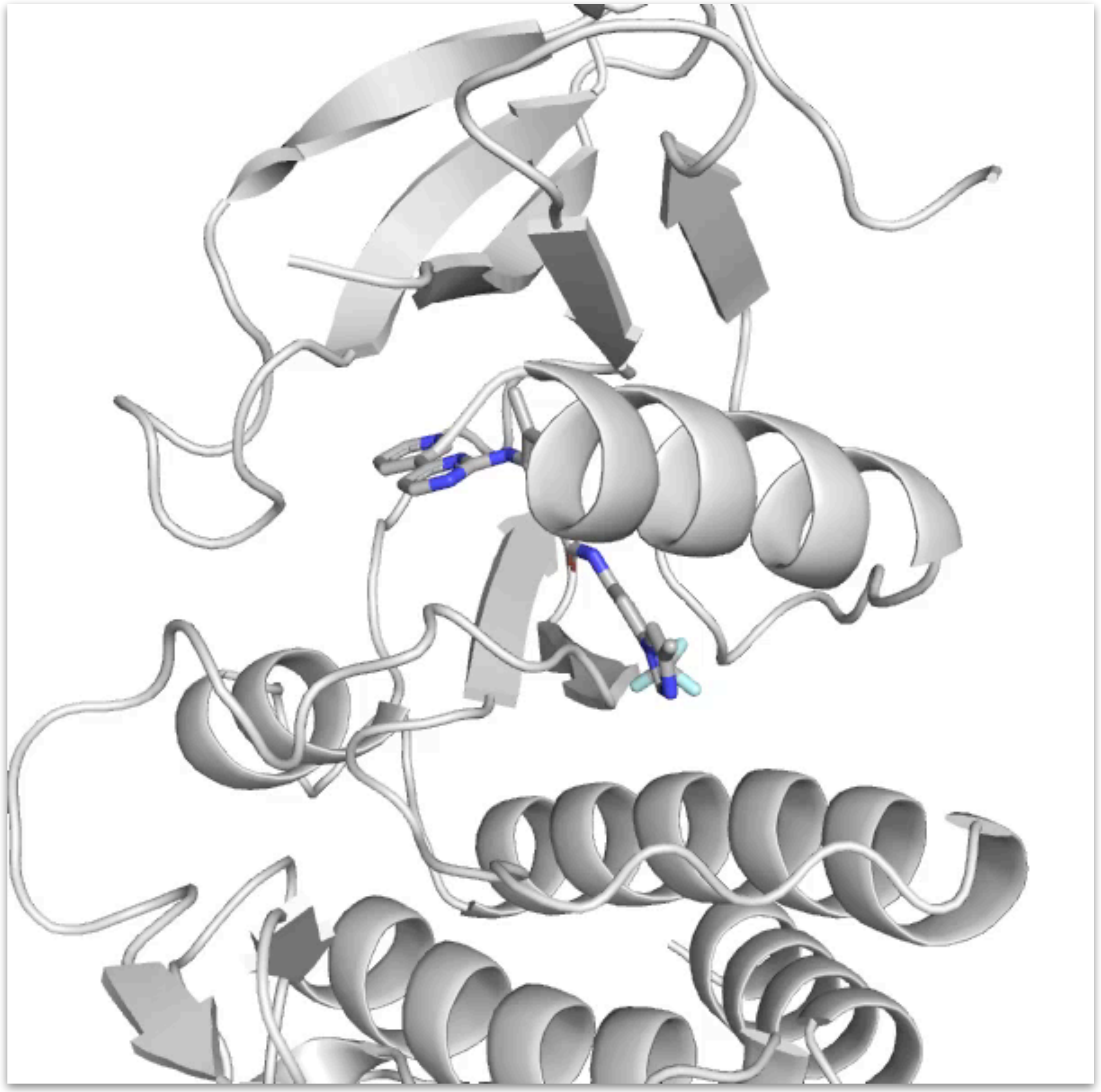




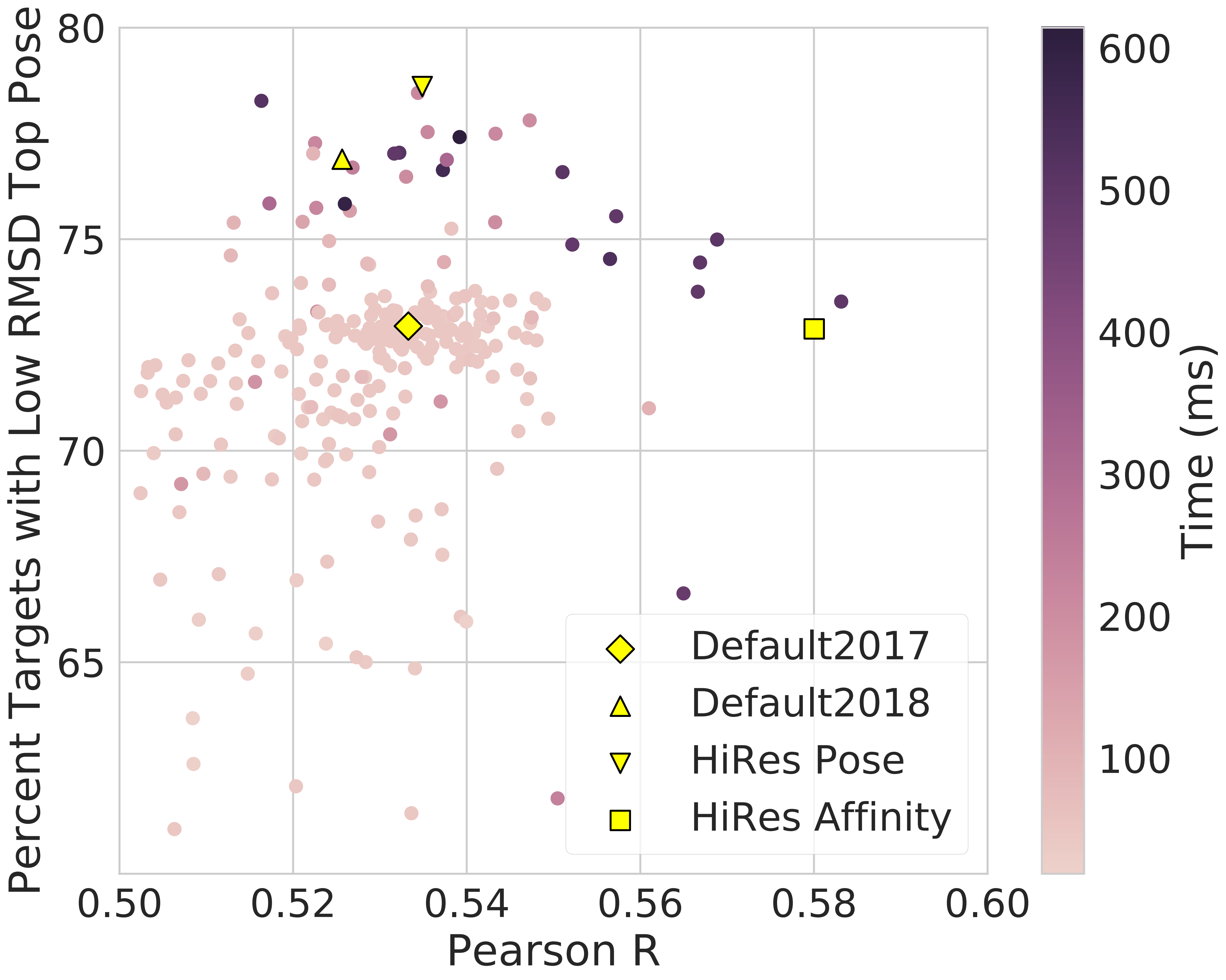
Data Augmentation



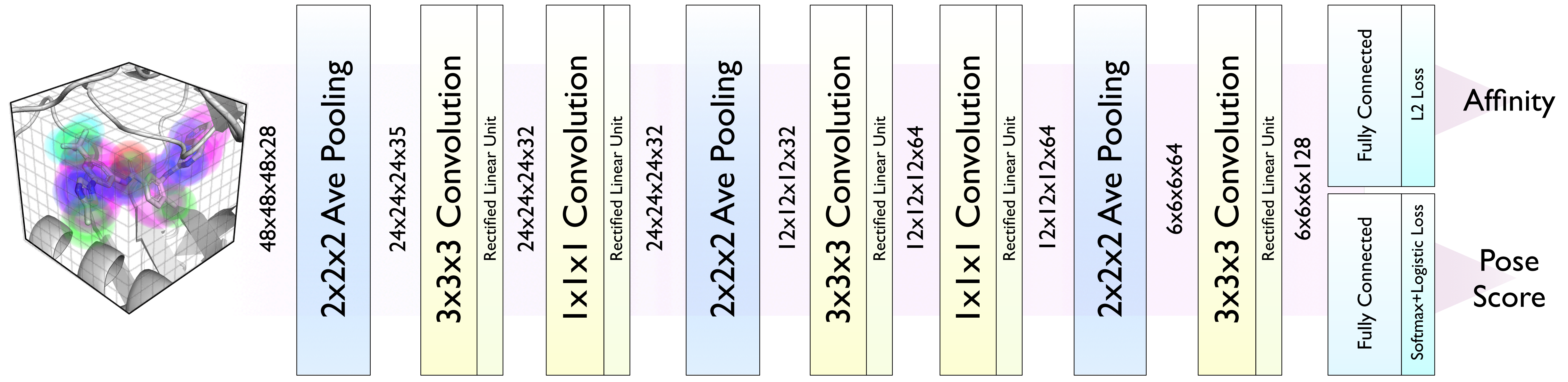


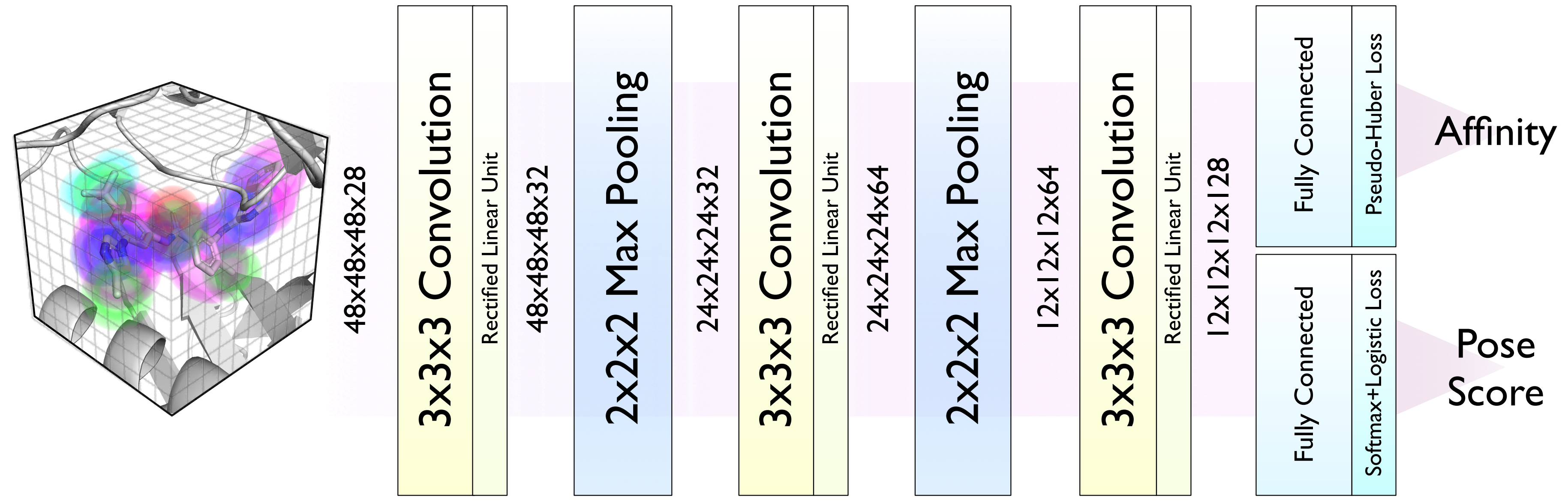


Optimized Models

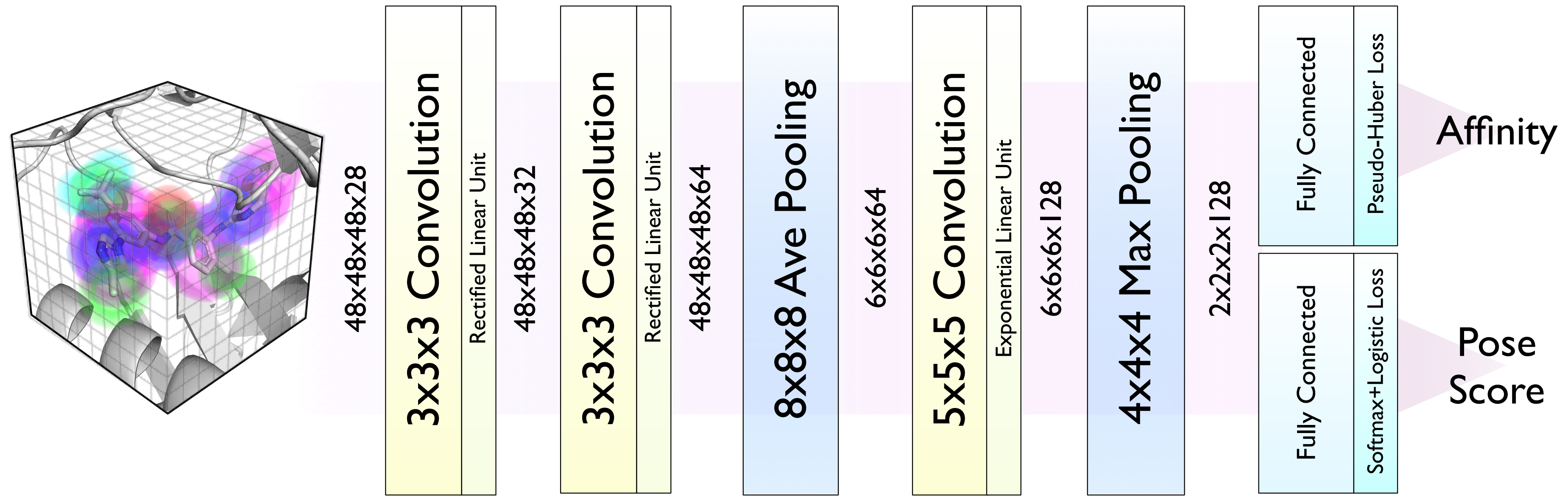


Default2018



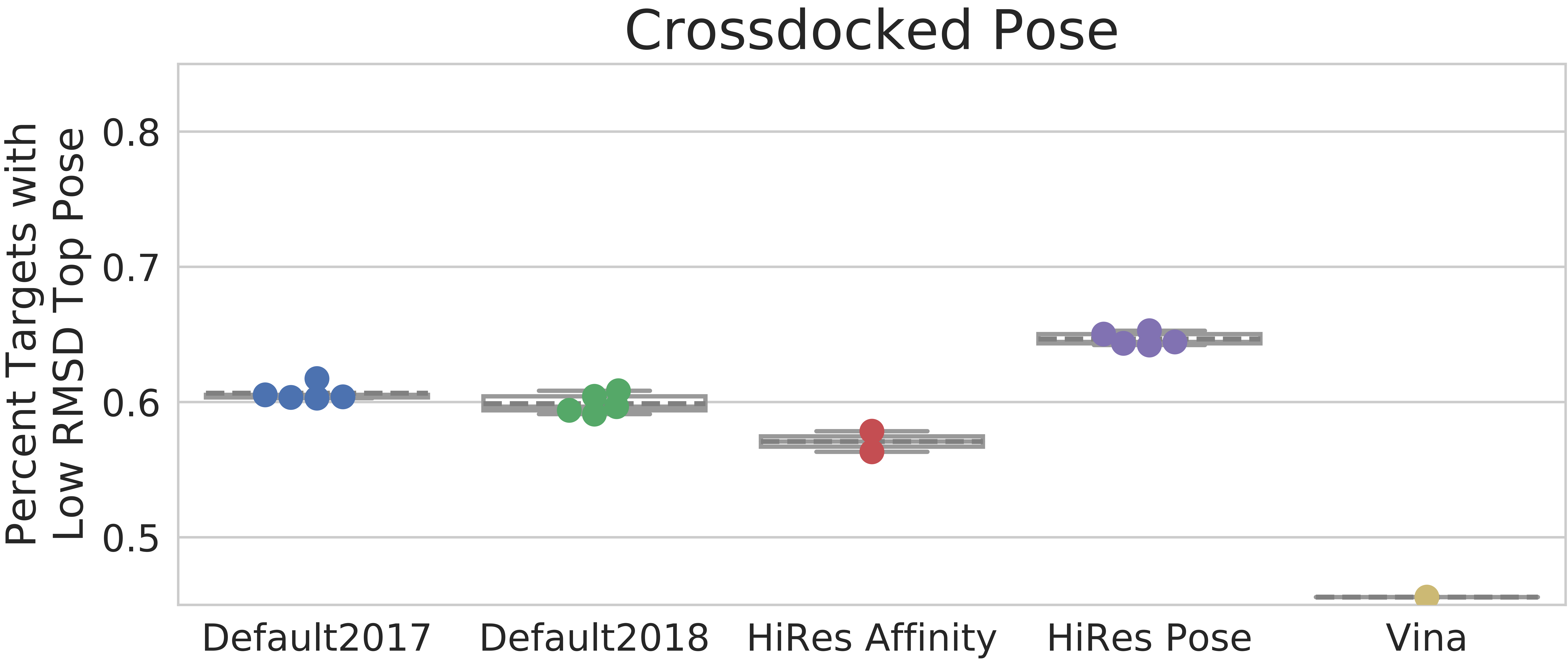


HiRes Pose

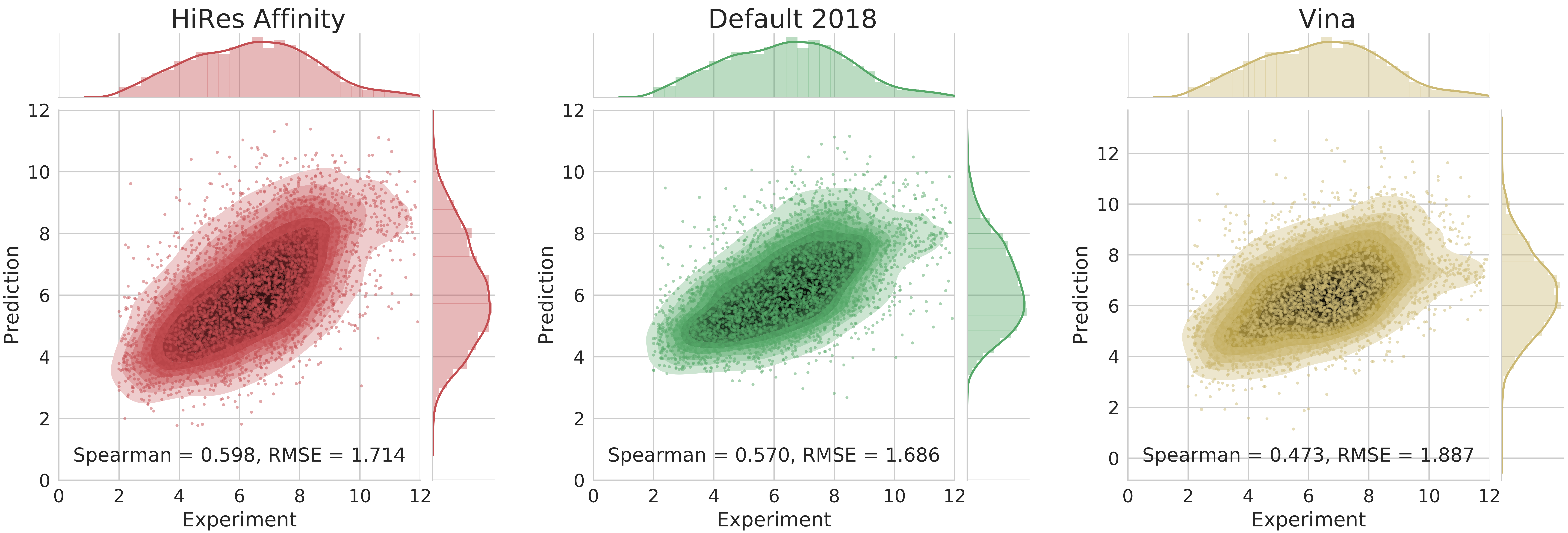


HiRes Affinity

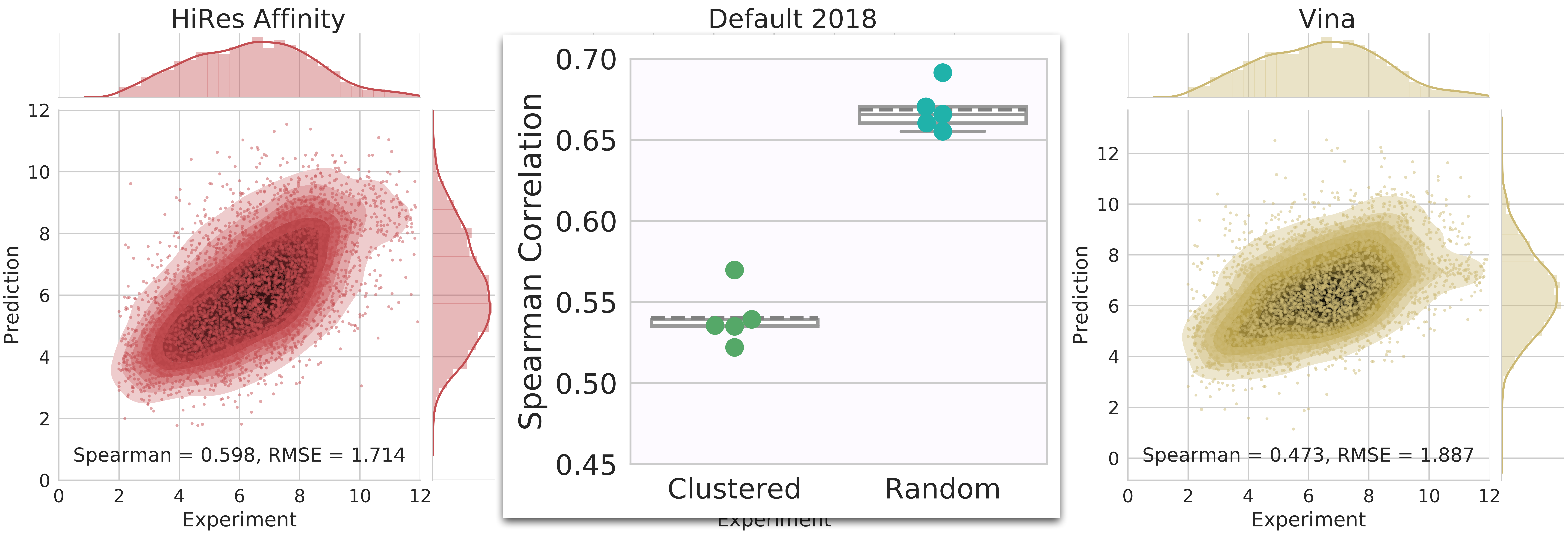
Pose Results



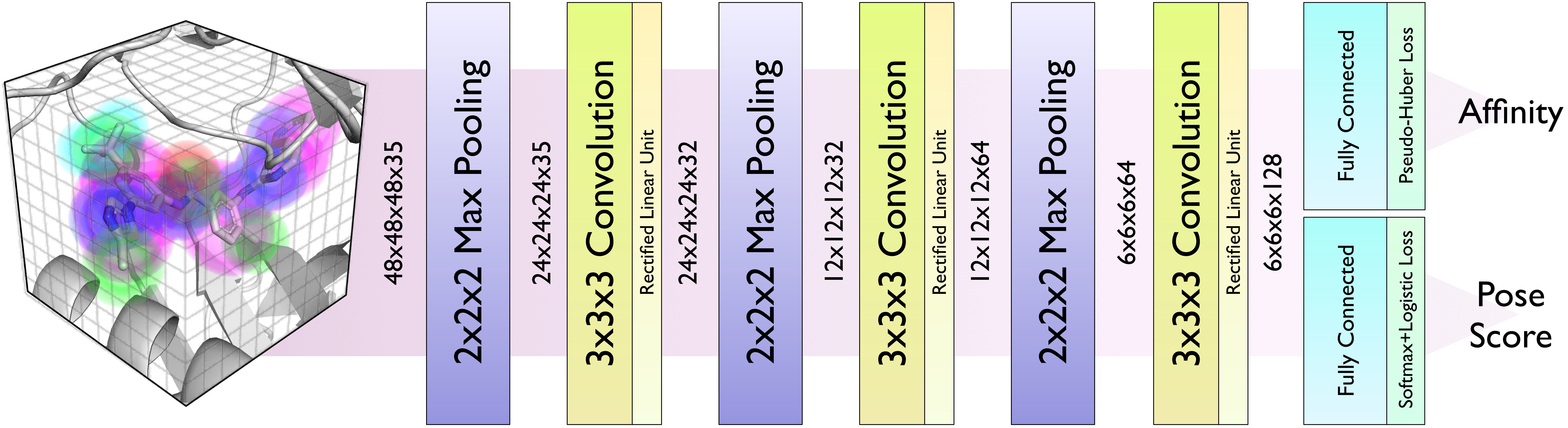
Affinity Results



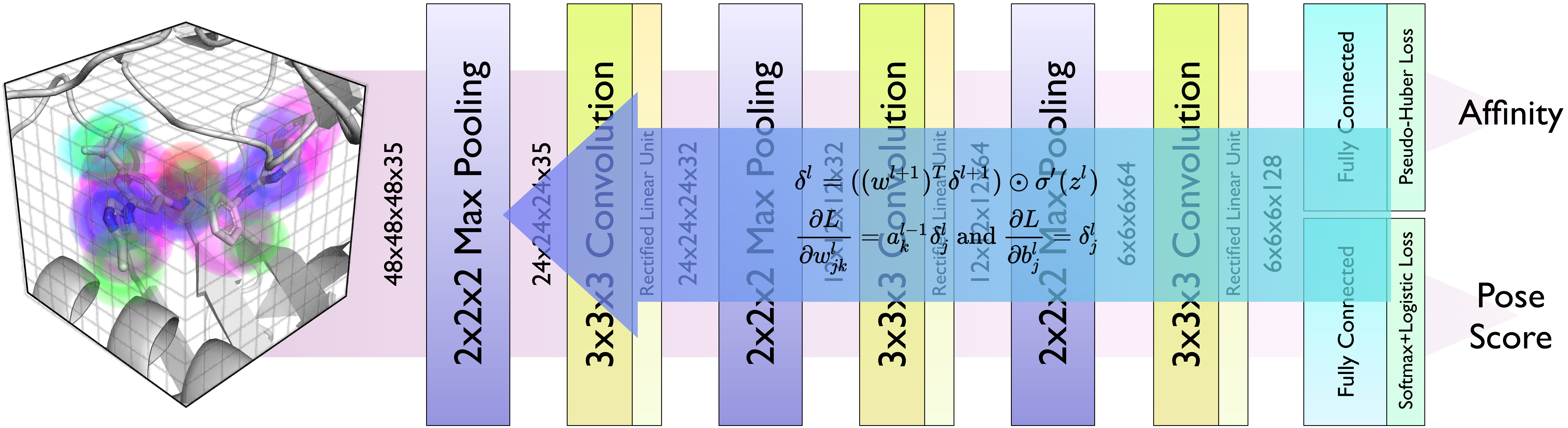
Affinity Results



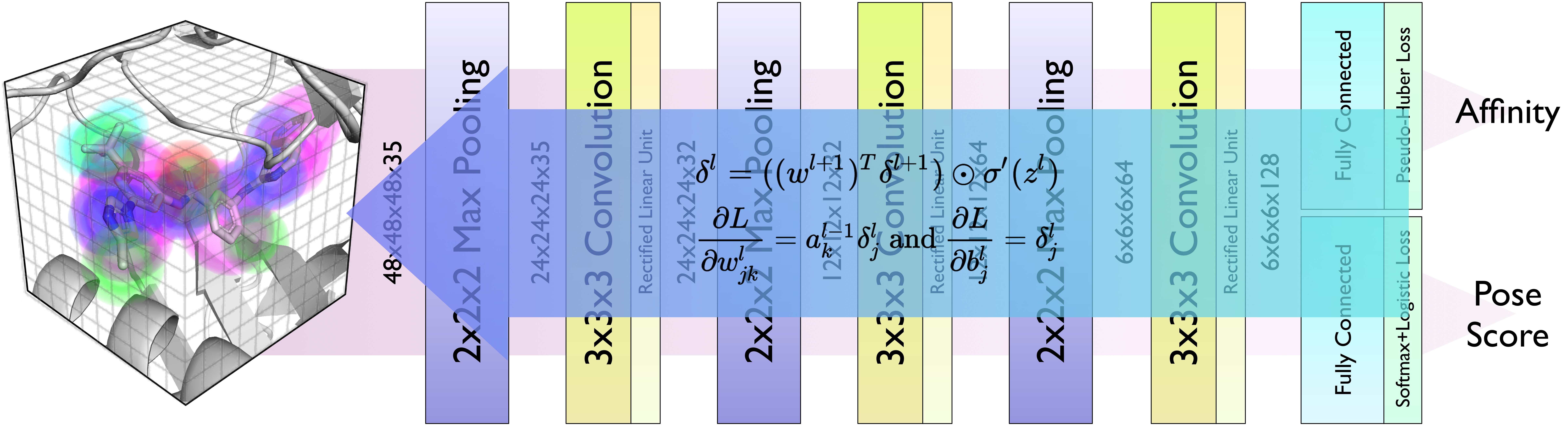
Beyond Scoring



Beyond Scoring



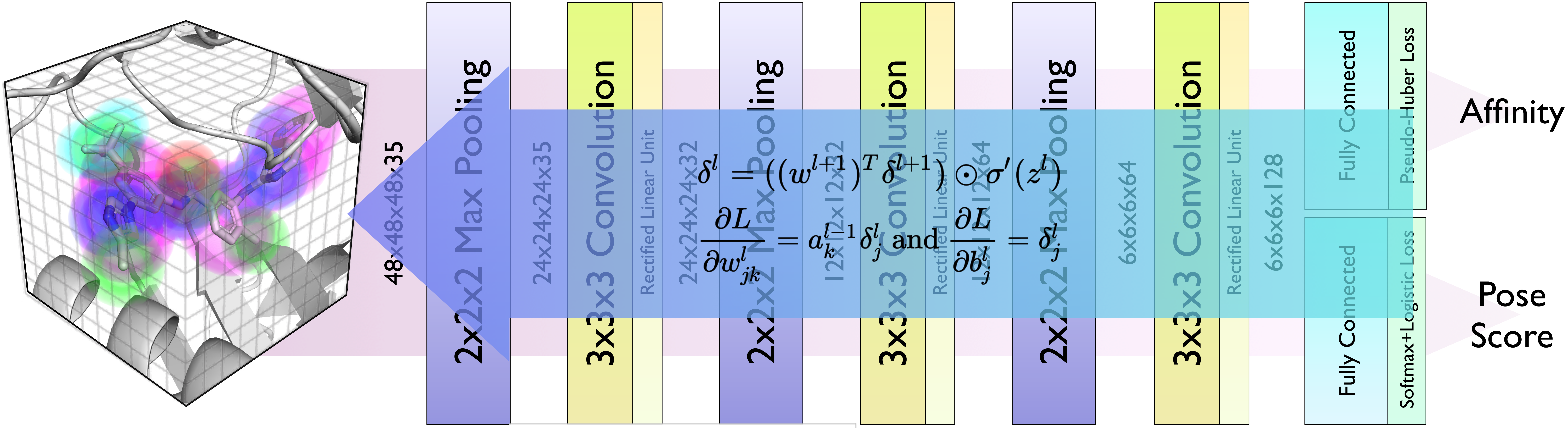
Beyond Scoring



$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

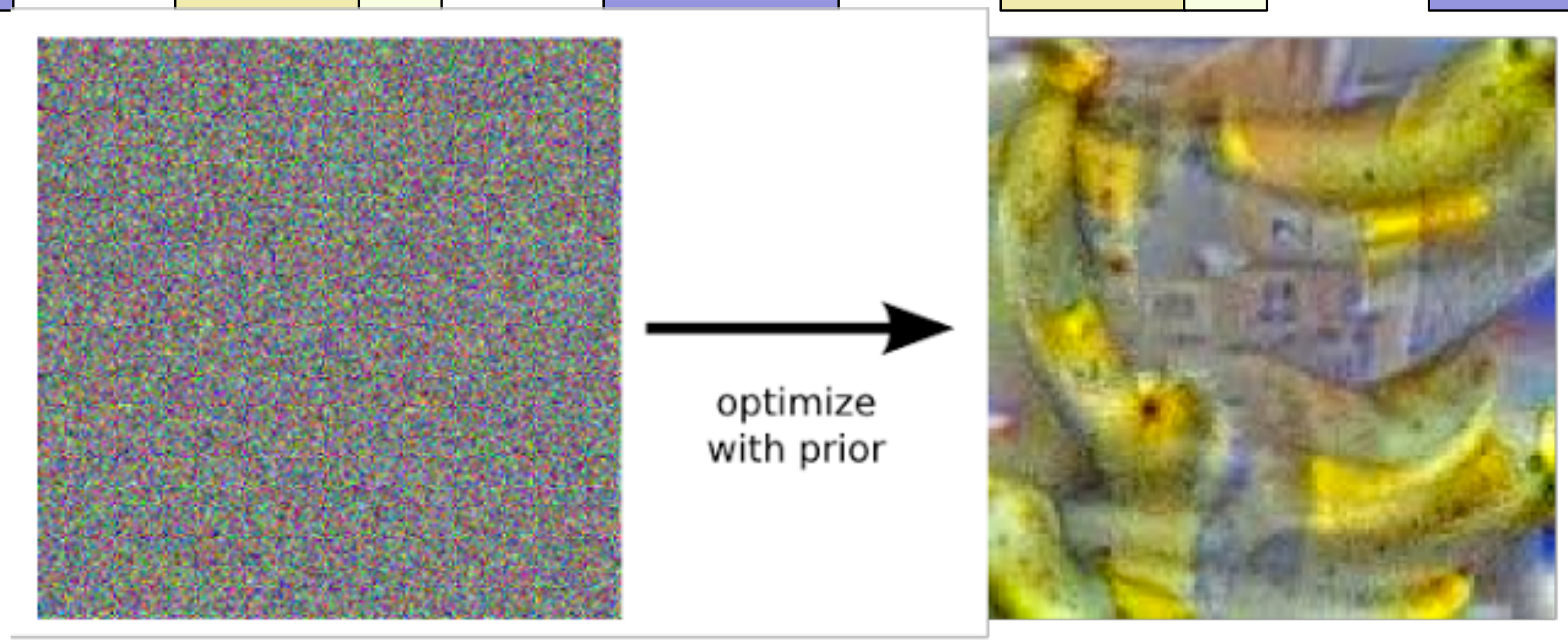
$$\frac{\partial L}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \text{ and } \frac{\partial L}{\partial b^l_j} = \delta^l_j$$

Beyond Scoring



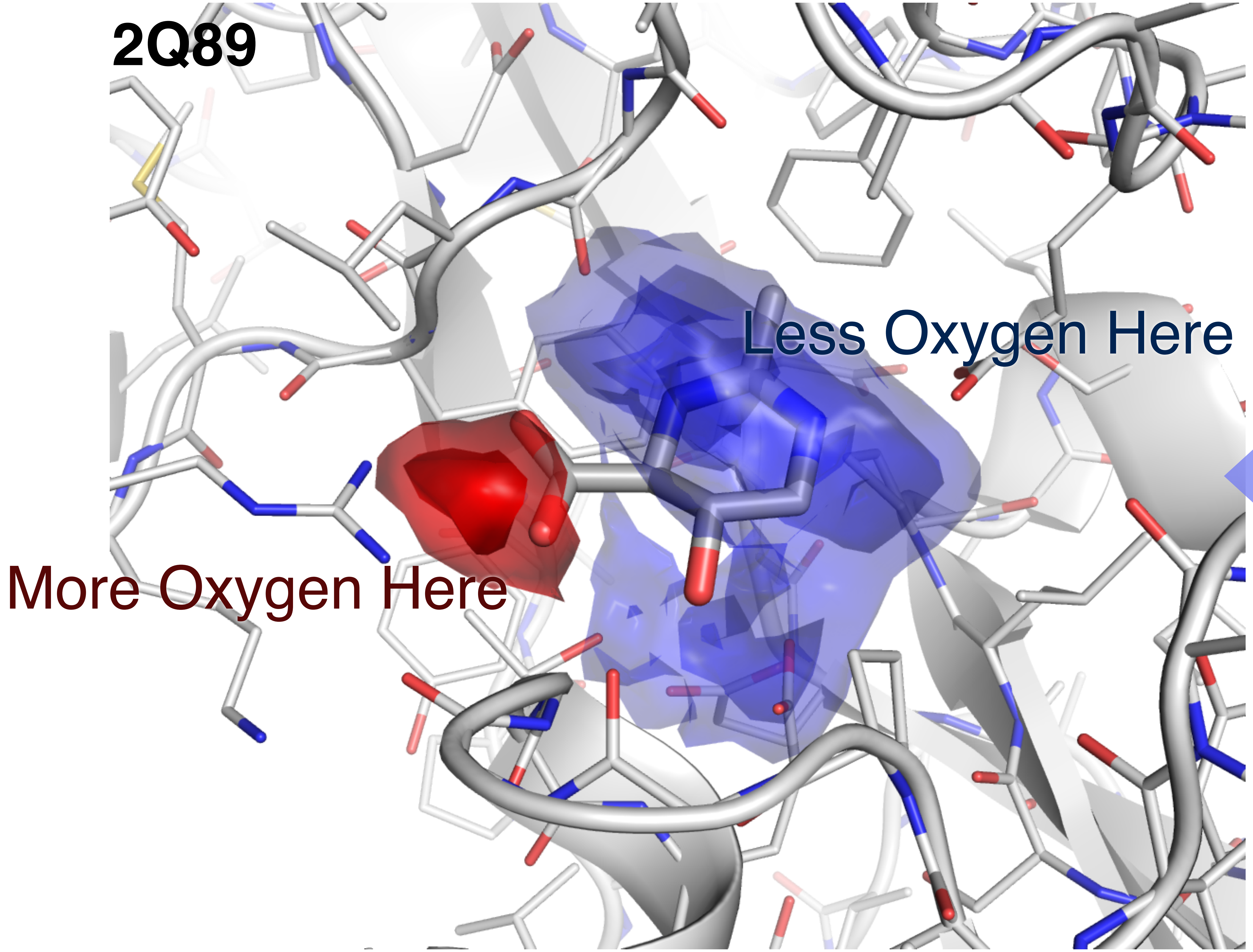
$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

$$\frac{\partial L}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \text{ and } \frac{\partial L}{\partial b^l_j} = \delta^l_j$$



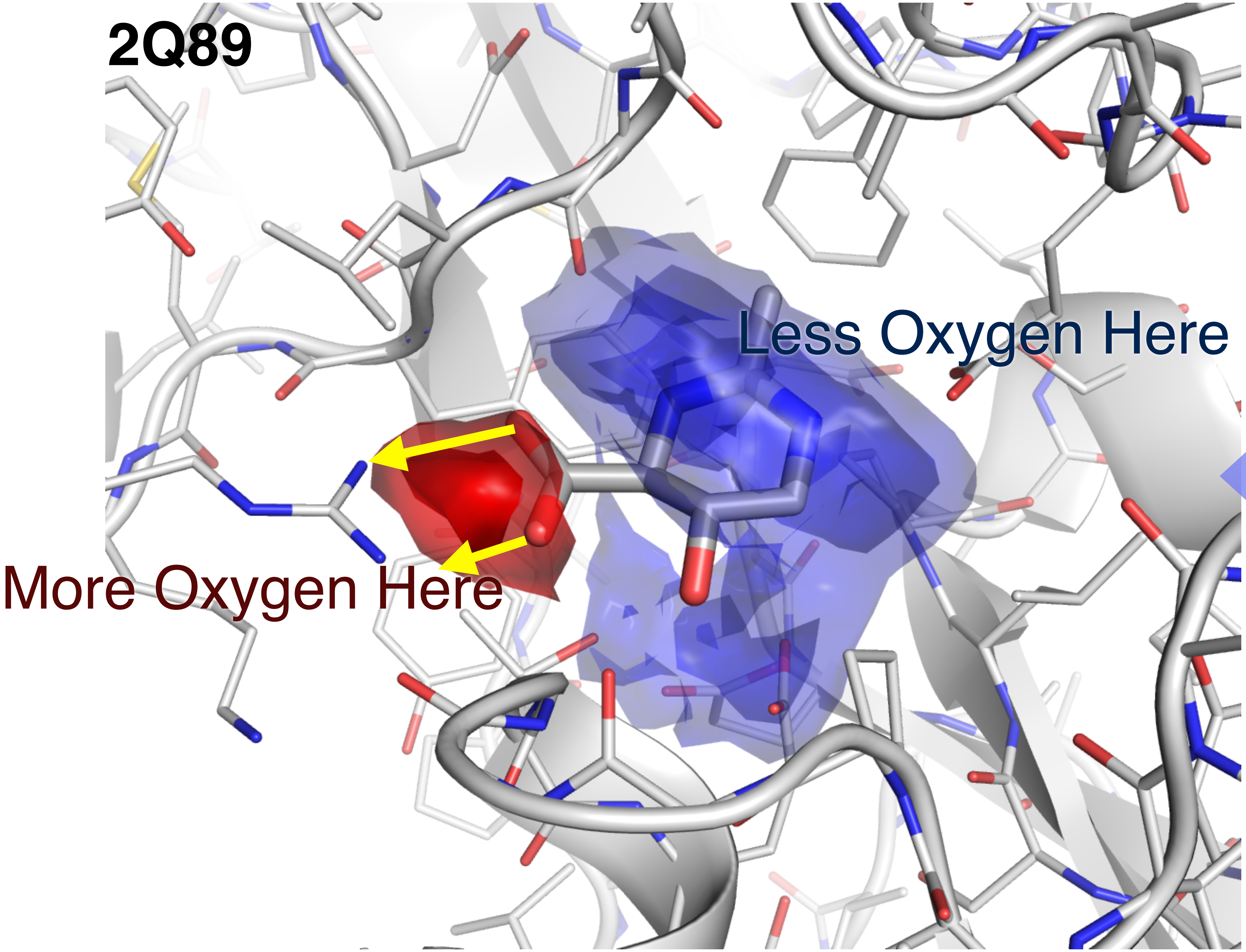
https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

Beyond Scoring

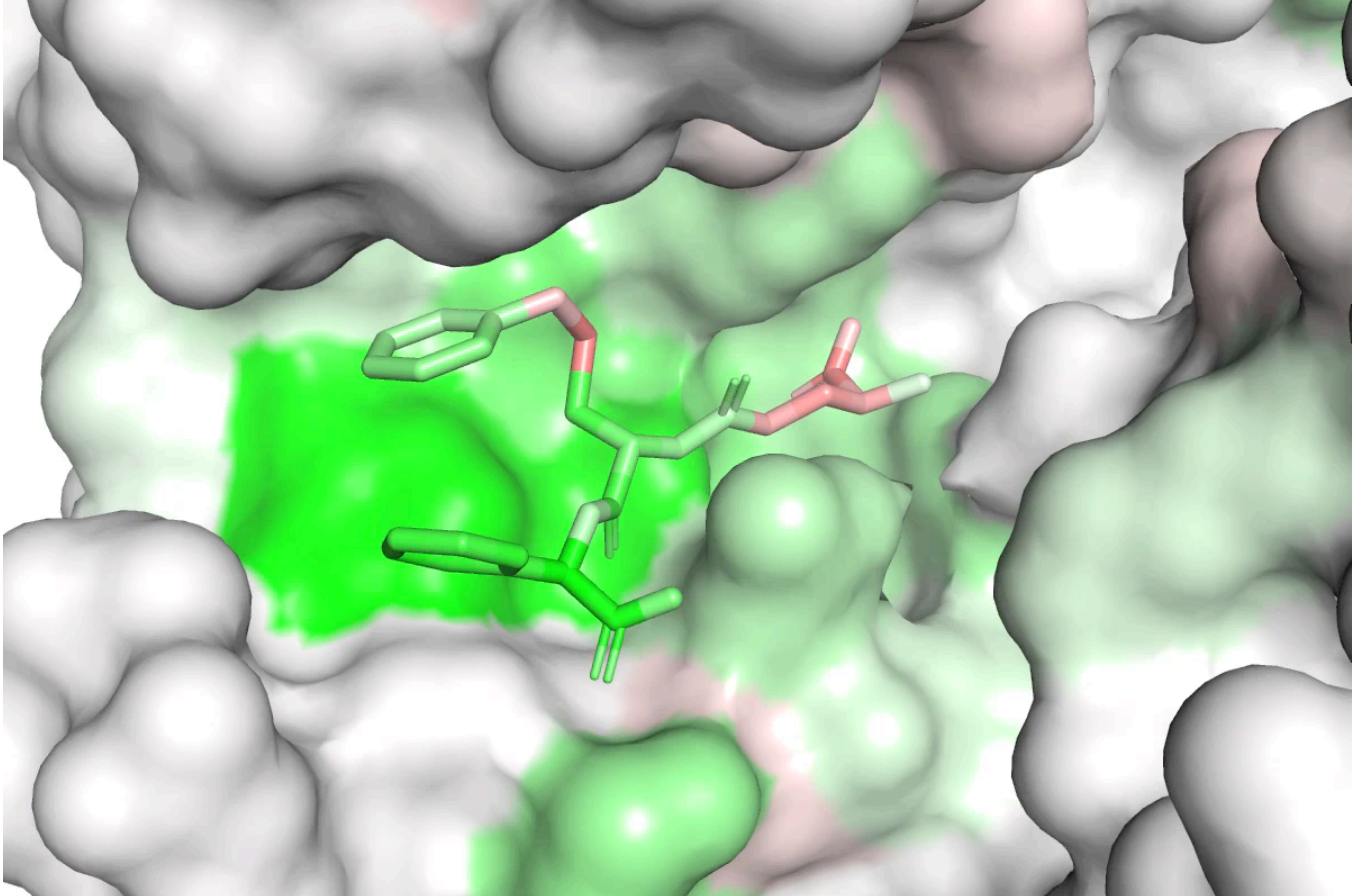


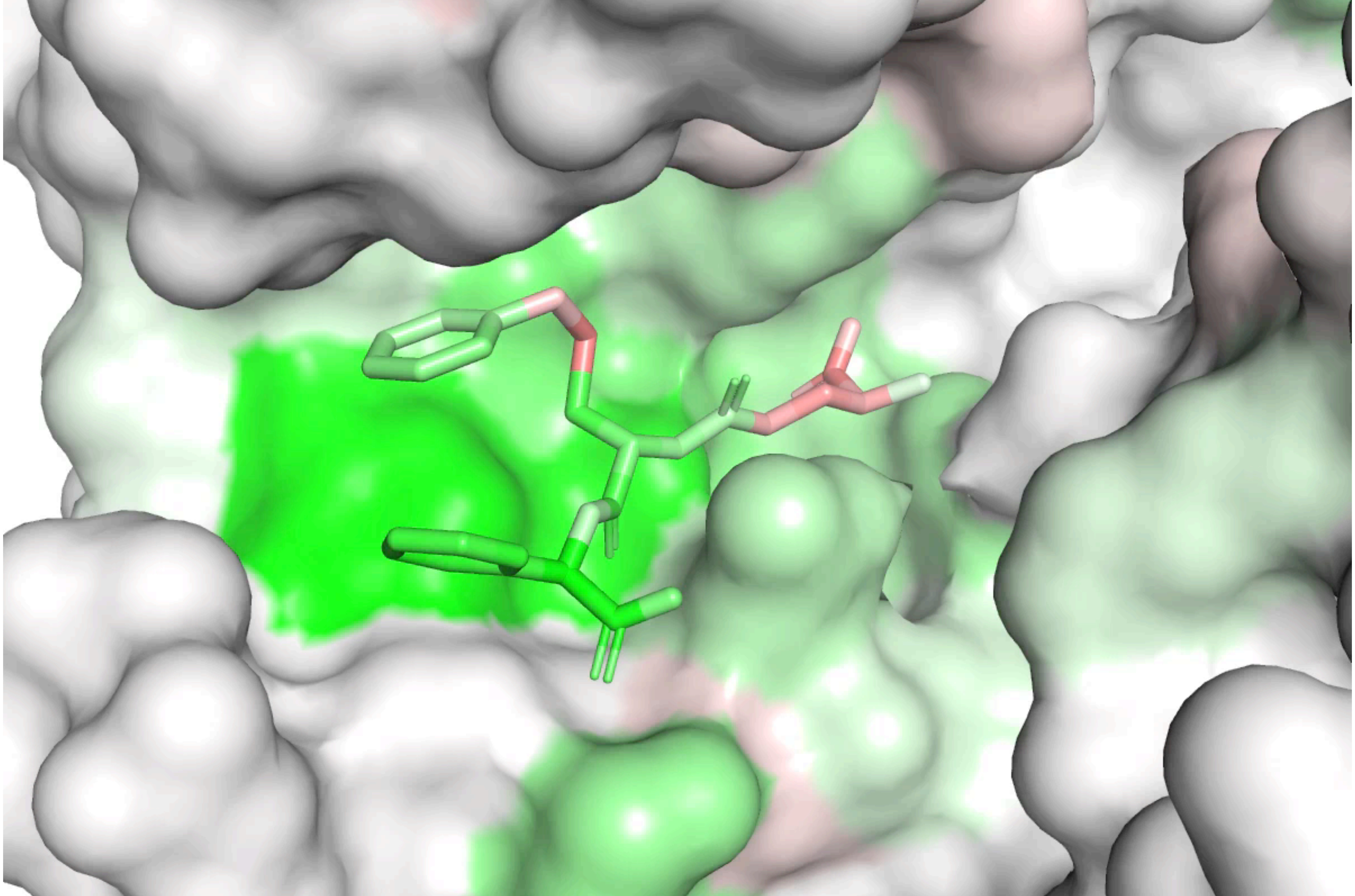
$$\frac{\partial L}{\partial A} = \sum_{i \in G_A} \frac{\partial L}{\partial G_i} \frac{\partial G_i}{\partial D} \frac{\partial D}{\partial A}$$

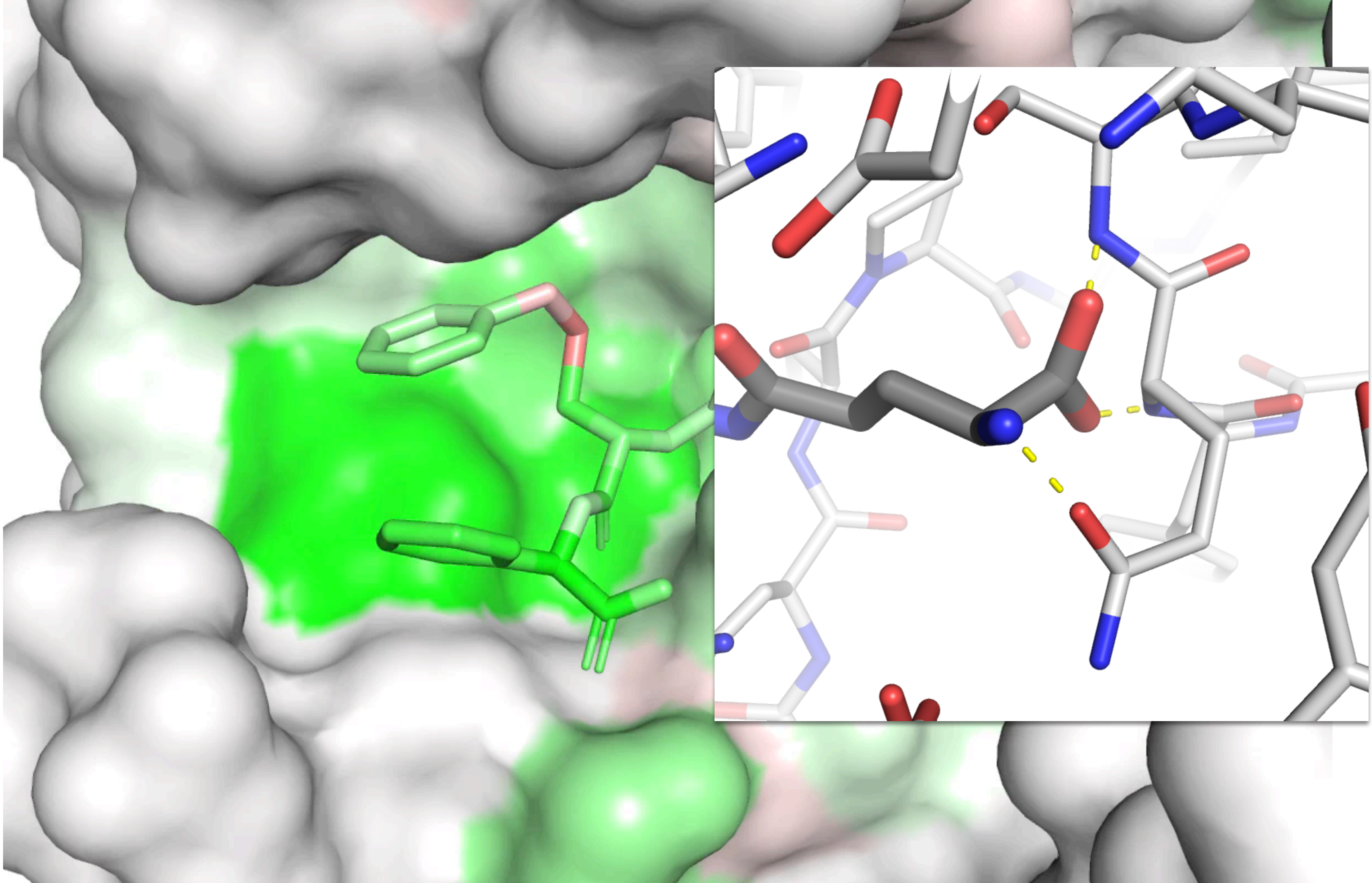
Beyond Scoring

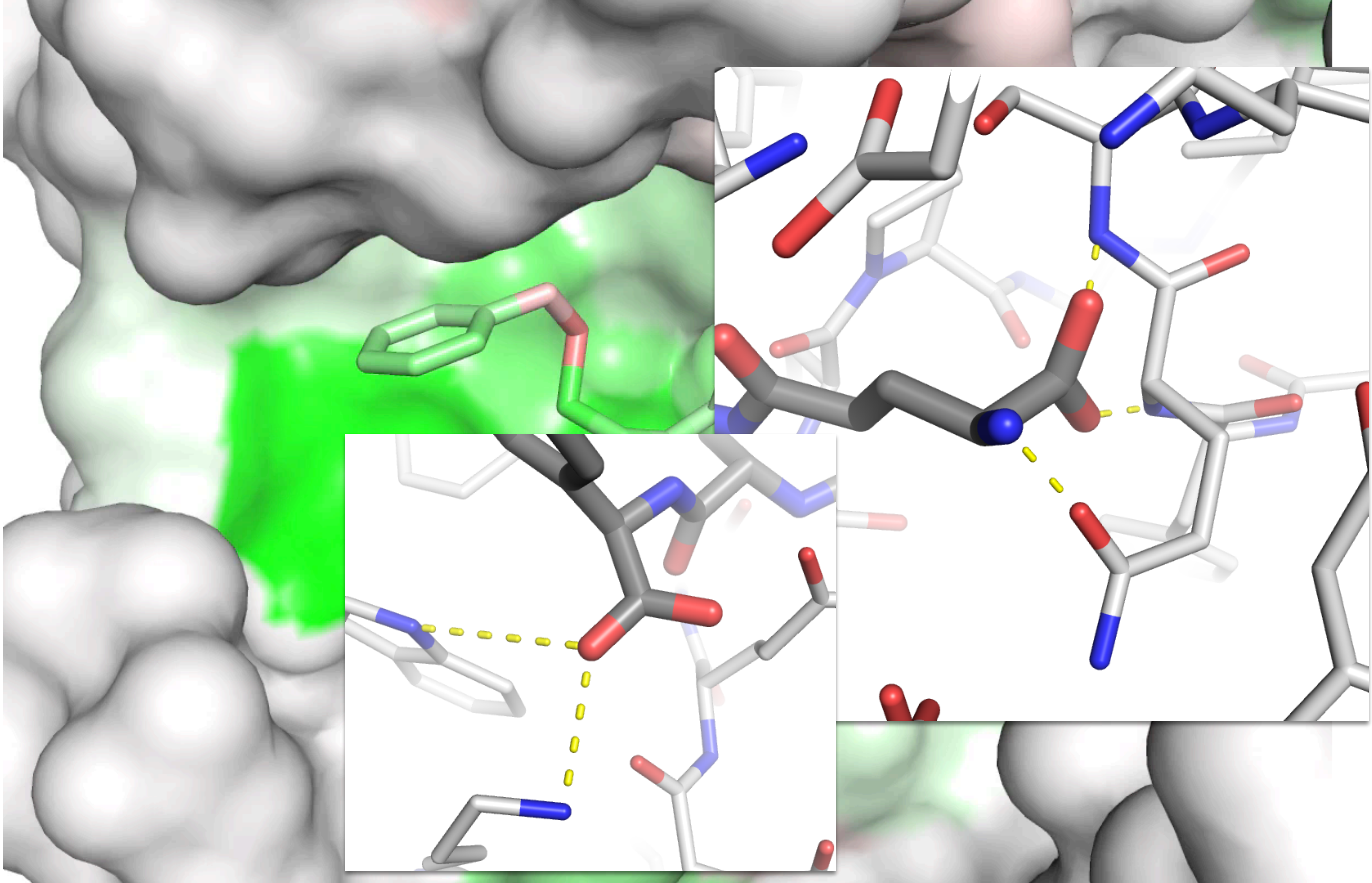


$$\frac{\partial L}{\partial A} = \sum_{i \in G_A} \frac{\partial L}{\partial G_i} \frac{\partial G_i}{\partial D} \frac{\partial D}{\partial A}$$

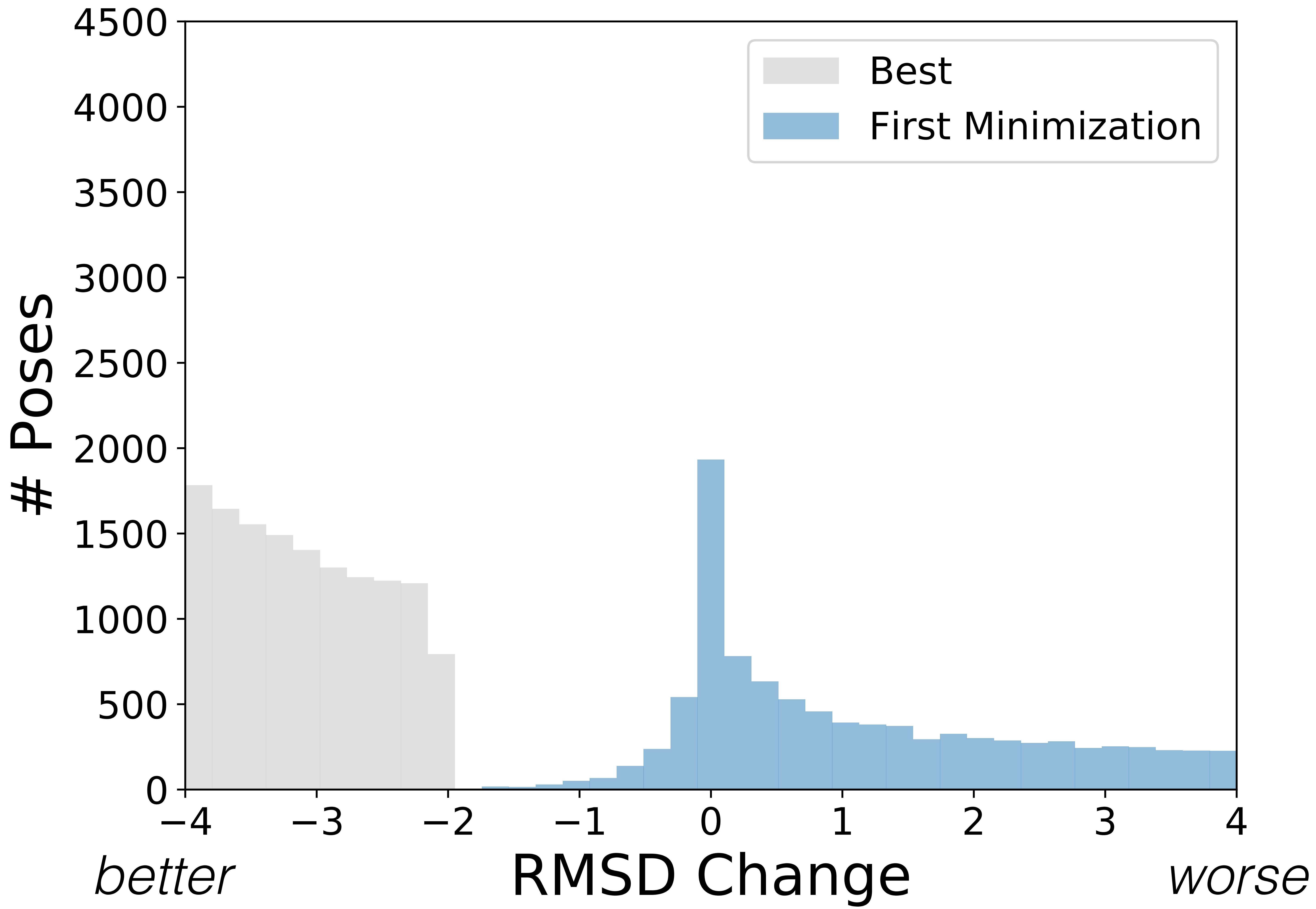




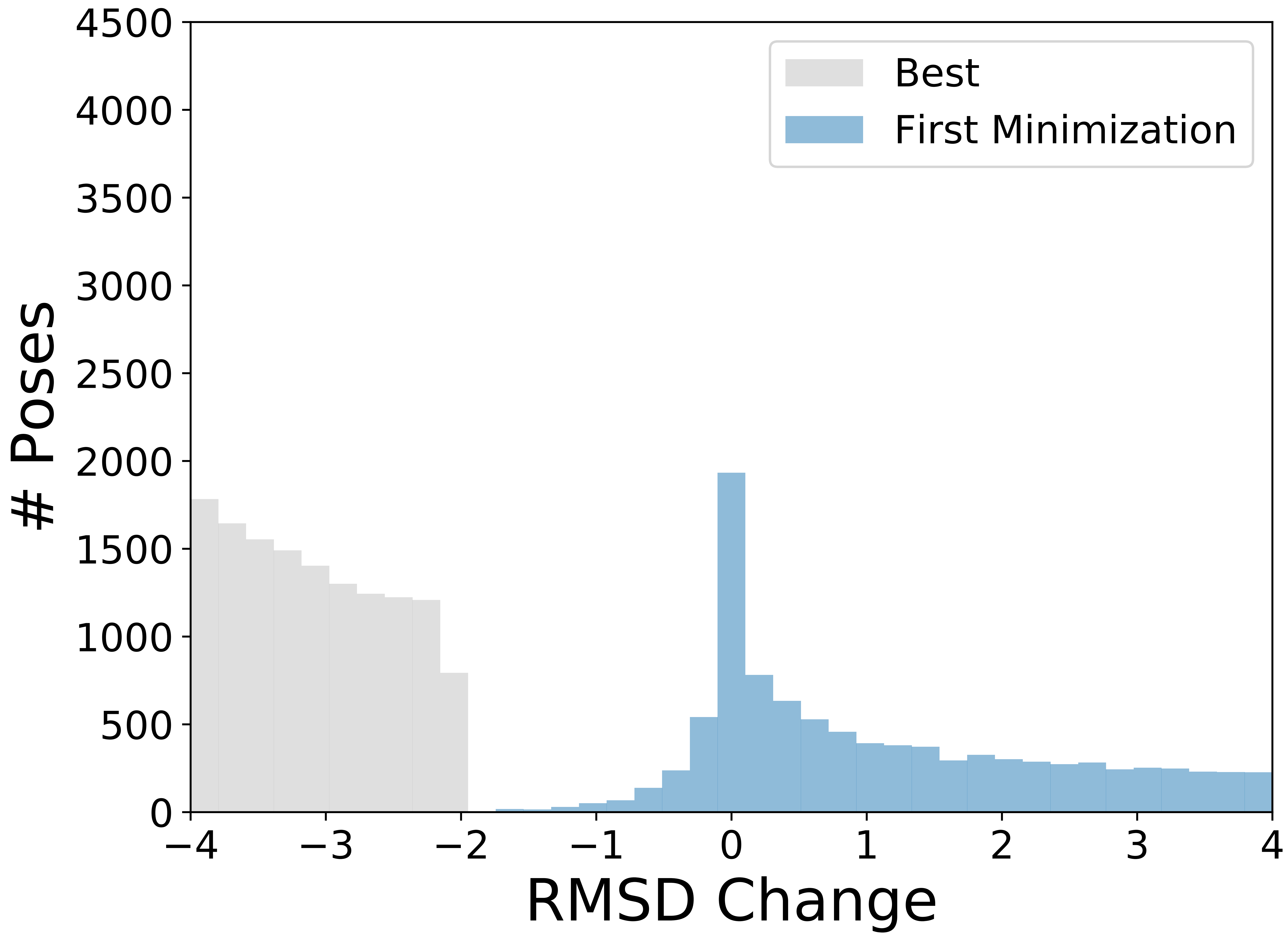




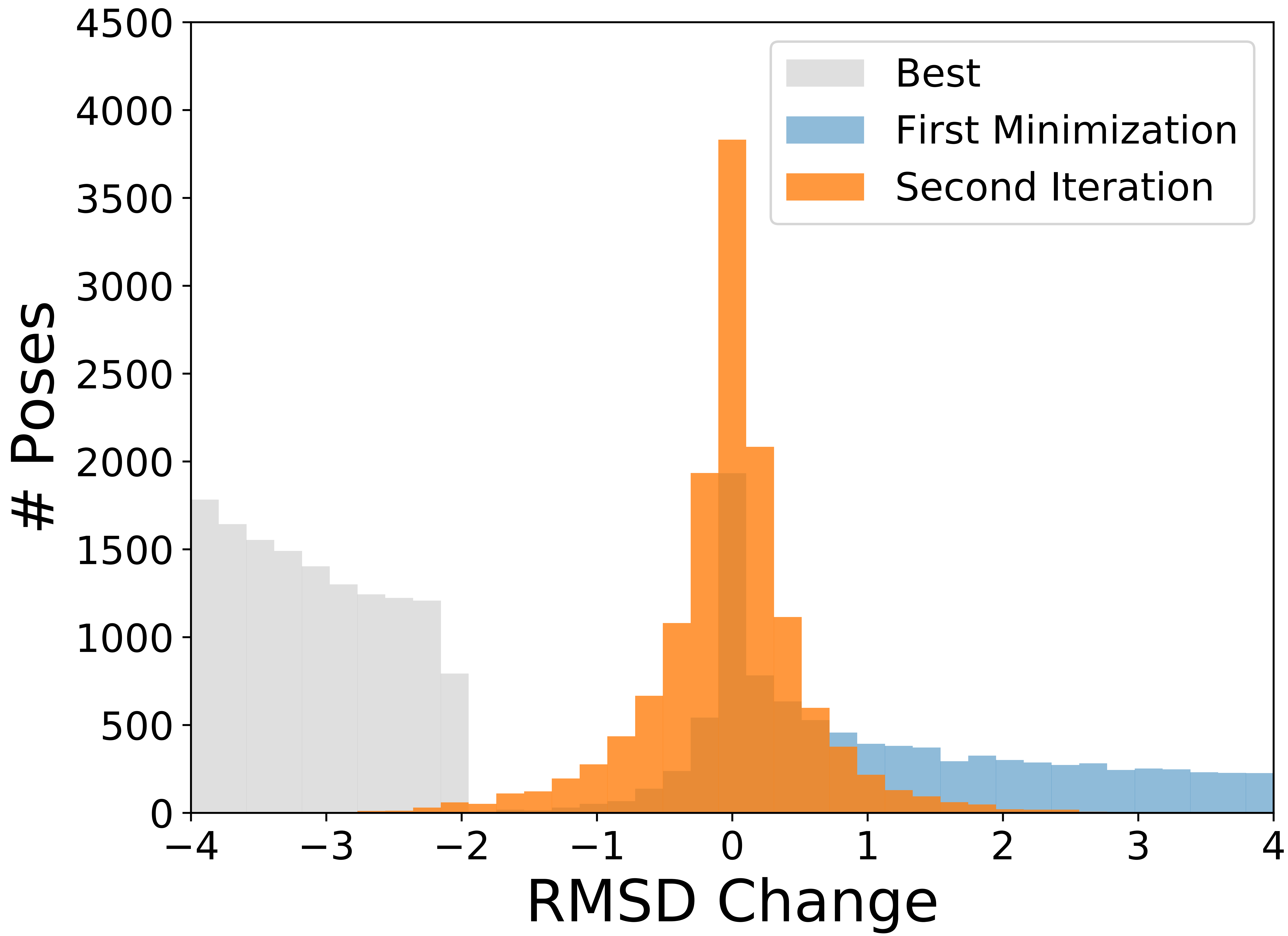
Minimizing Low RMSD Poses



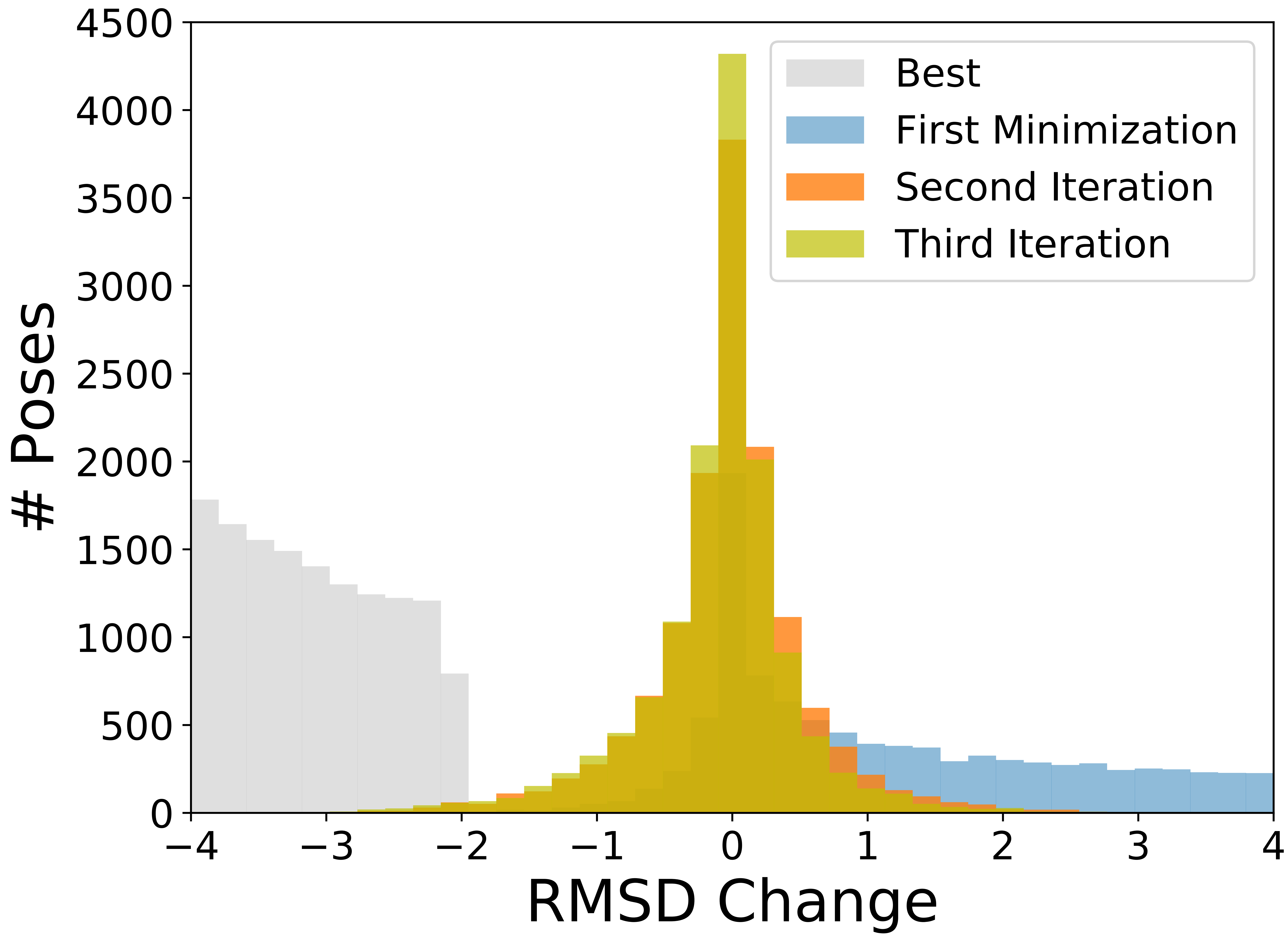
Iterative Refinement



Iterative Refinement

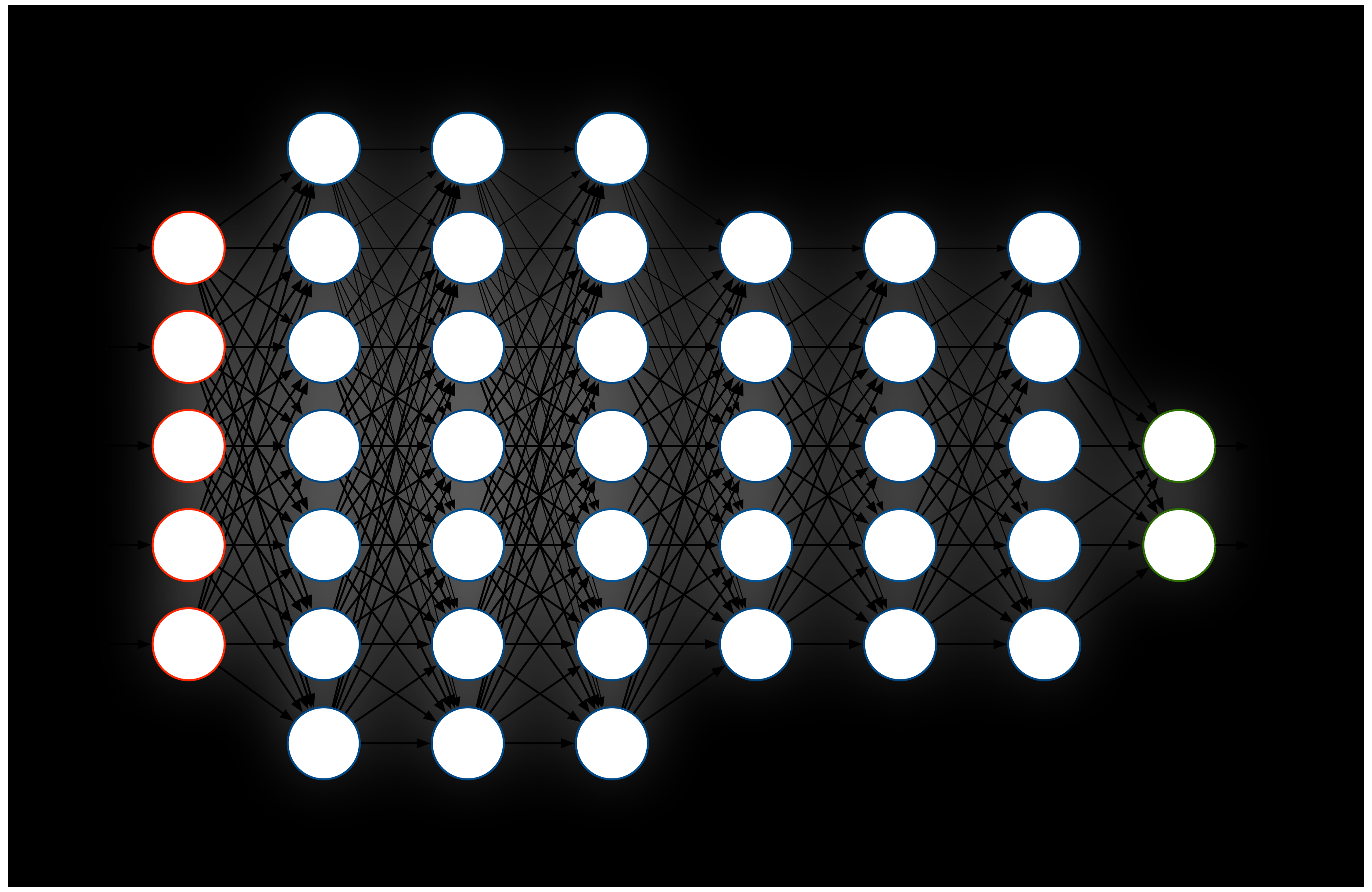


Iterative Refinement



Generative Modeling

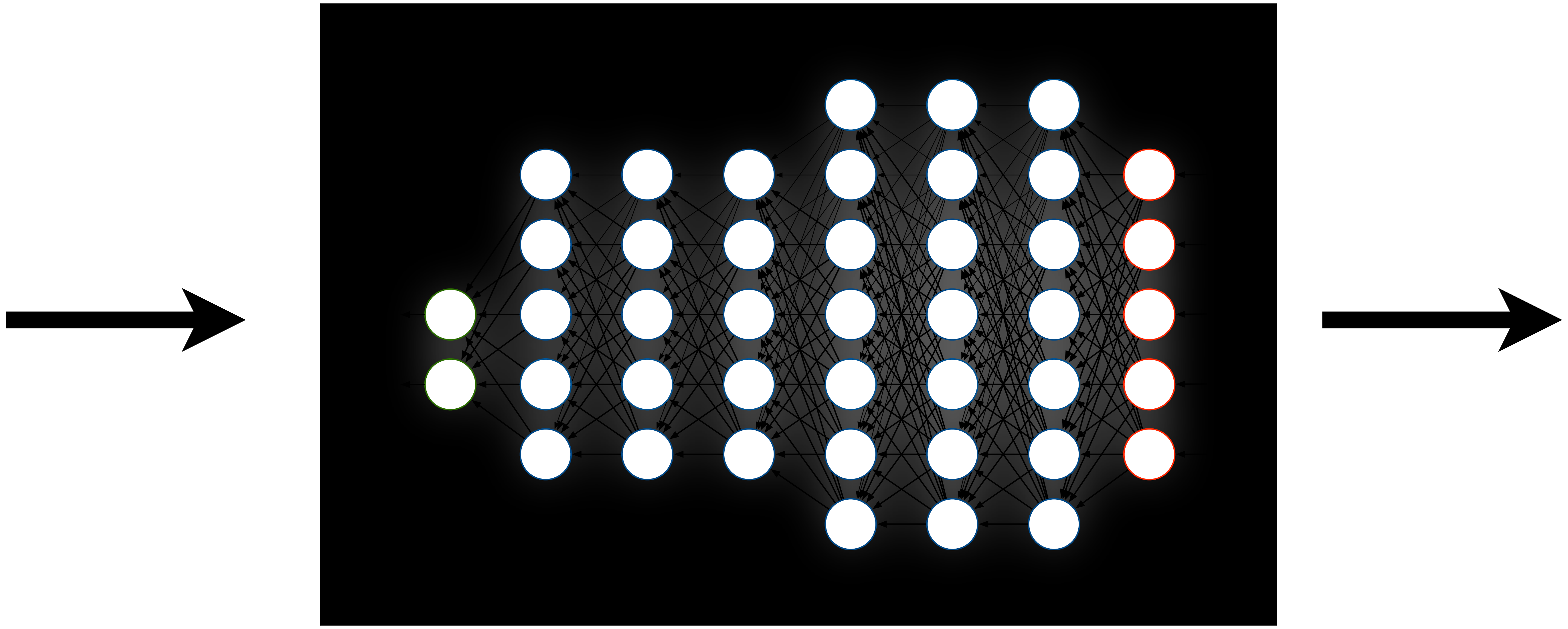
Discriminative Model

Features $\mathbf{X}$ $\longrightarrow$  $\longrightarrow$ Prediction $\mathbf{y}$

Generative Model

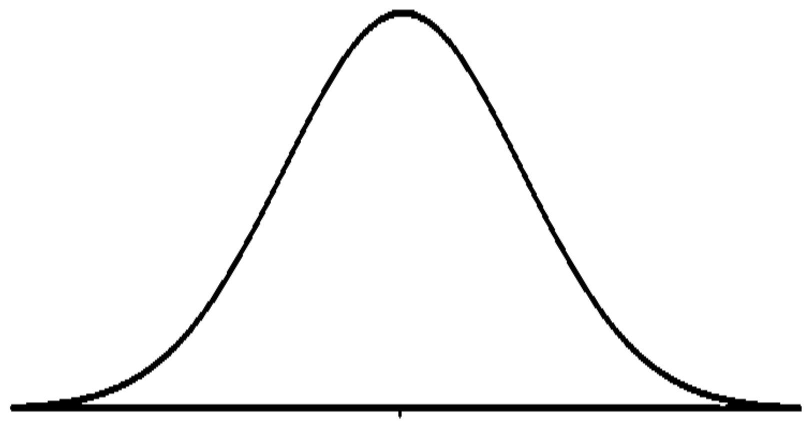
Features $\mathbf{X}$

Generative Model

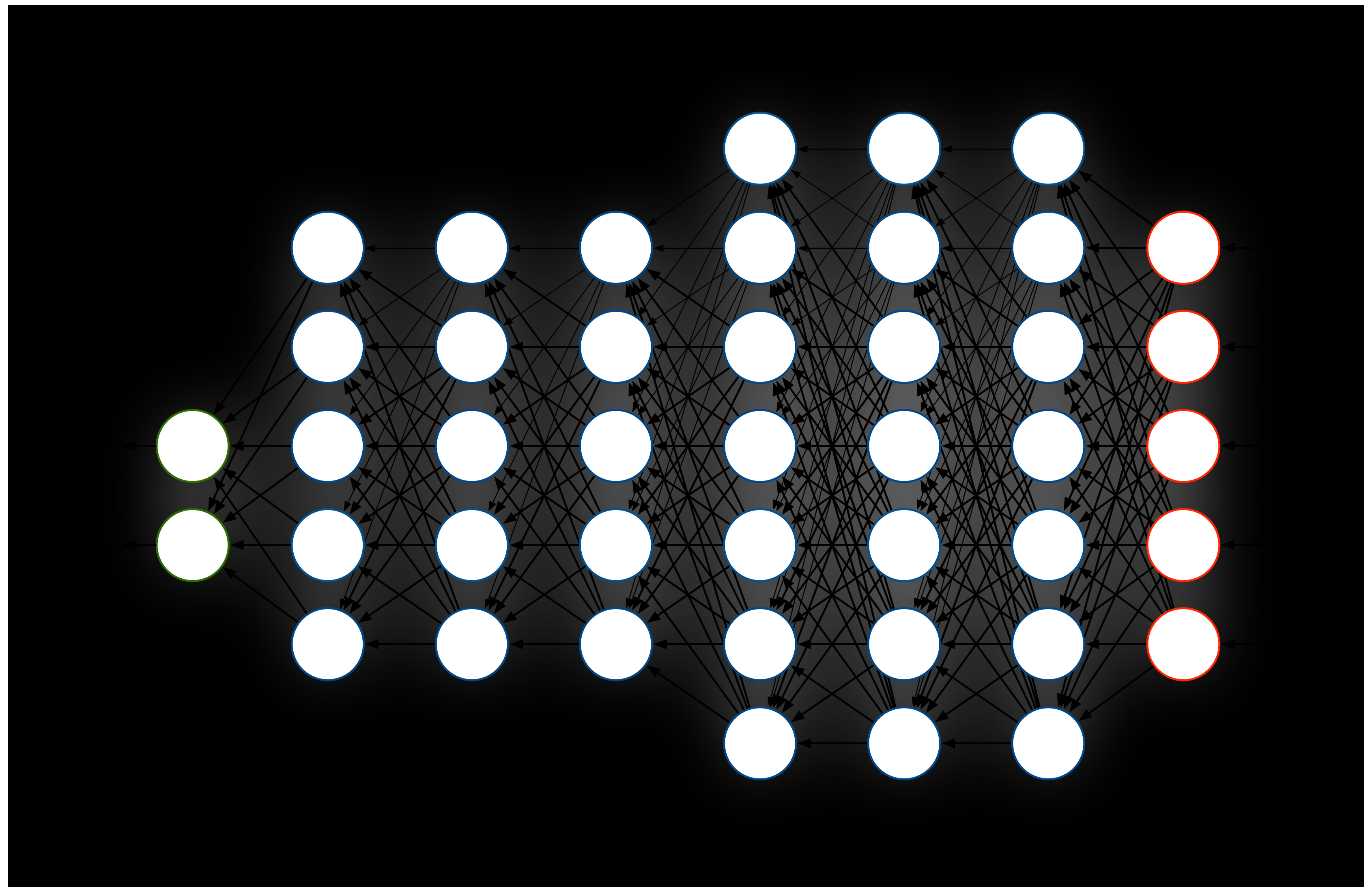


Features $\mathbf{X}$

Generative Model

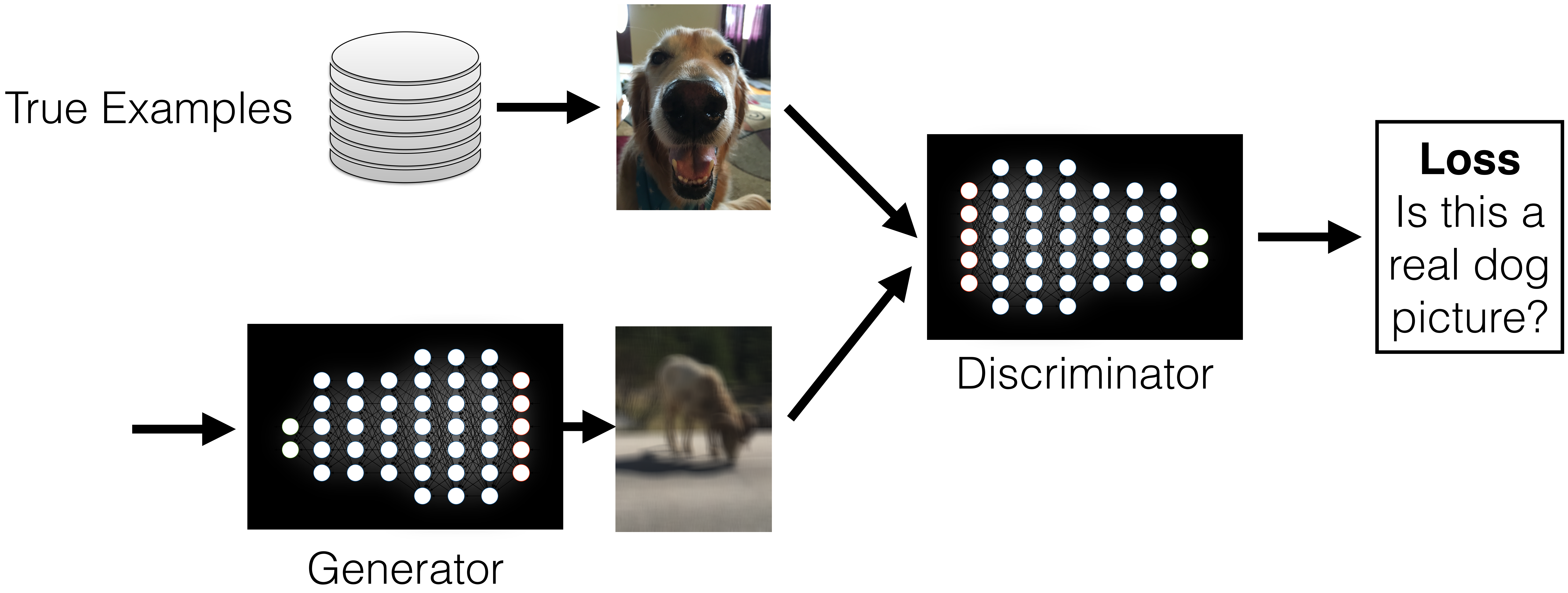


$\mathbf{y}$?

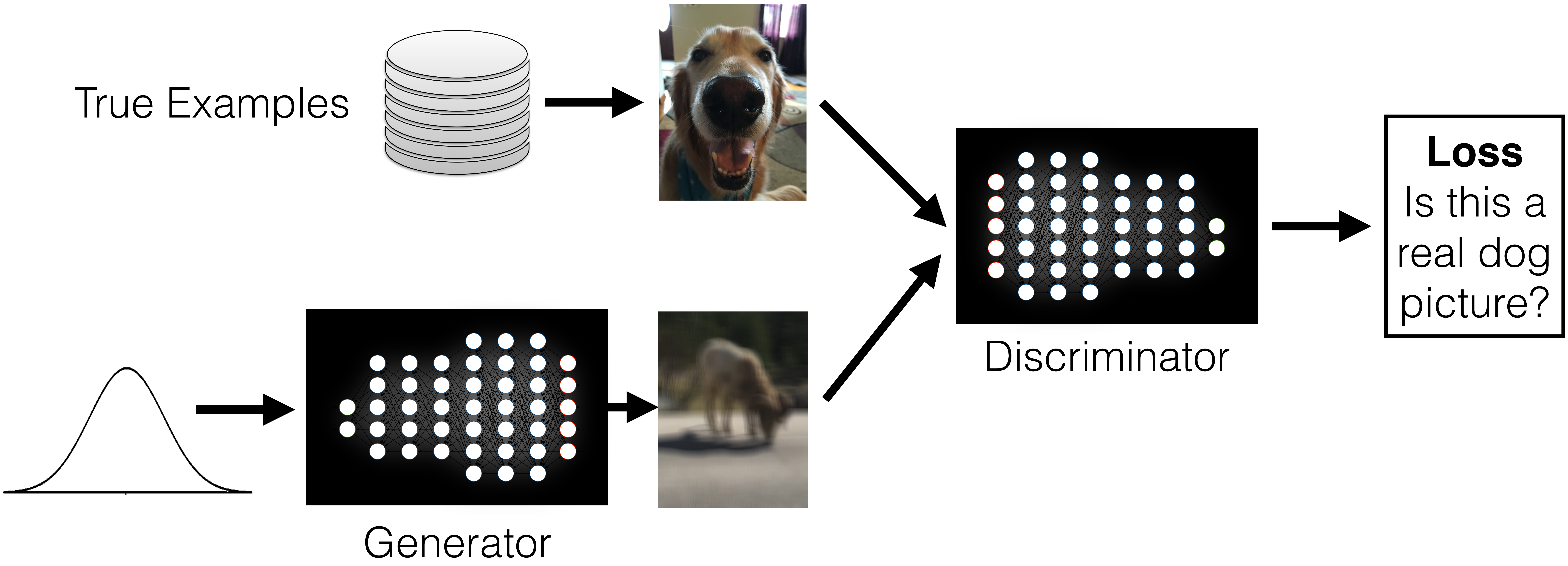


Features $\mathbf{X}$

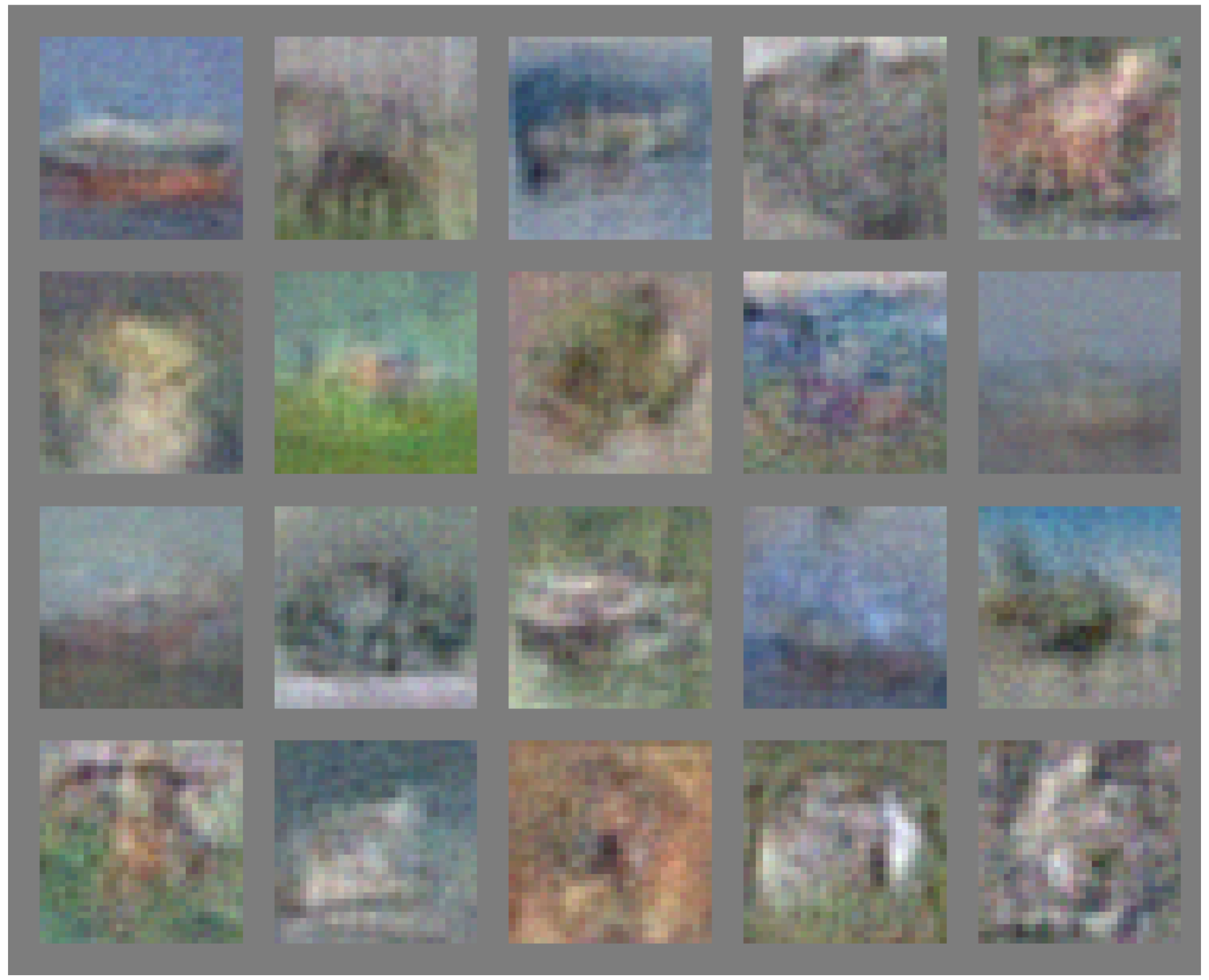
Generative Adversarial Networks



Generative Adversarial Networks



Generative Adversarial Networks



Generative Adversarial Networks
https://arxiv.org › stat
by IJ Goodfellow - 2014 - Cited by 4339 - Related articles
Jun 10, 2014 - Submission history. From: Ian **Goodfellow** [view email] [v1] Tue, 10 Jun **2014** 18:58:17 GMT (1257kb,D). Which authors of this paper are ...

Generative Adversarial Networks

Generative Adversarial Networks
https://arxiv.org › stat ▾
by IJ Goodfellow - 2014 - Cited by 4339 - Related articles
Jun 10, 2014 - Submission history. From: Ian **Goodfellow** [view email] [v1] Tue, 10 Jun **2014** 18:58:17 GMT (1257kb,D). Which authors of this paper are ...

http://torch.ch/blog/2015/11/13/gan.html

Progressive Growing of GANs for Improved Quality, Stability, and Variation

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

Jaakko Lehtinen
NVIDIA
Aalto University

NVIDIA

https://youtu.be/G06dEcZ-QTg

Progressive Growing of GANs for Improved Quality, Stability, and Variation

Tero Karras
NVIDIA

Timo Aila
NVIDIA

Samuli Laine
NVIDIA

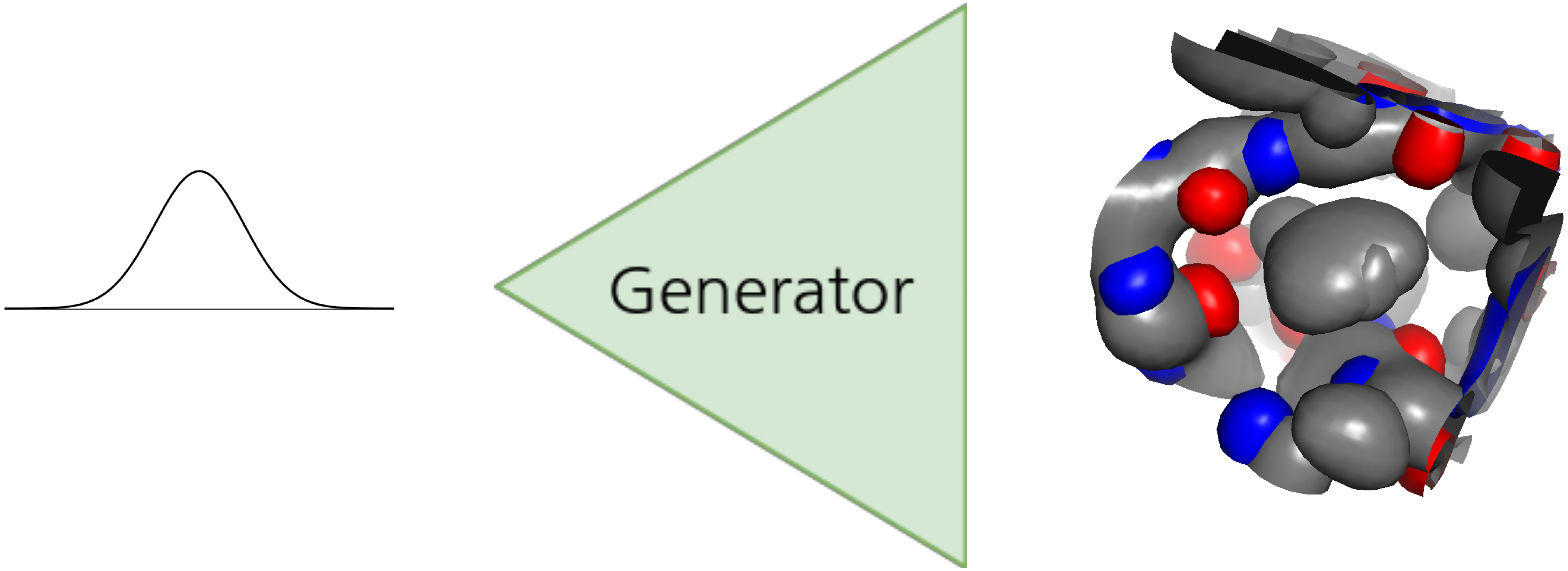
Jaakko Lehtinen
NVIDIA
Aalto University

NVIDIA

https://youtu.be/G06dEcZ-QTg

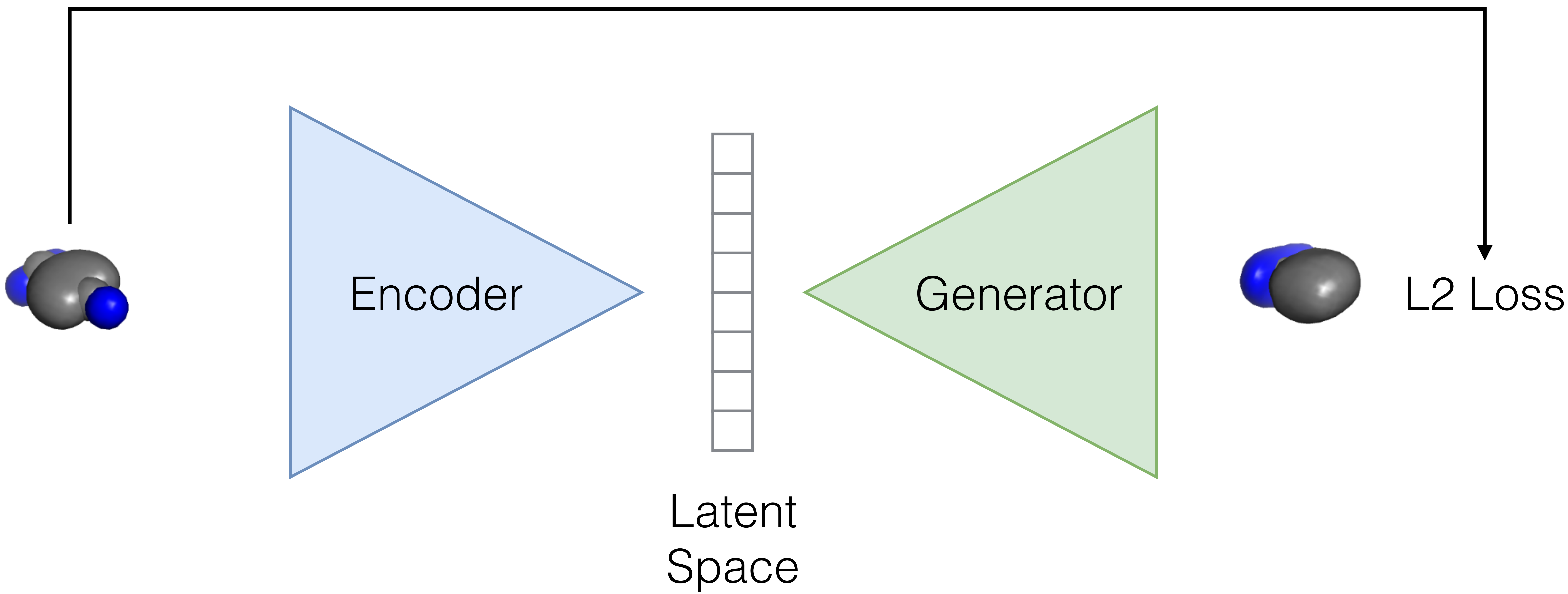
Generative Models

Generative models approximate a data distribution directly. They can map samples from one distribution (noise or input data) to realistic samples from an output distribution of interest.



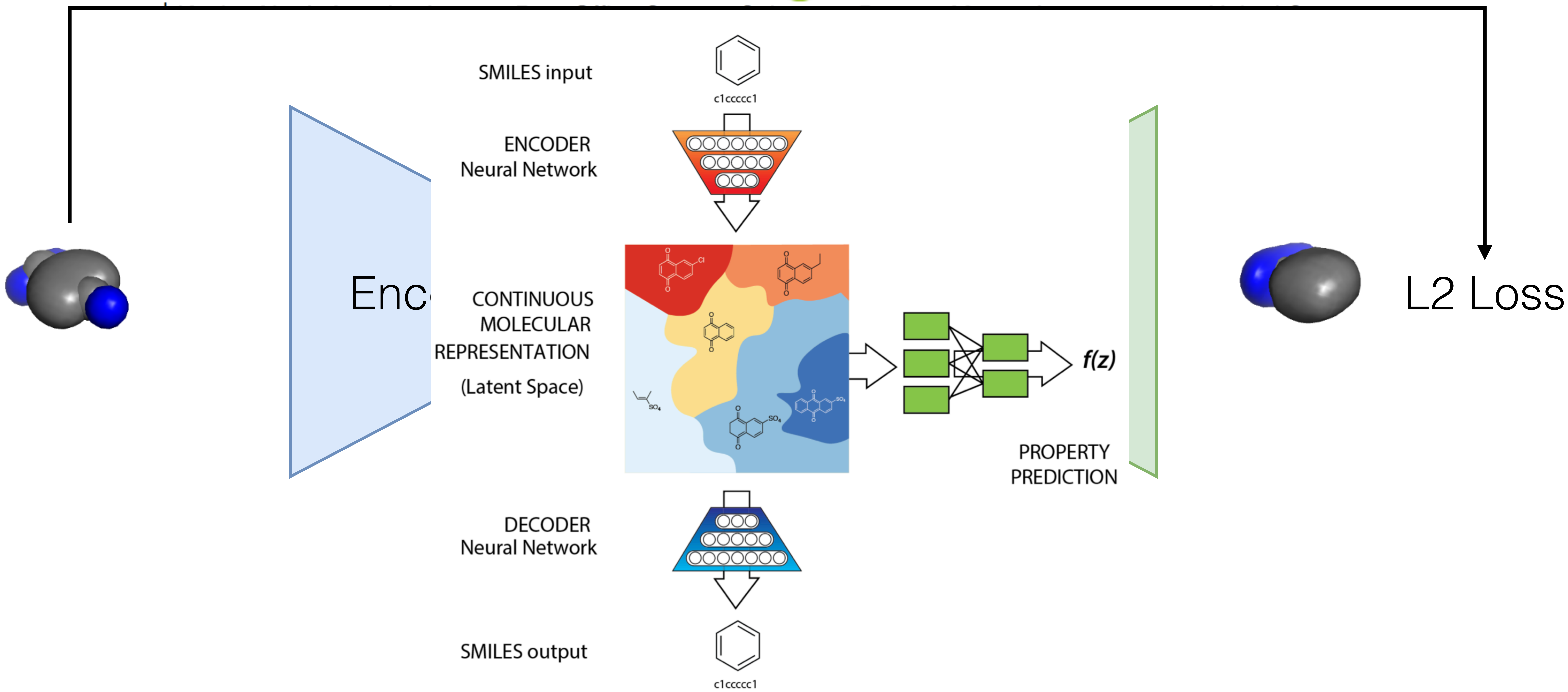
noise sample *generated receptor & ligand grid*

Autoencoding

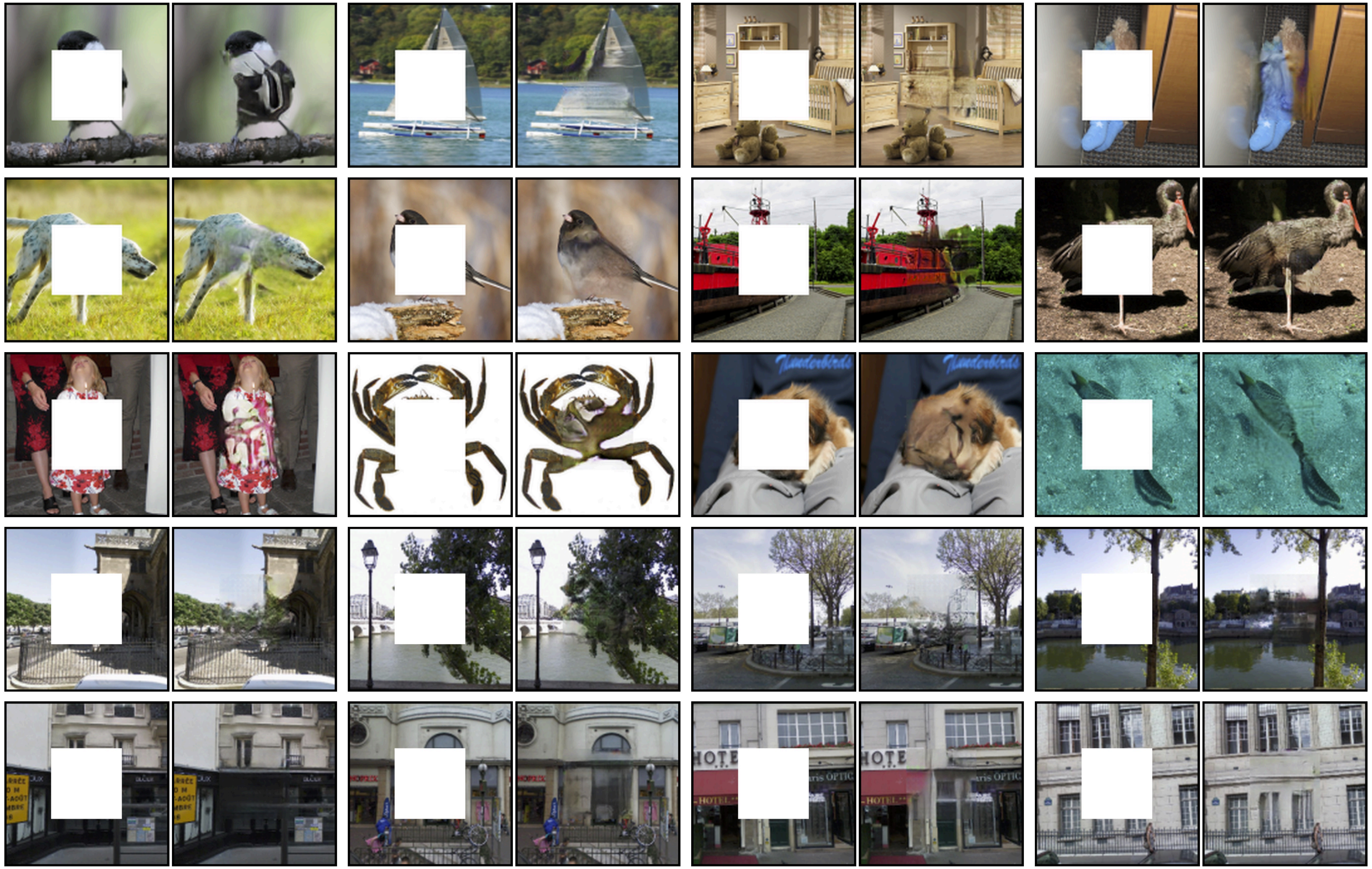


Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules

Rafael Gómez-Bombarelli†#, Jennifer N. Wei‡#, David Duvenaud¶#, José Miguel Hernández-Lobato§#, Benjamín Sánchez-Lengeling‡, Dennis Sheberla‡, Jorge Aguilera-Iparraguirre†, Timothy D. Hirzel†, Ryan P. Adams∇, and Alán Aspuru-Guzik*‡⊥

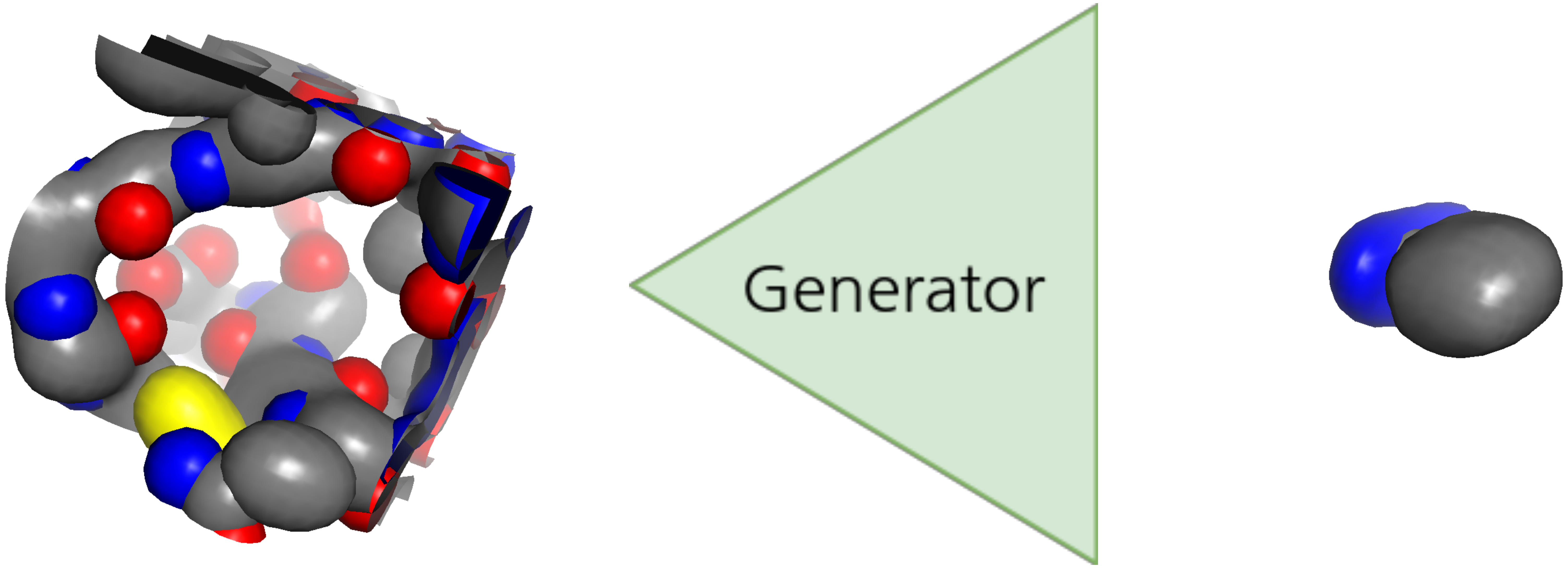


Context Encoding



http://people.eecs.berkeley.edu/~pathak/context_encoder/

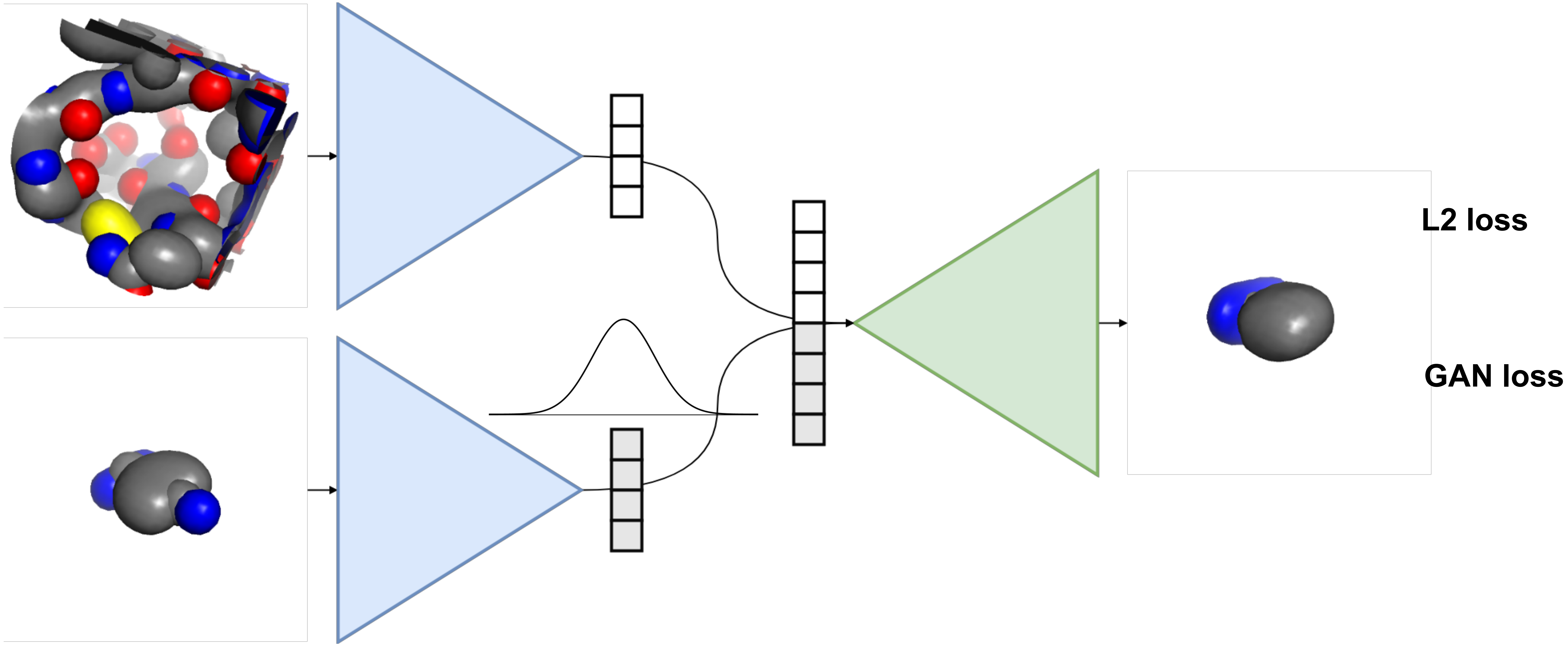
Context Encoding



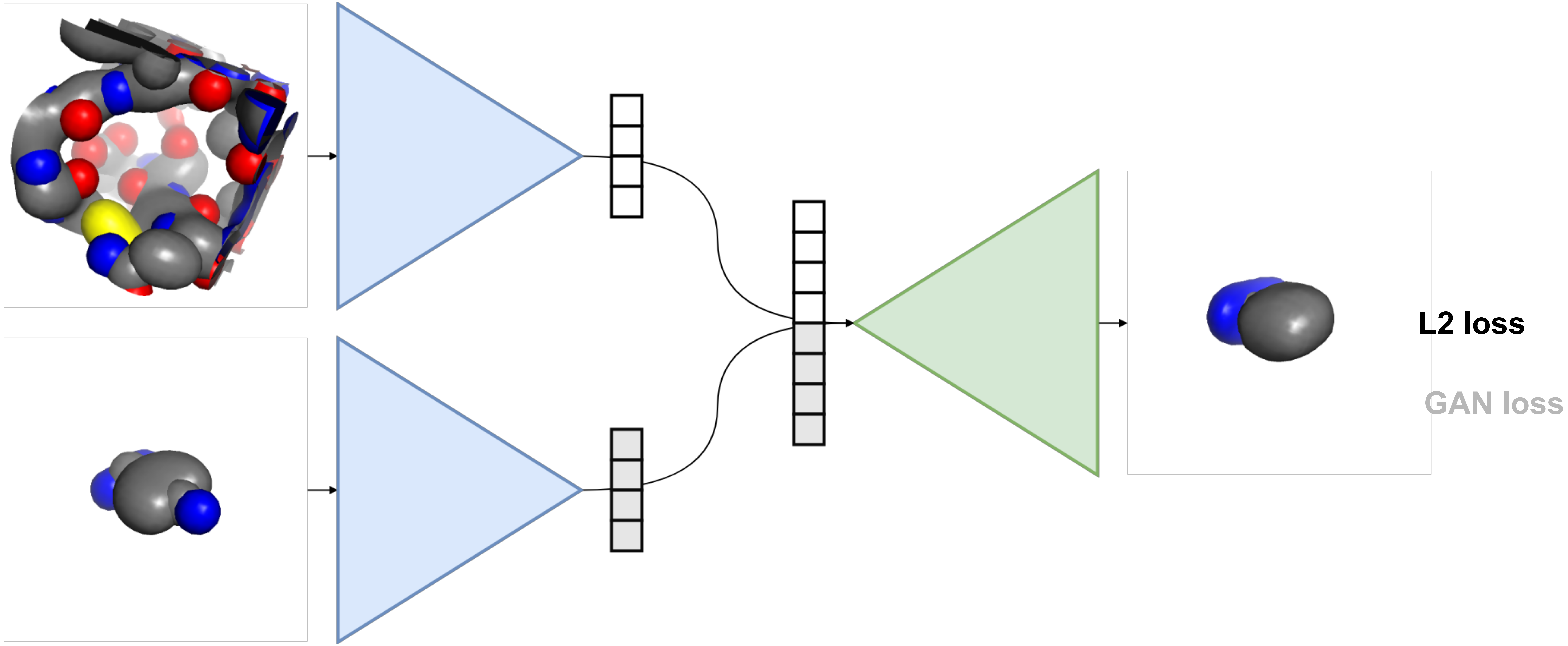
receptor grid

generated ligand grid

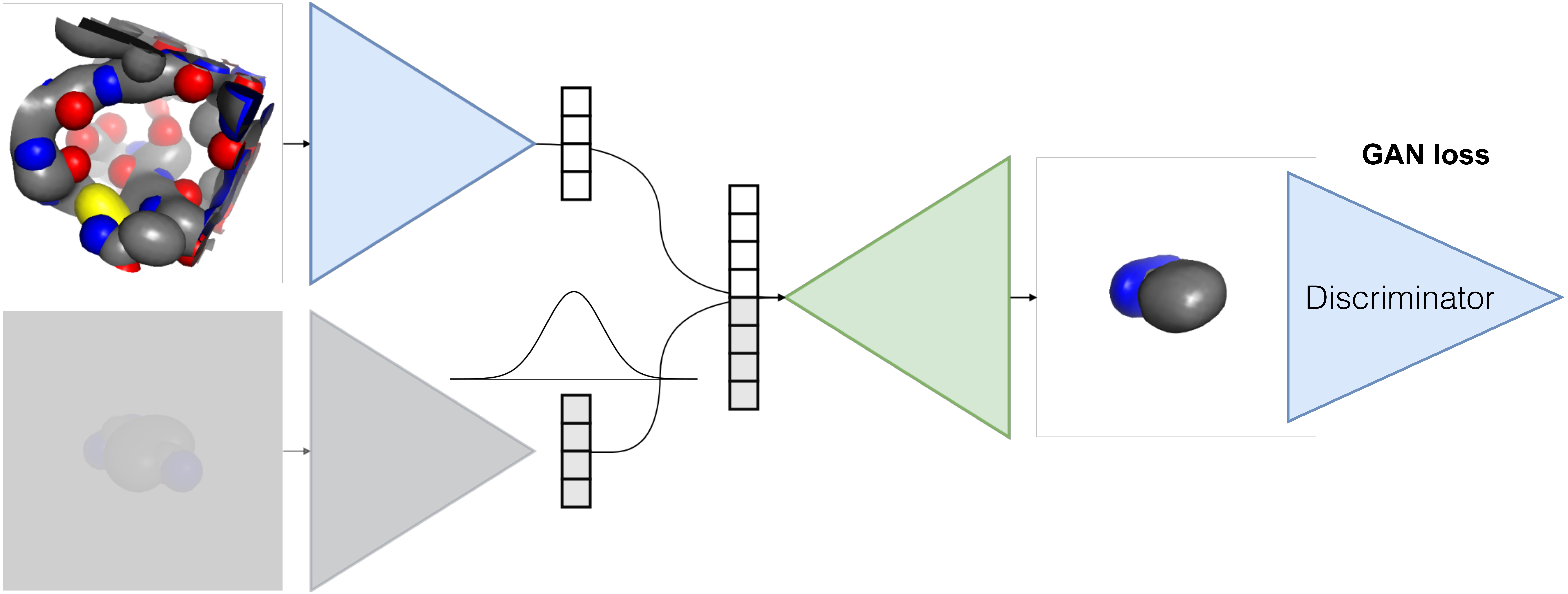
Receptor-Conditional Ligand-Variational Model



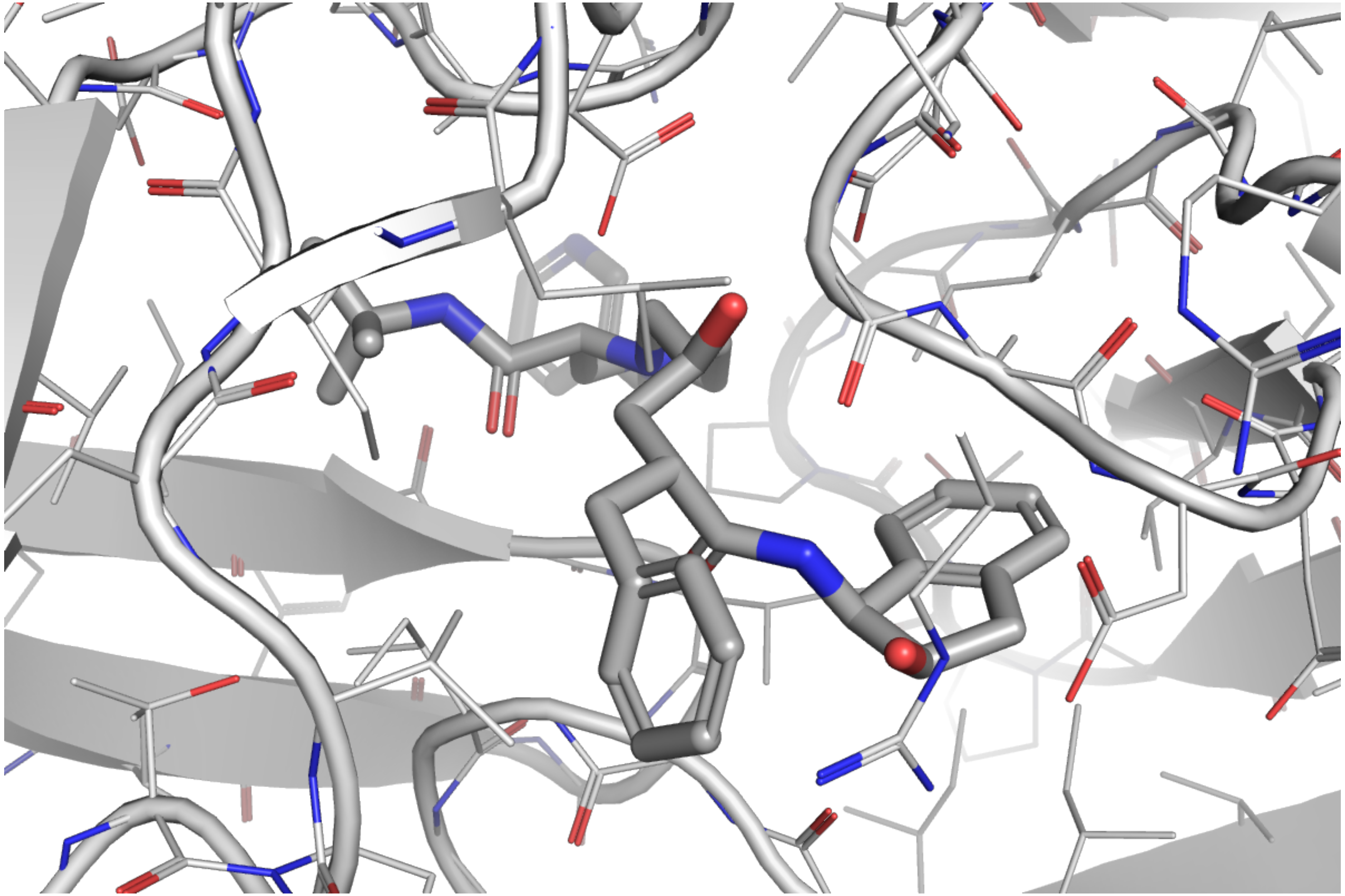
Receptor-Conditional Ligand-Variational Model



Receptor-Conditional Ligand-Variational Model

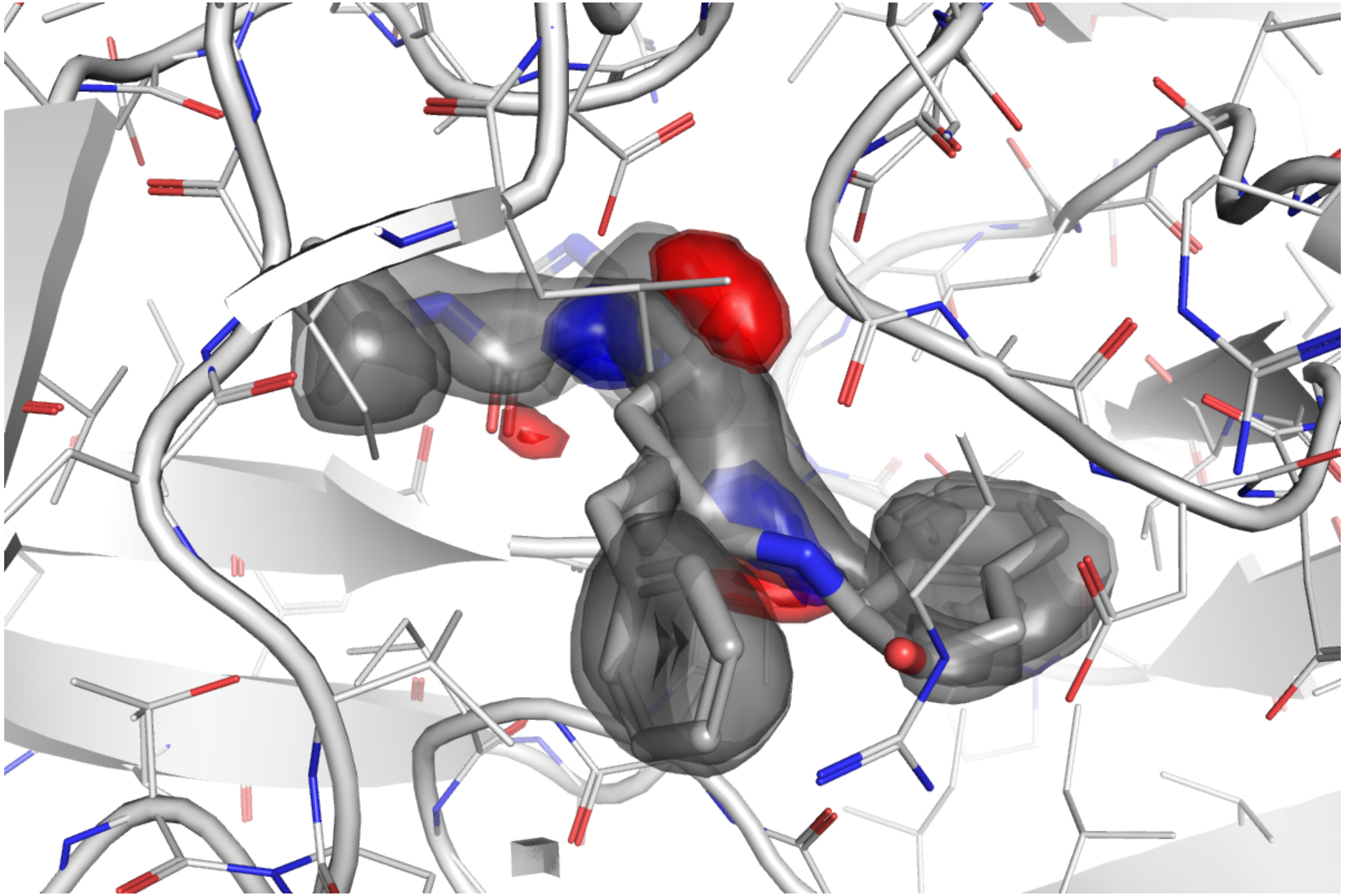


Autoencoding Examples



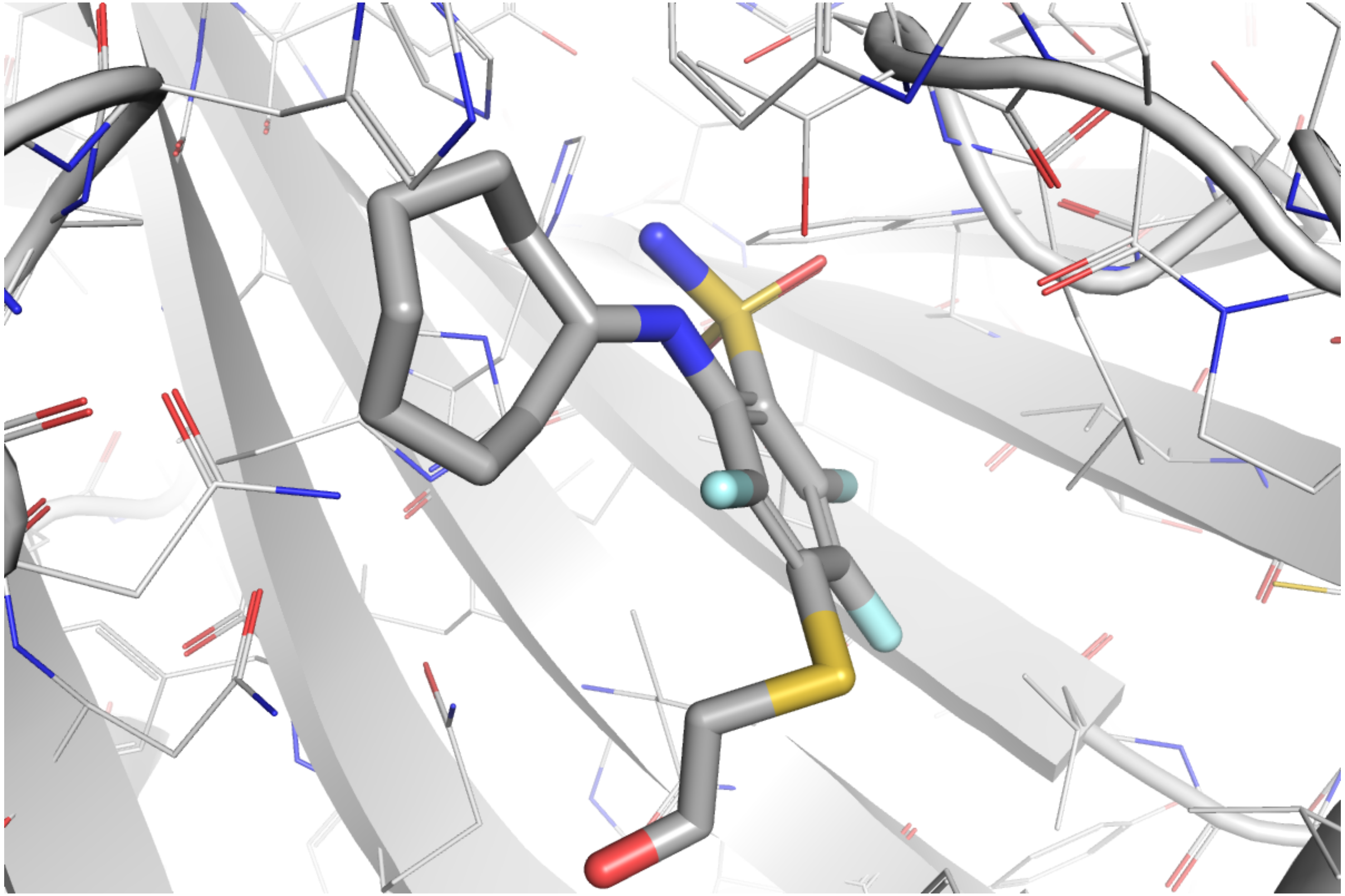
2AVO

Autoencoding Examples



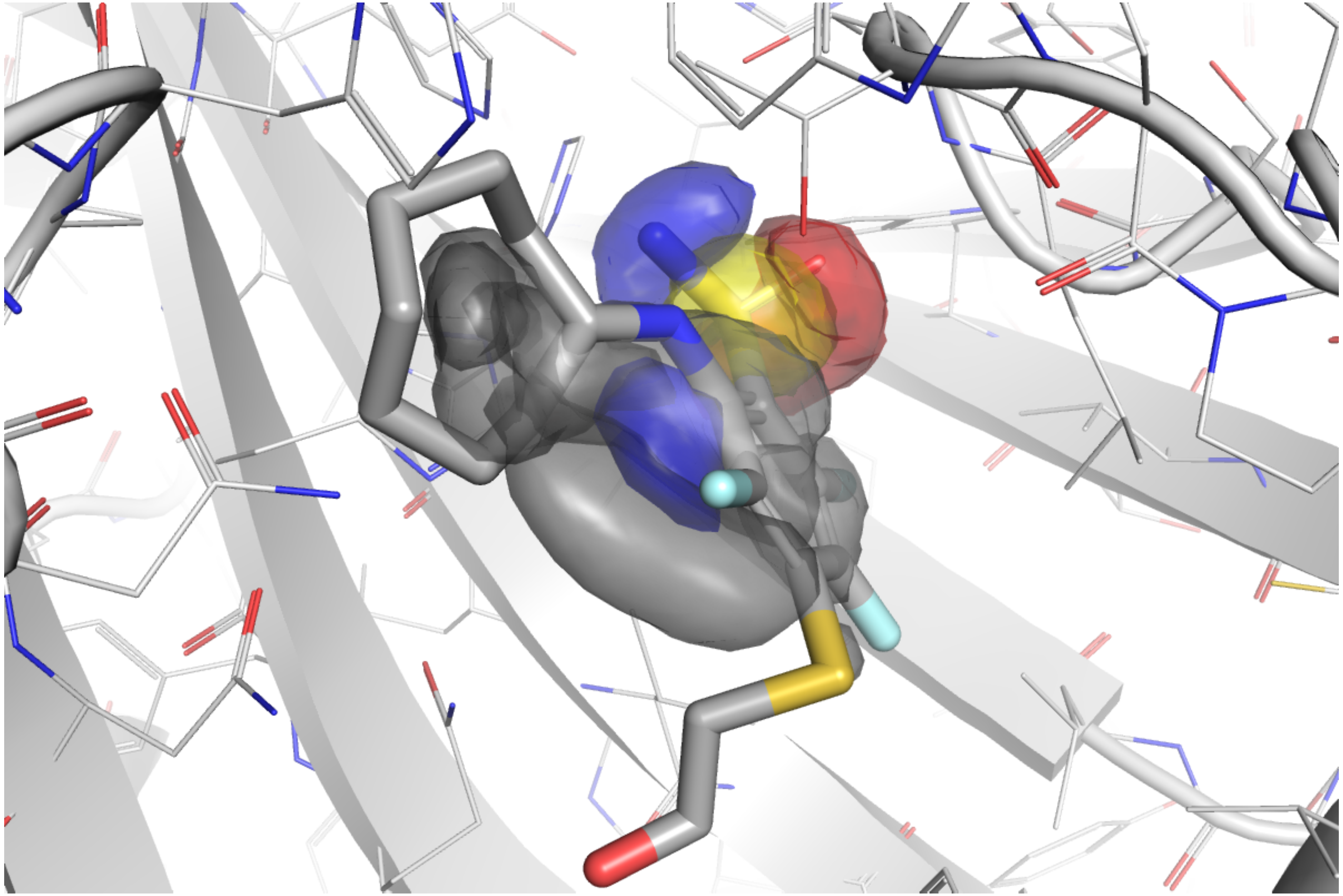
2AVO

Autoencoding Examples



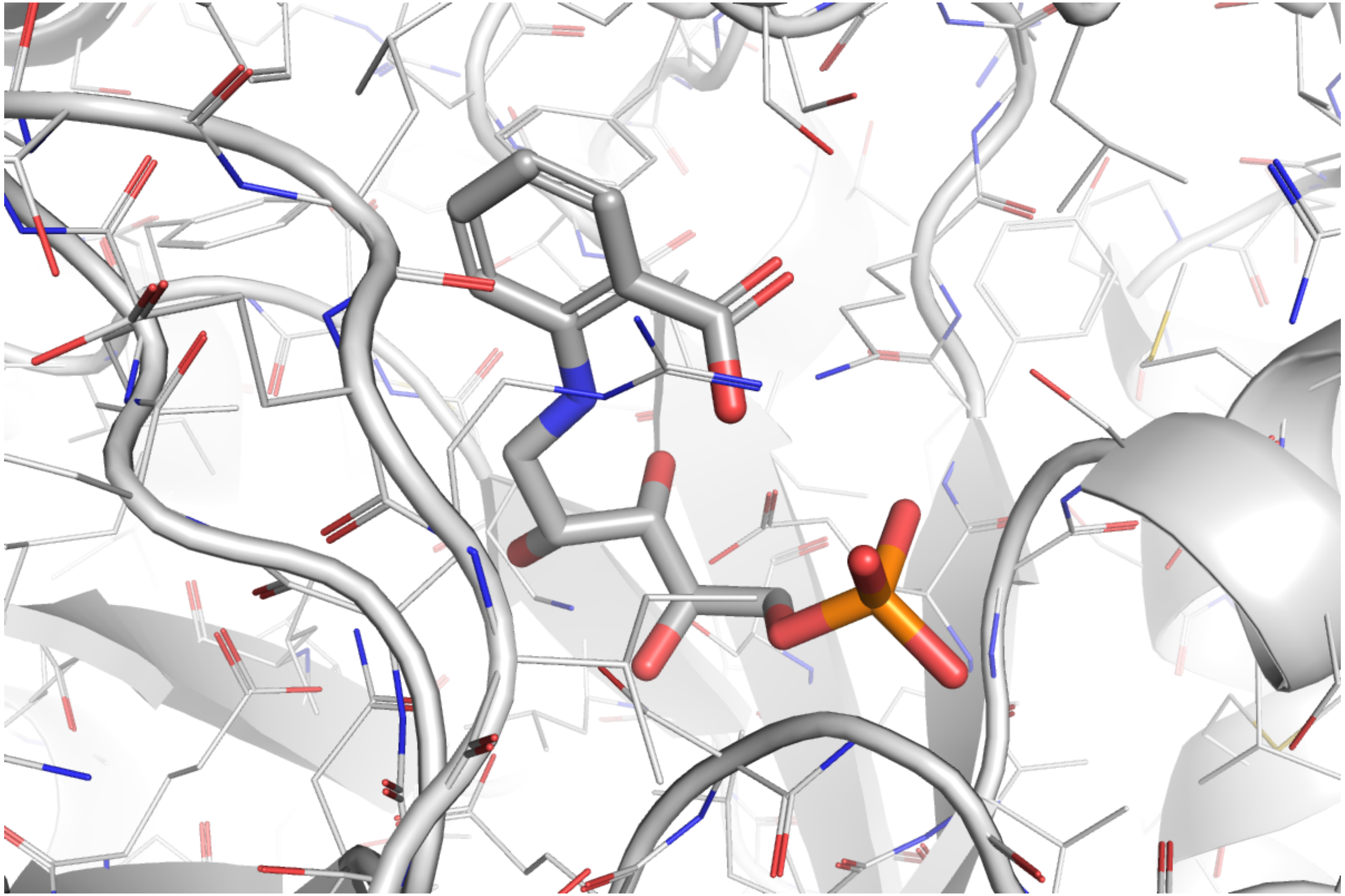
4PYX

Autoencoding Examples



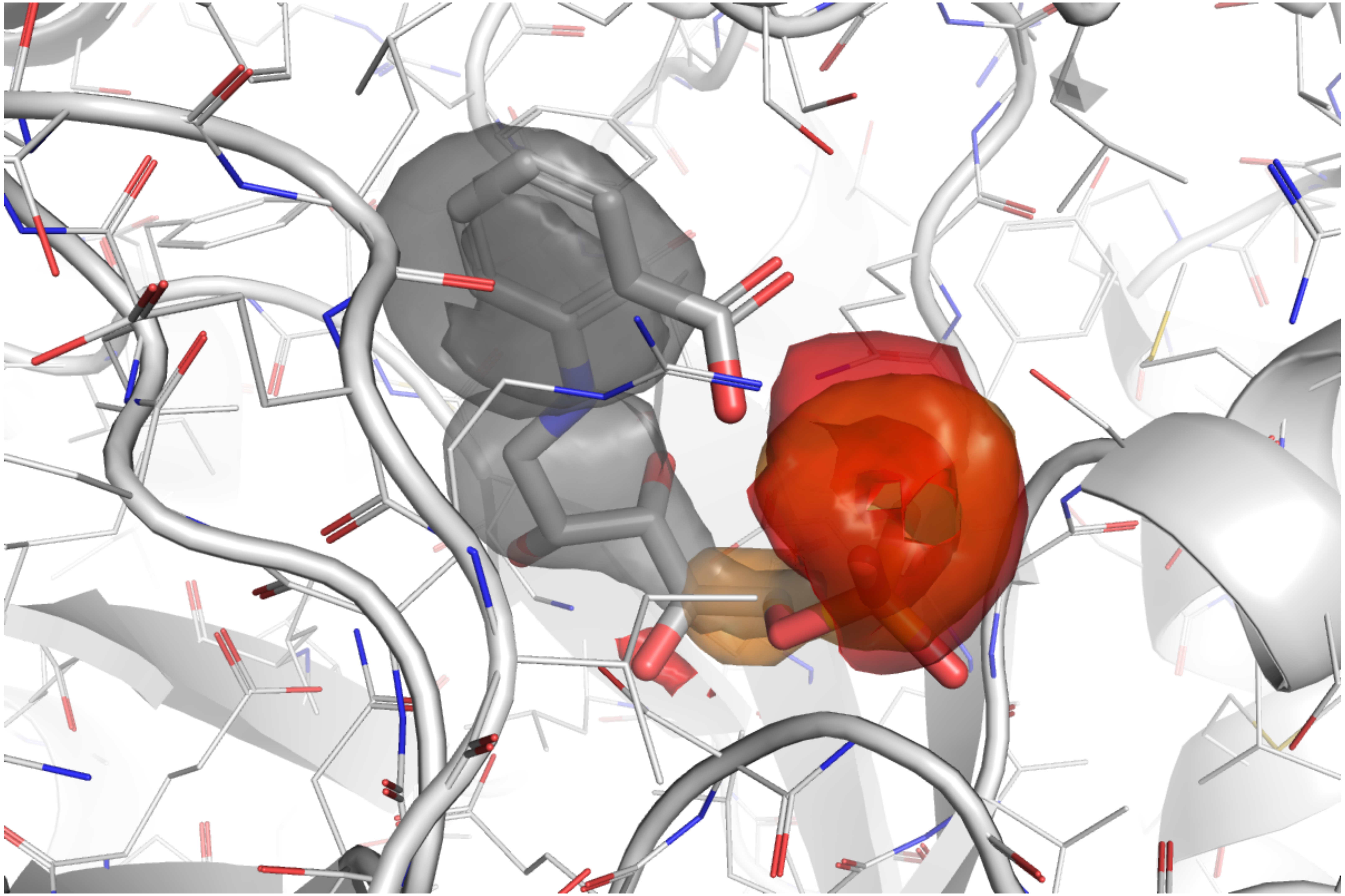
4PYX

Autoencoding Examples



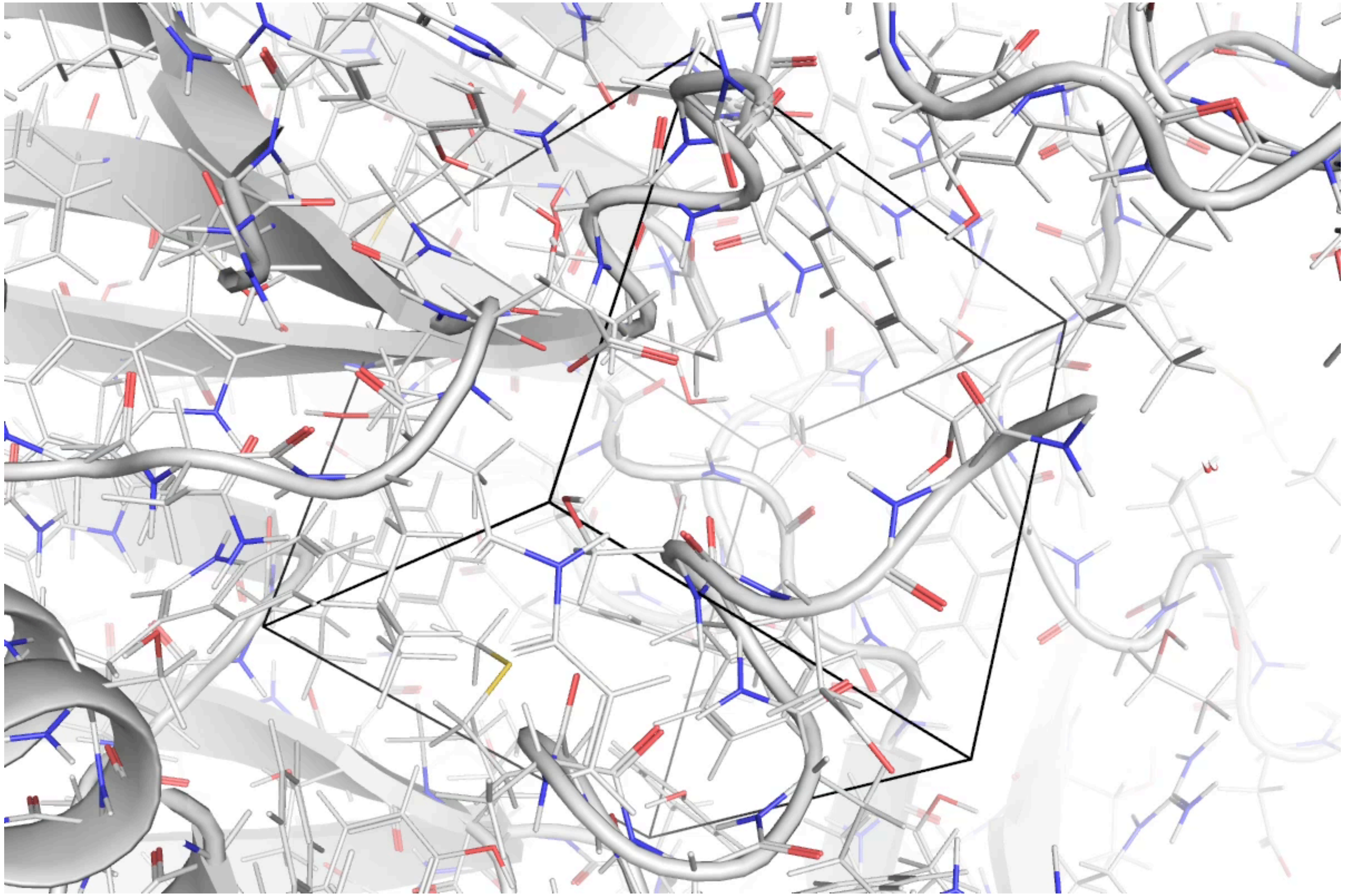
1LBF

Autoencoding Examples

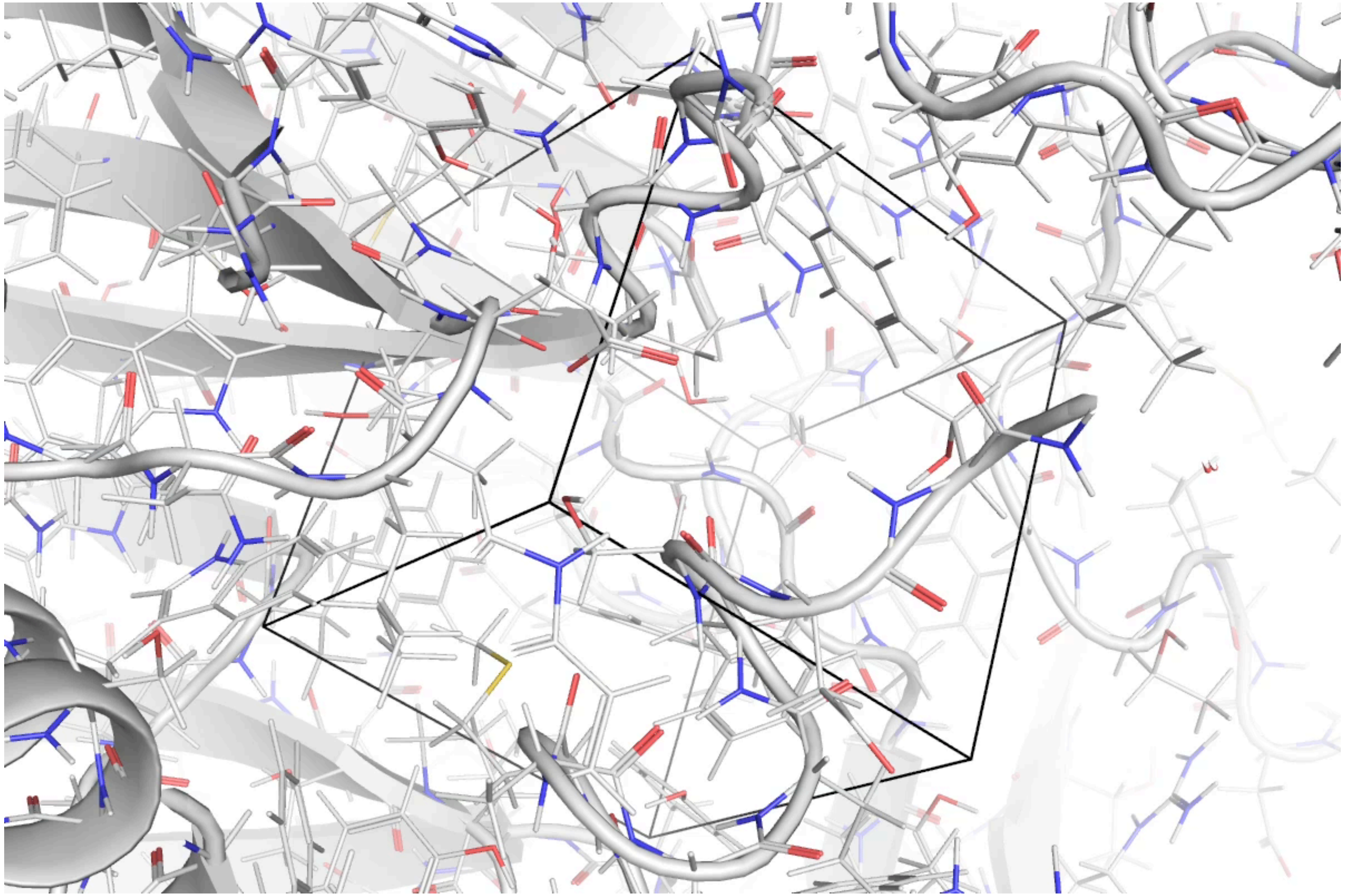


1LBF

Conditioning on the Receptor

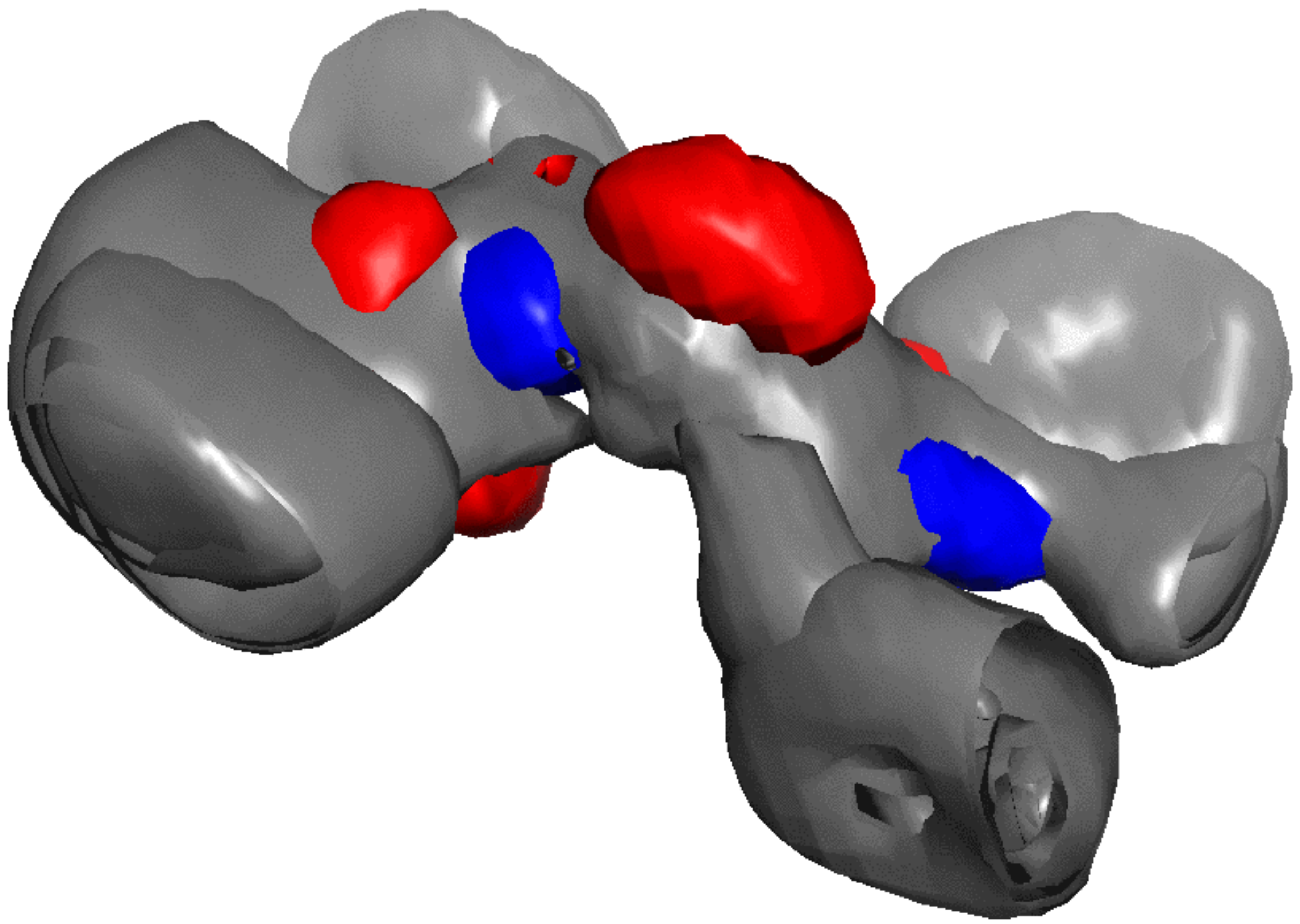


Conditioning on the Receptor



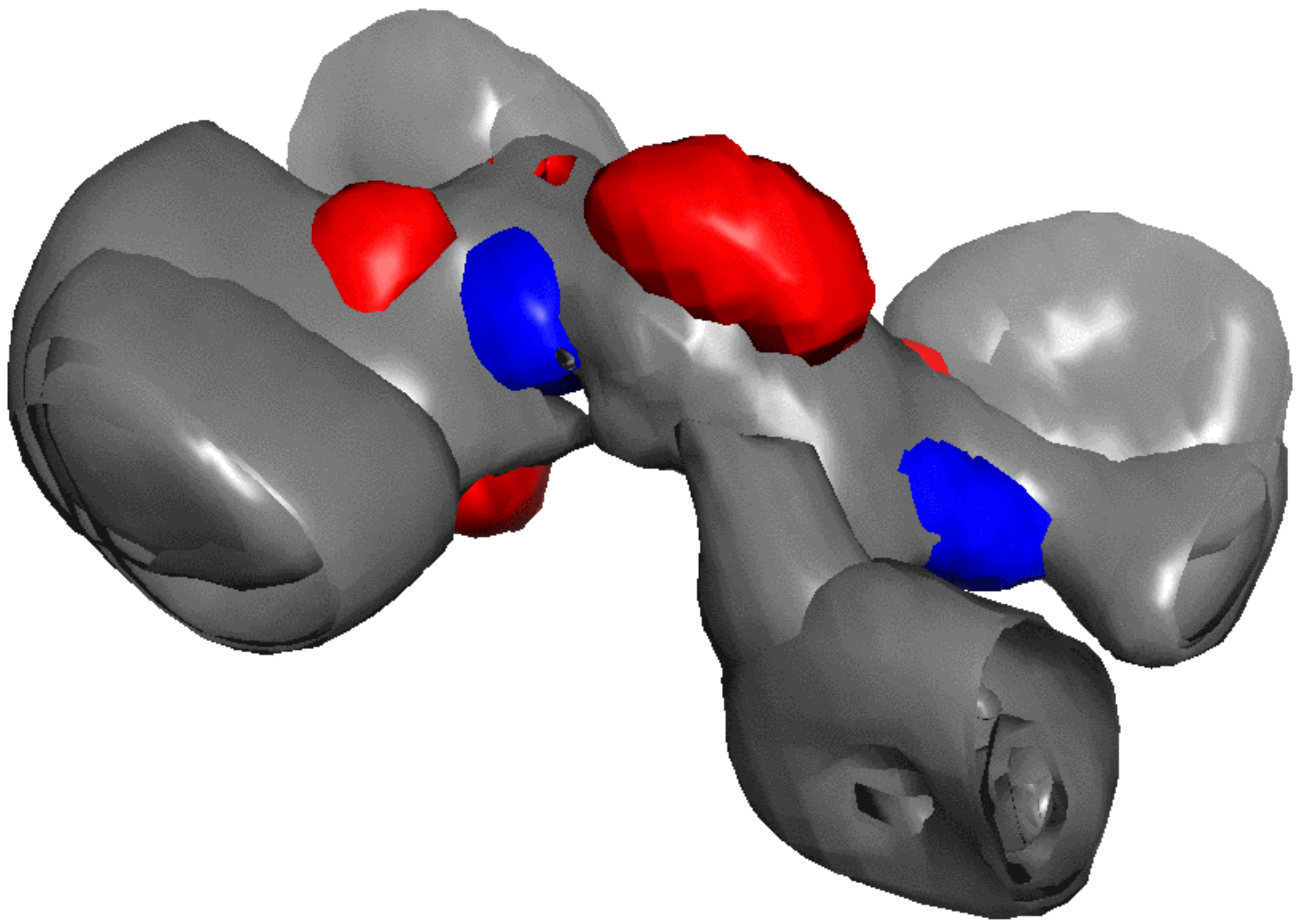
Atom Fitting

$$a^* = \underset{a}{\operatorname{argmin}} \|d - D(a)\|_2^2 + \lambda E(a)$$

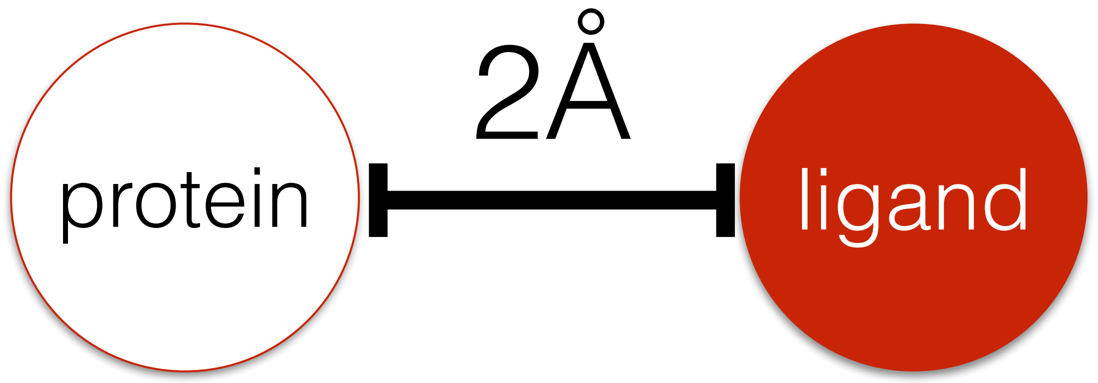


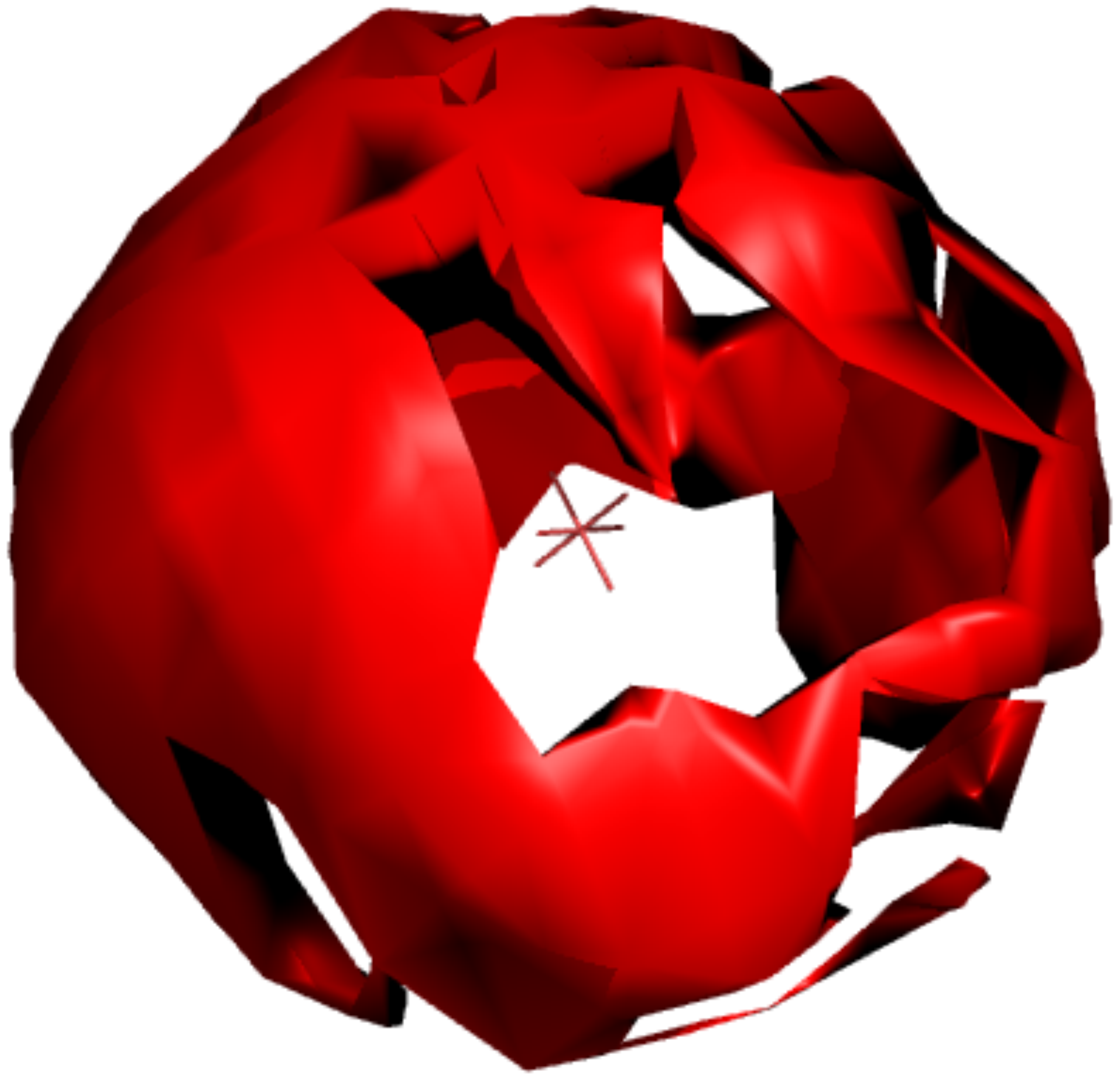
Atom Fitting

$$a^* = \underset{a}{\operatorname{argmin}} \|d - D(a)\|_2^2 + \lambda E(a)$$

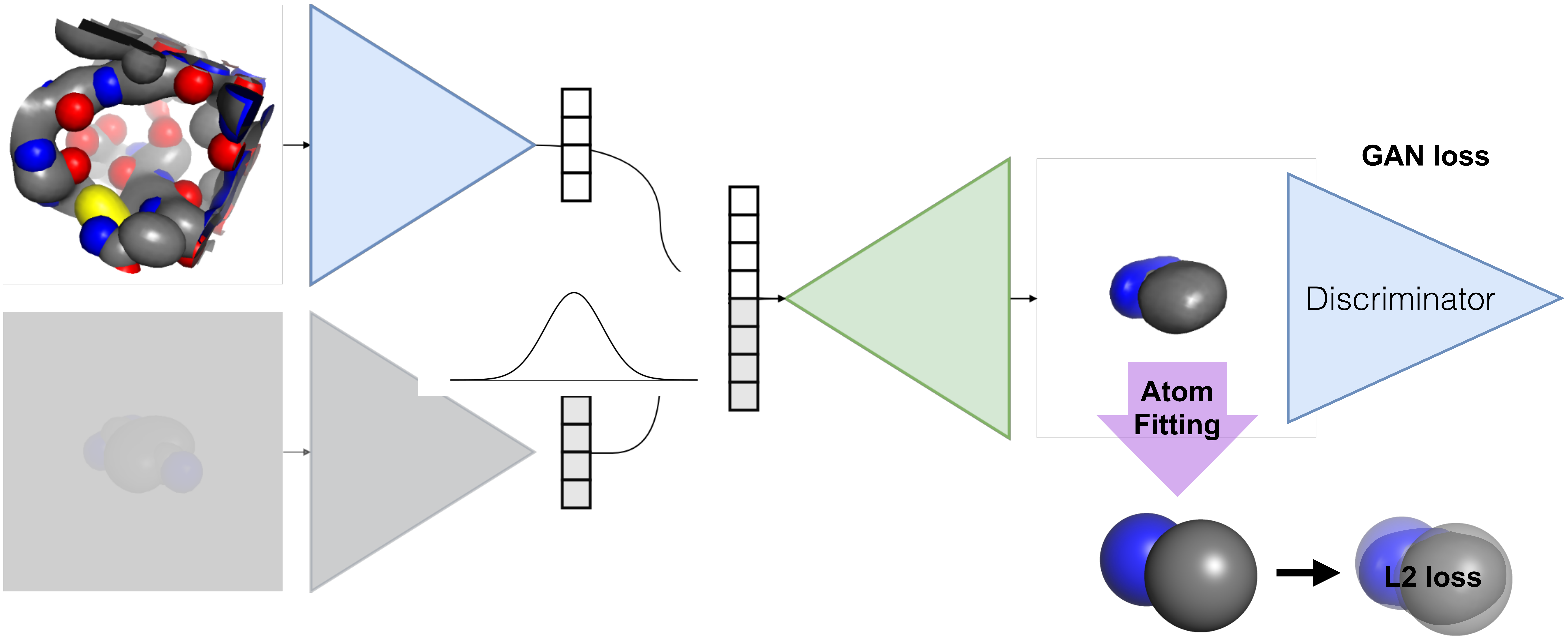


Two Atom Toy System

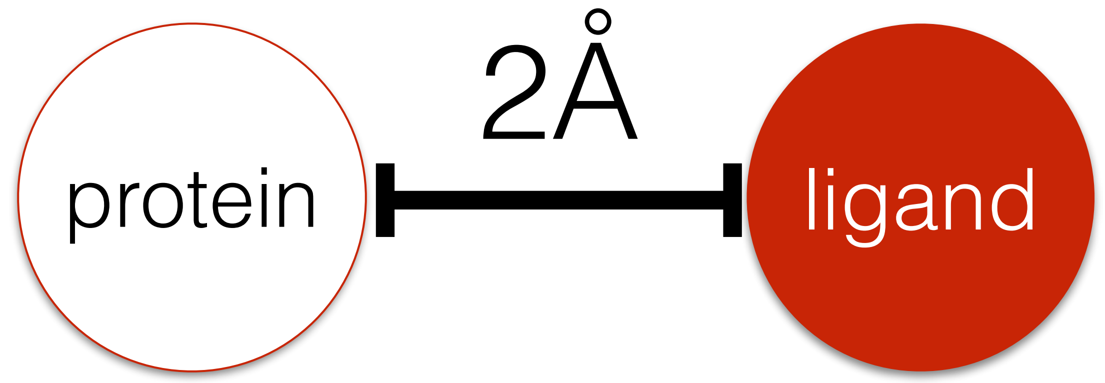




Atom Fitting Loss

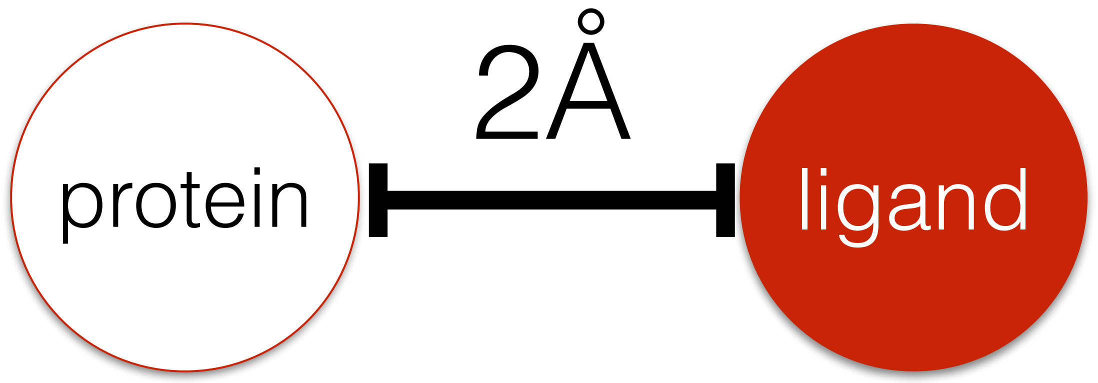


Atom Fitting Loss



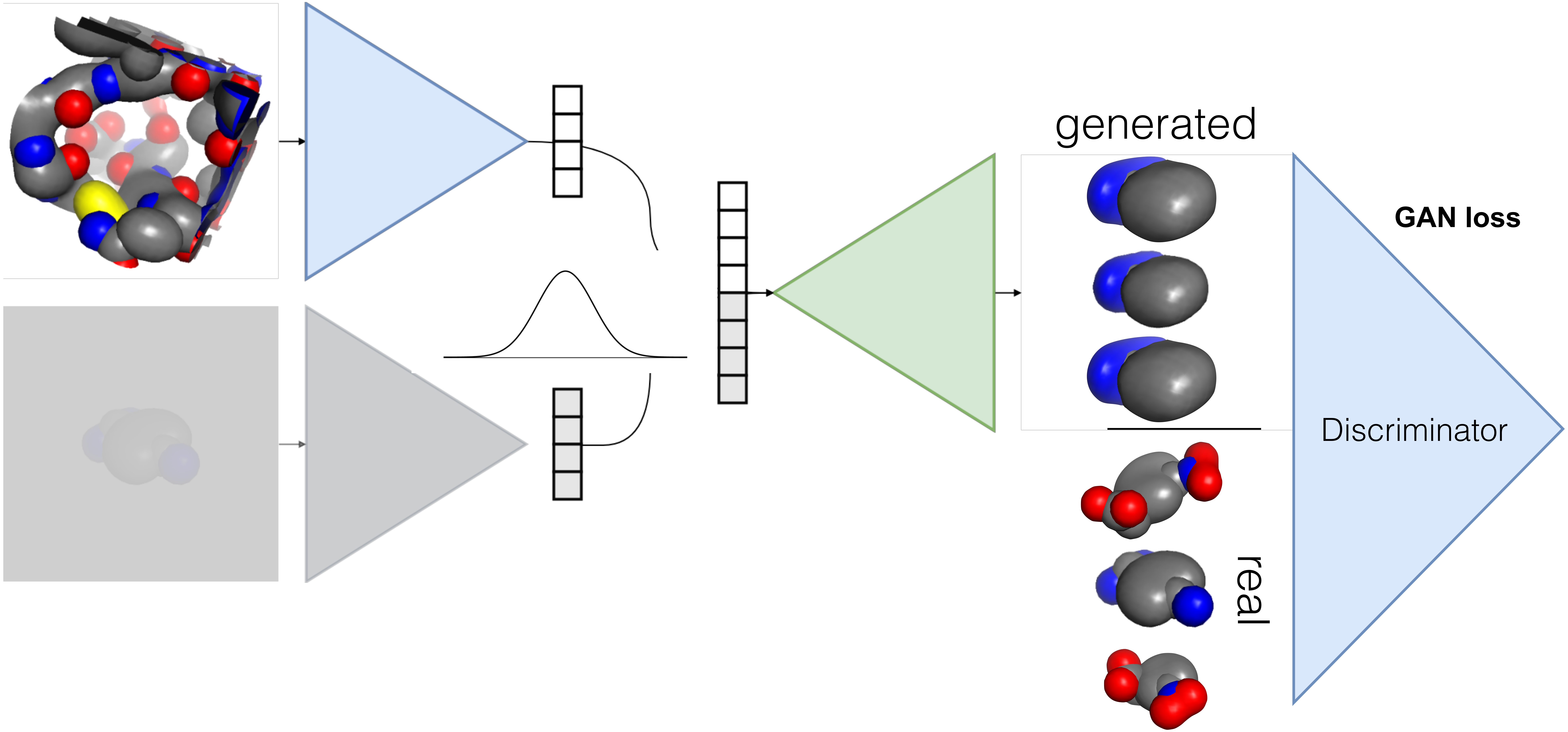
Two atom toy system

Atom Fitting Loss

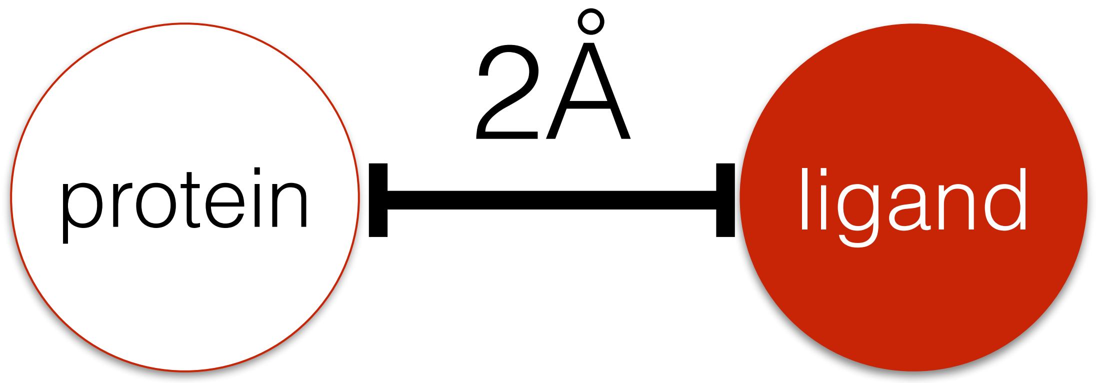


Two atom toy system

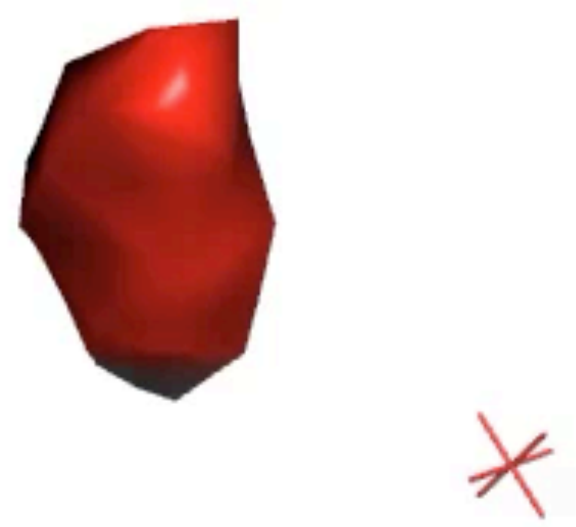
Batch Discrimination



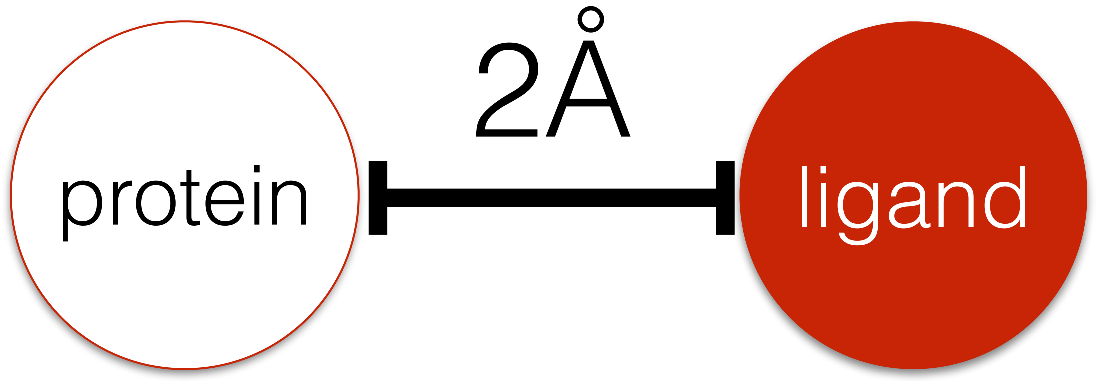
Atom Fitting + Batch Discrimination



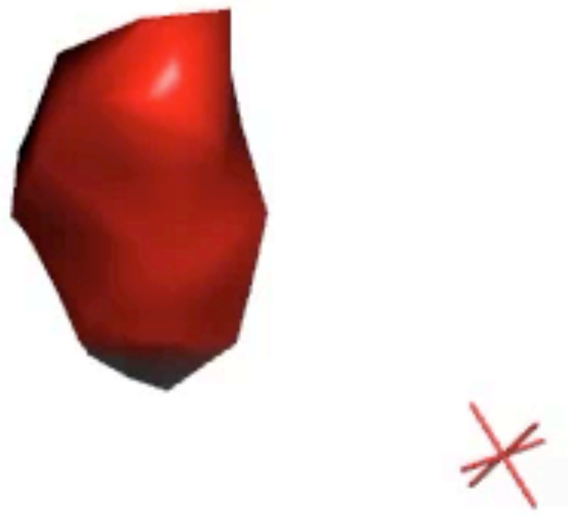
Two atom toy system



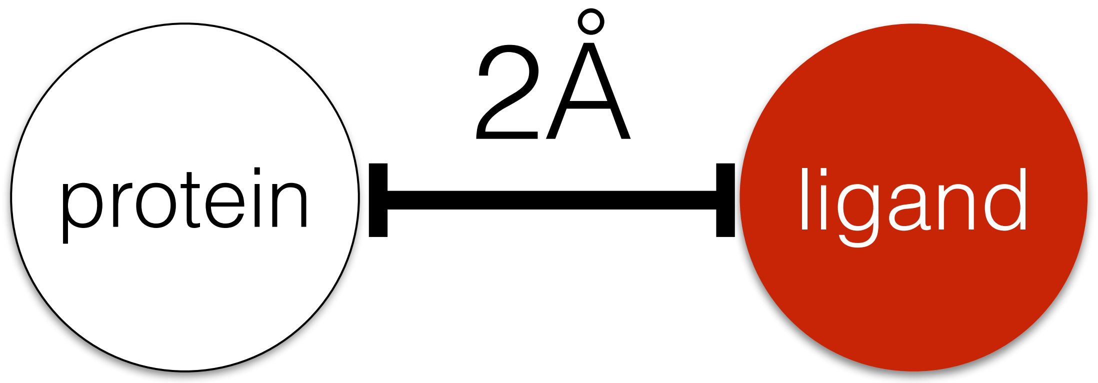
Atom Fitting + Batch Discrimination



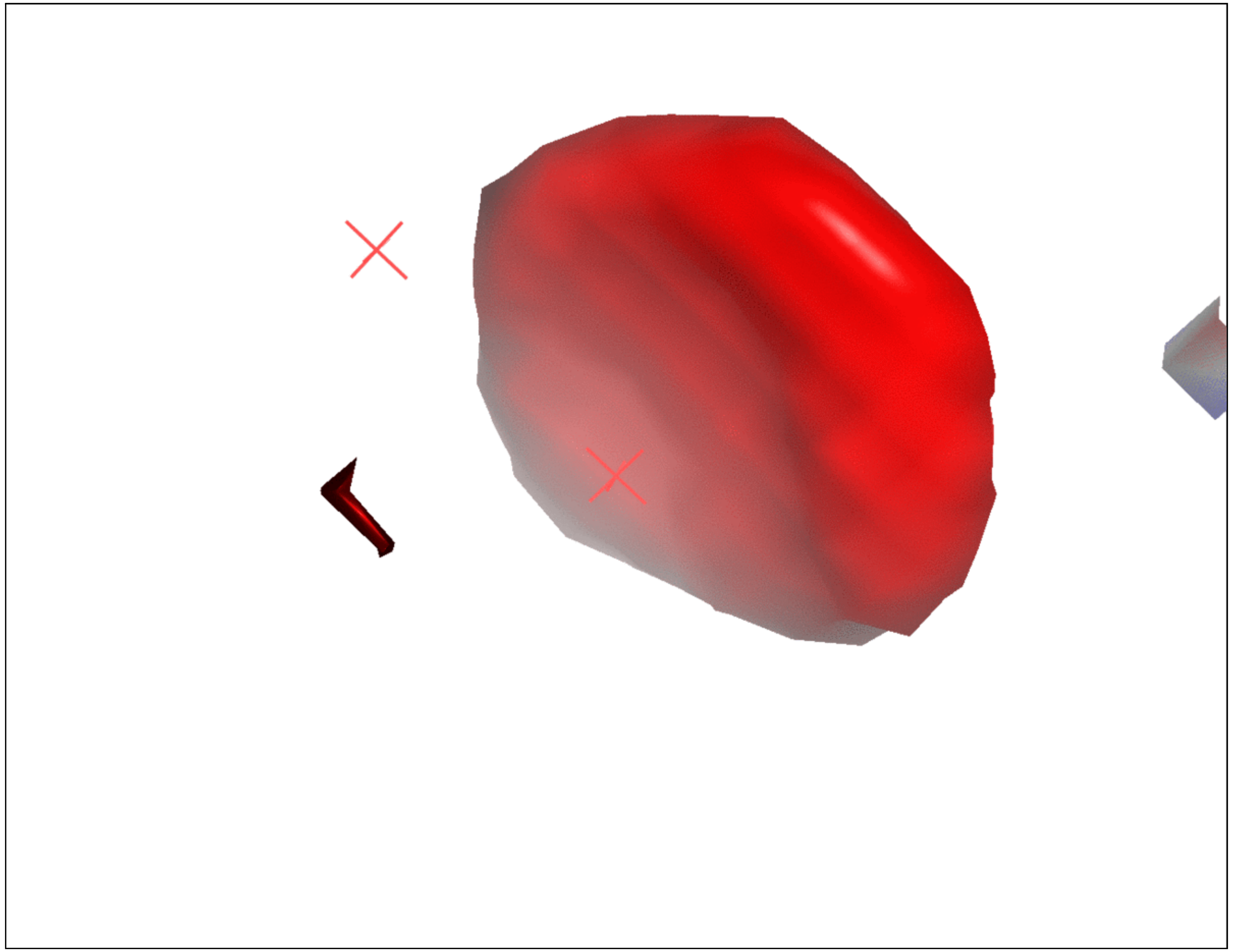
Two atom toy system



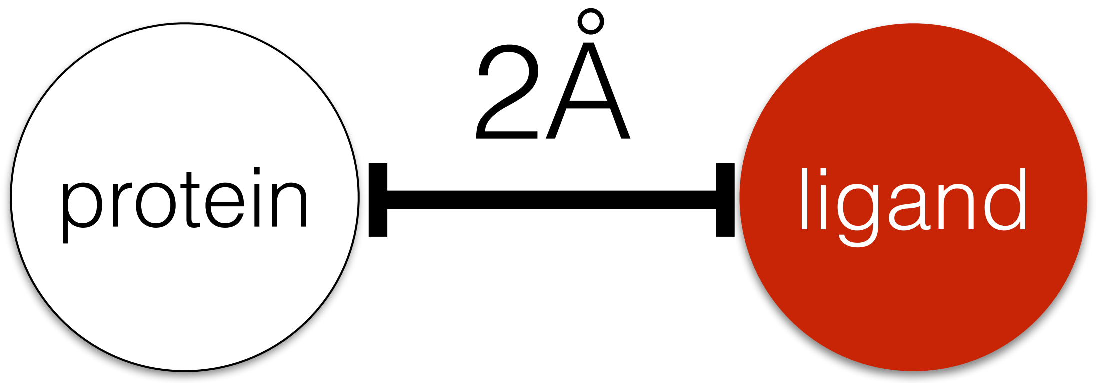
Iterpolating



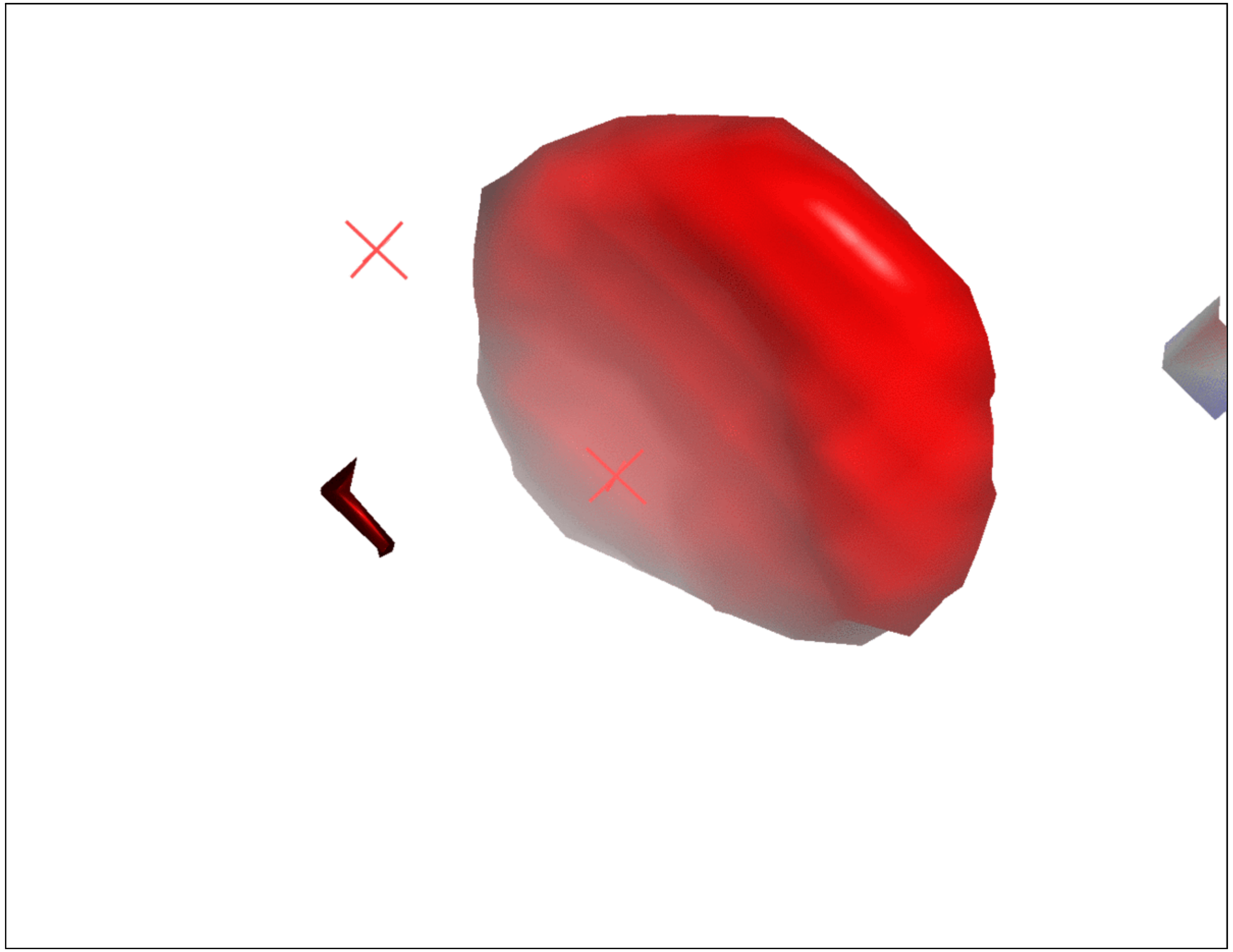
Two atom toy system



Iterpolating



Two atom toy system

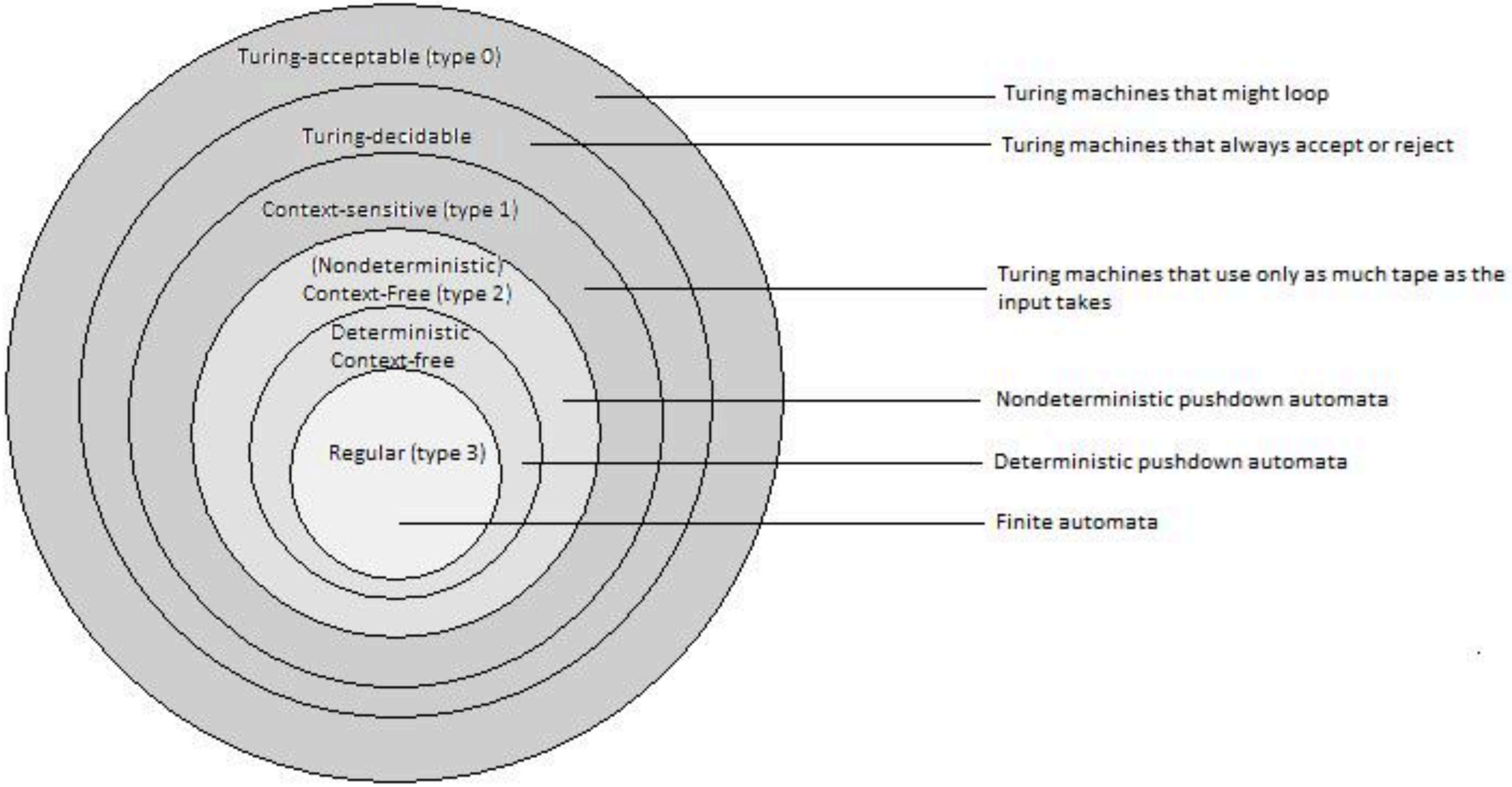


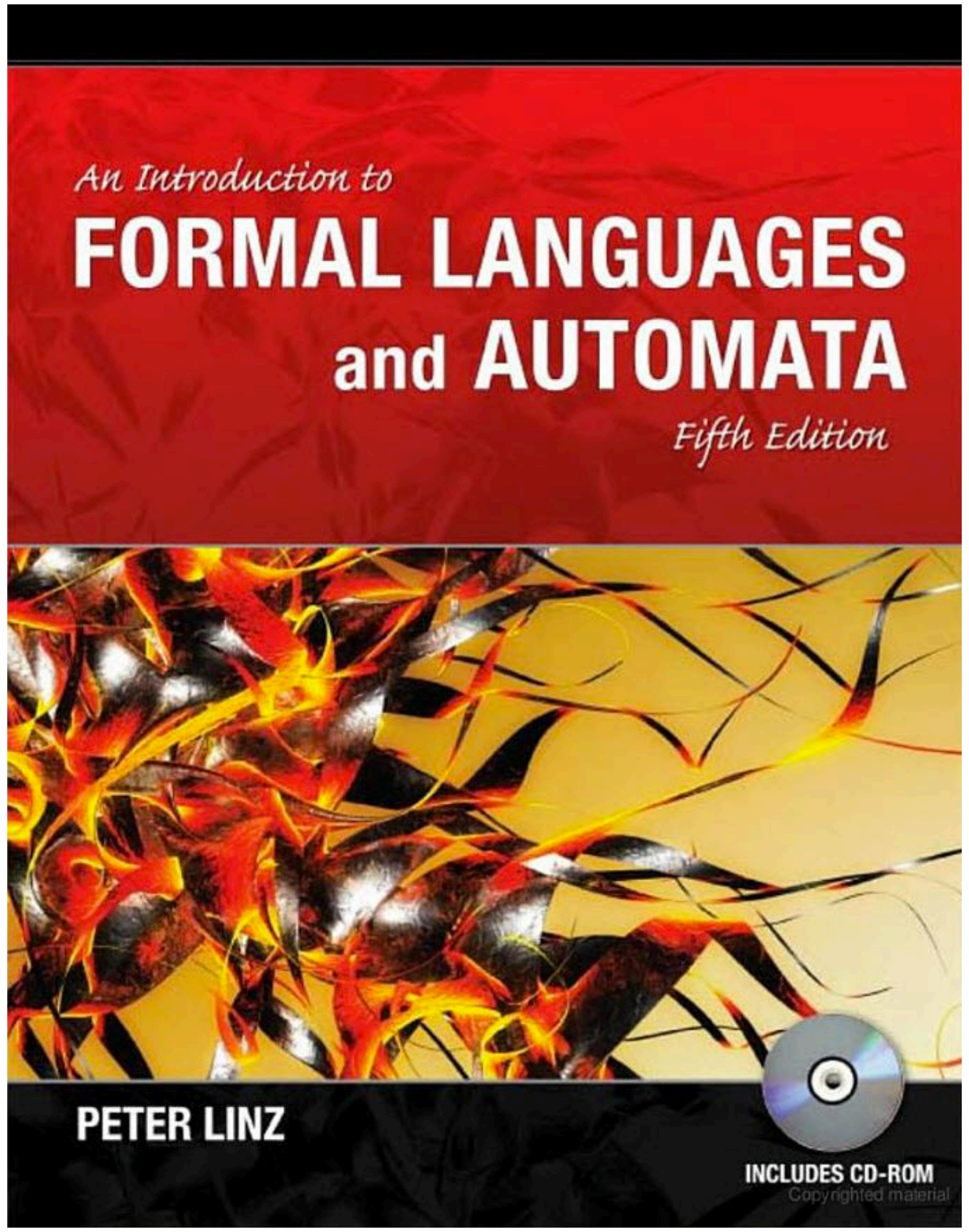

And Now For Something
Completely Different

LALRNN

Removing the third dimension

Chomsky Hierarchy





http://www.cs.appstate.edu/~dap/classes/2490/chapter11print.html

Grammars

Balanced Parentheses

$S \to \varepsilon$
$S \to (S)$
$S \to SS$

```
()
((()))()(())
()()()()()
```

Palindromes

$S \to \varepsilon$
$S \to aSa$
$S \to bSb$

```
aa
babbab
abbaabba
```

Arithmetic

```
E ::= id
    | num
    | E + E
    | E * E
    | ( E )
```

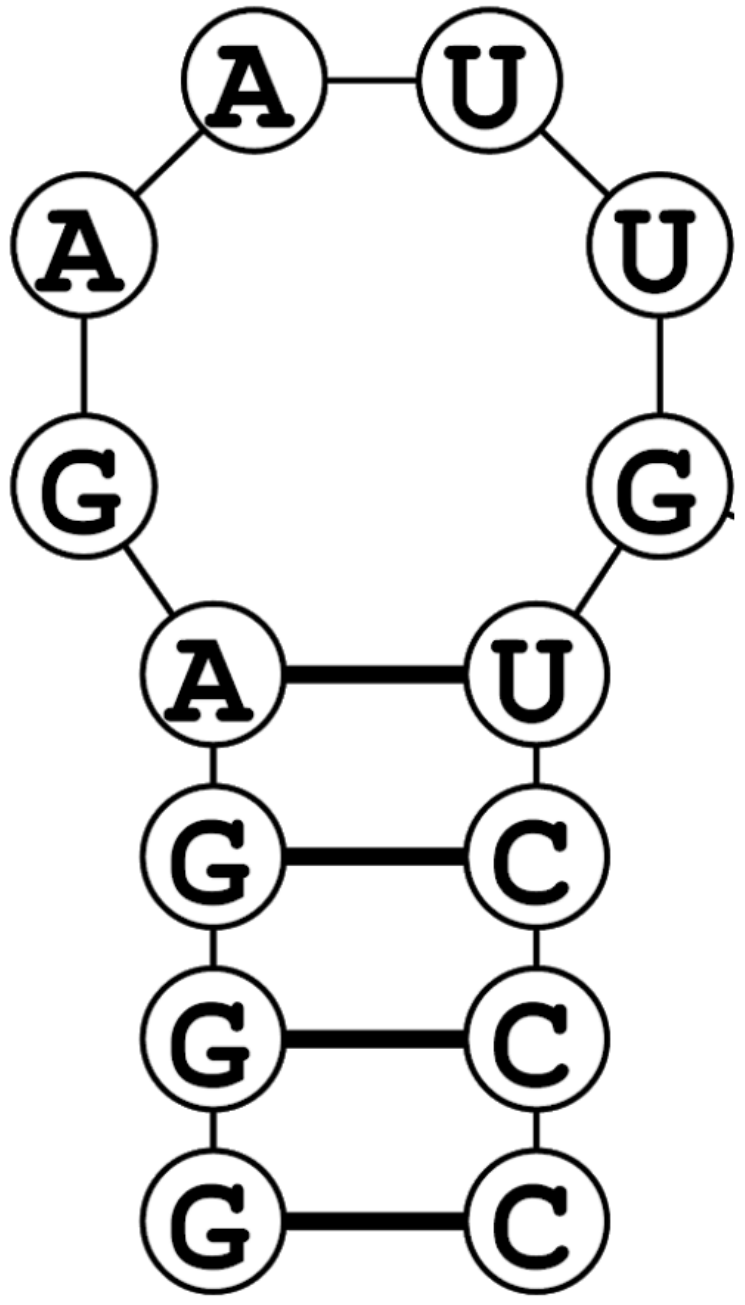
```
3 + 4 * 5
(3 + 4) * 5
```

Grammars

Balanced Parentheses

$S \rightarrow \varepsilon$
$S \rightarrow (S)$
$S \rightarrow SS$

( )
((( ))) ( ) (( ))
( ) ( ) ( ) ( ) ( )



GGGAGAAUUGUCCC
((((......))))

Palindromes

$S \rightarrow \varepsilon$
$S \rightarrow aSa$
$S \rightarrow bSb$

aa
babbab
abbaabba

Arithmetic

```
E ::= id
    | num
    | E + E
    | E * E
    | ( E )
```

3 + 4 * 5
(3 + 4) * 5

Bottom Up Parsing (LALR)

A PDA can be implemented with a *parse table*

	action			goto	
state	*ident*	+	$	E	T
0	**s3**			**g1**	**g2**
1			**a**		
2		**s4**	**r2**		
3		**r3**	**r3**		
4	**s3**			**g5**	**g2**
5			**r1**		

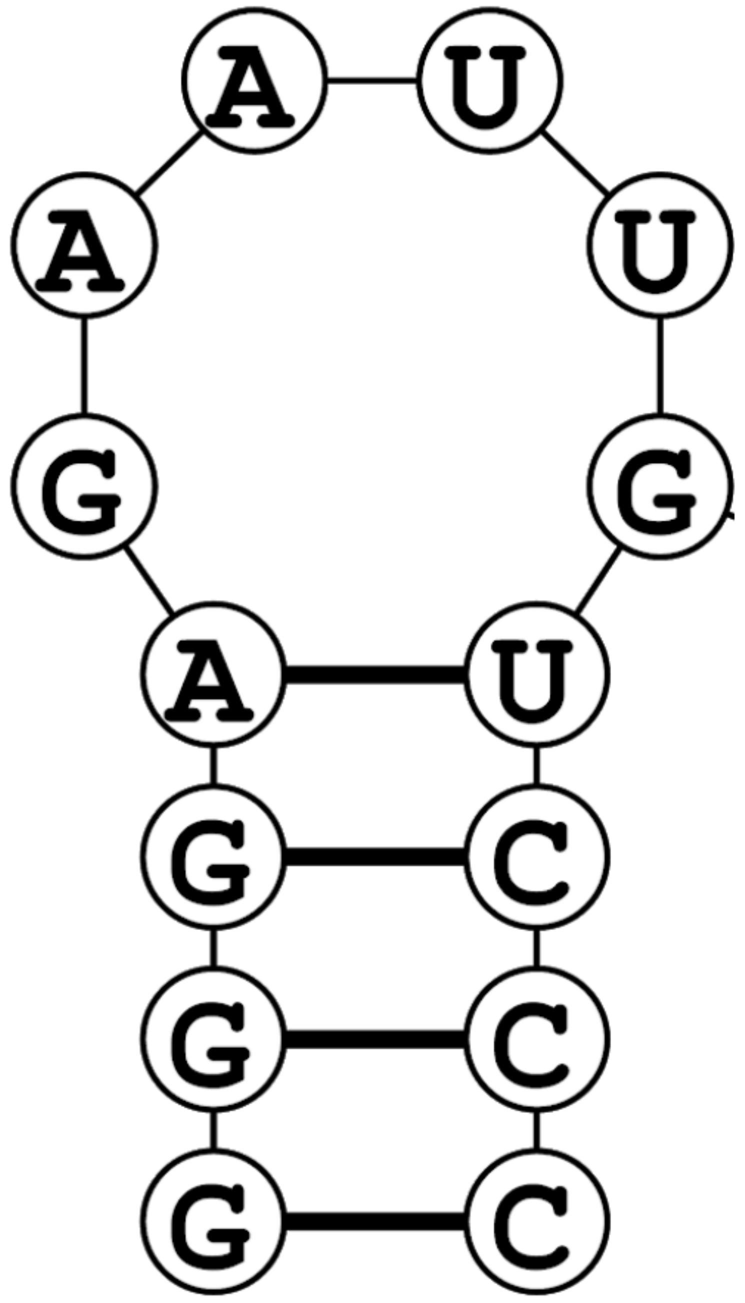
$S \rightarrow E\$$
$E \rightarrow T + E$
$E \rightarrow T$
$T \rightarrow$ *identifier*

states != rules

x + y$

```
while(true)
  s = state on top of stack
  a = current input token
  if(action[s][a] == sN)             shift
    push N
    a = next input token
  else if(action[s][a] == rR)        reduce
    remove rhs of rule R from stack
    X = lhs of rule R
    N = state on top of stack
    push goto[N][X]
  else if(action[s][a] == a)         accept :-)
    return success
  else                        error
    return failure
```

RNA Secondary Structure



GGGAGAAUUGUCCC

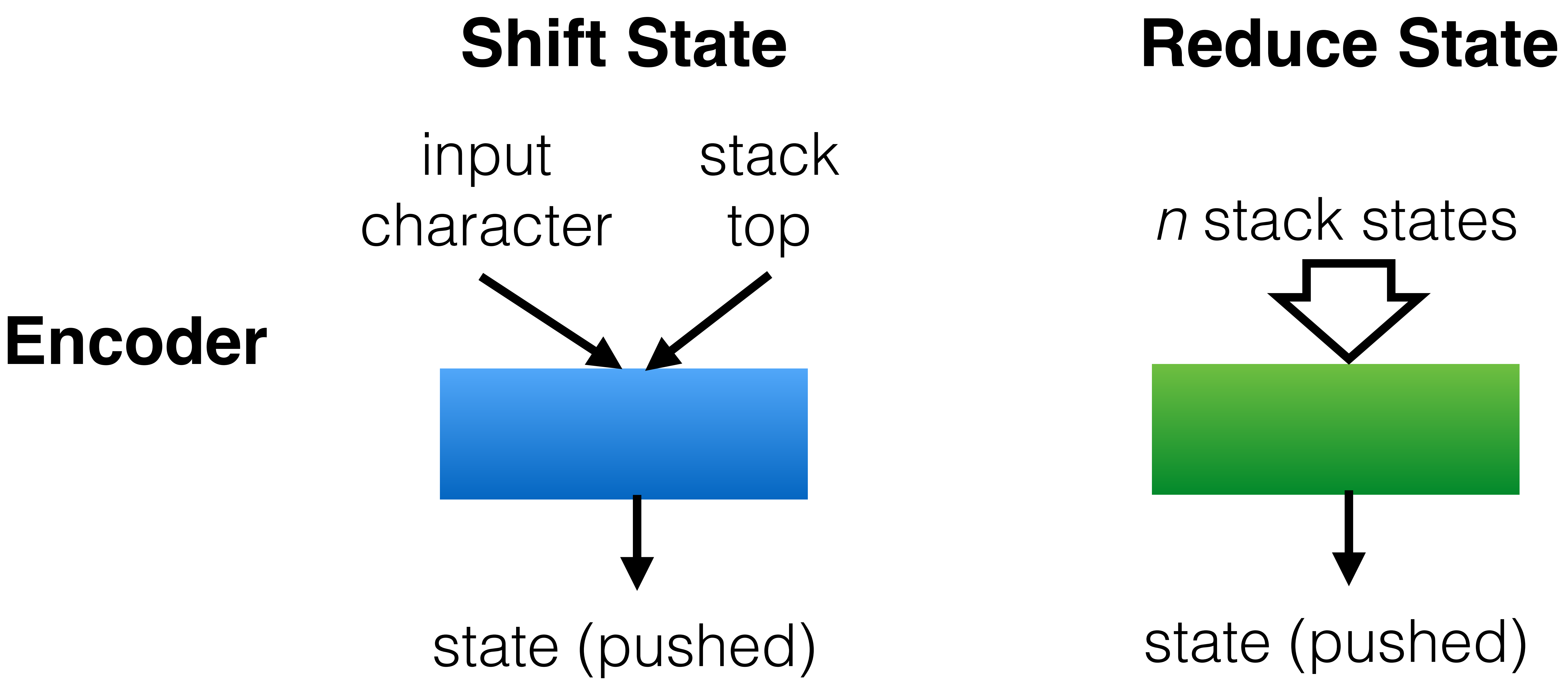
((((......))))

$$S \rightarrow .$$
$$S \rightarrow ()$$
$$S \rightarrow (S)$$
$$S \rightarrow S(S)$$
$$S \rightarrow S.$$
$$S \rightarrow S()$$

state	.	(	)	S
0	s6	s1		g5
1	s6	s1	s7	g4
2	s6	s1	s11	g3
3	s10	s2	s9	
4	s10	s2	s8	
5	s10	s2		
6	Reduce S → .			
7	Reduce S → ()			
8	Reduce S → (S)			
9	Reduce S → S(S)			
10	Reduce S → S.			
11	Reduce S → S()			
12	END			

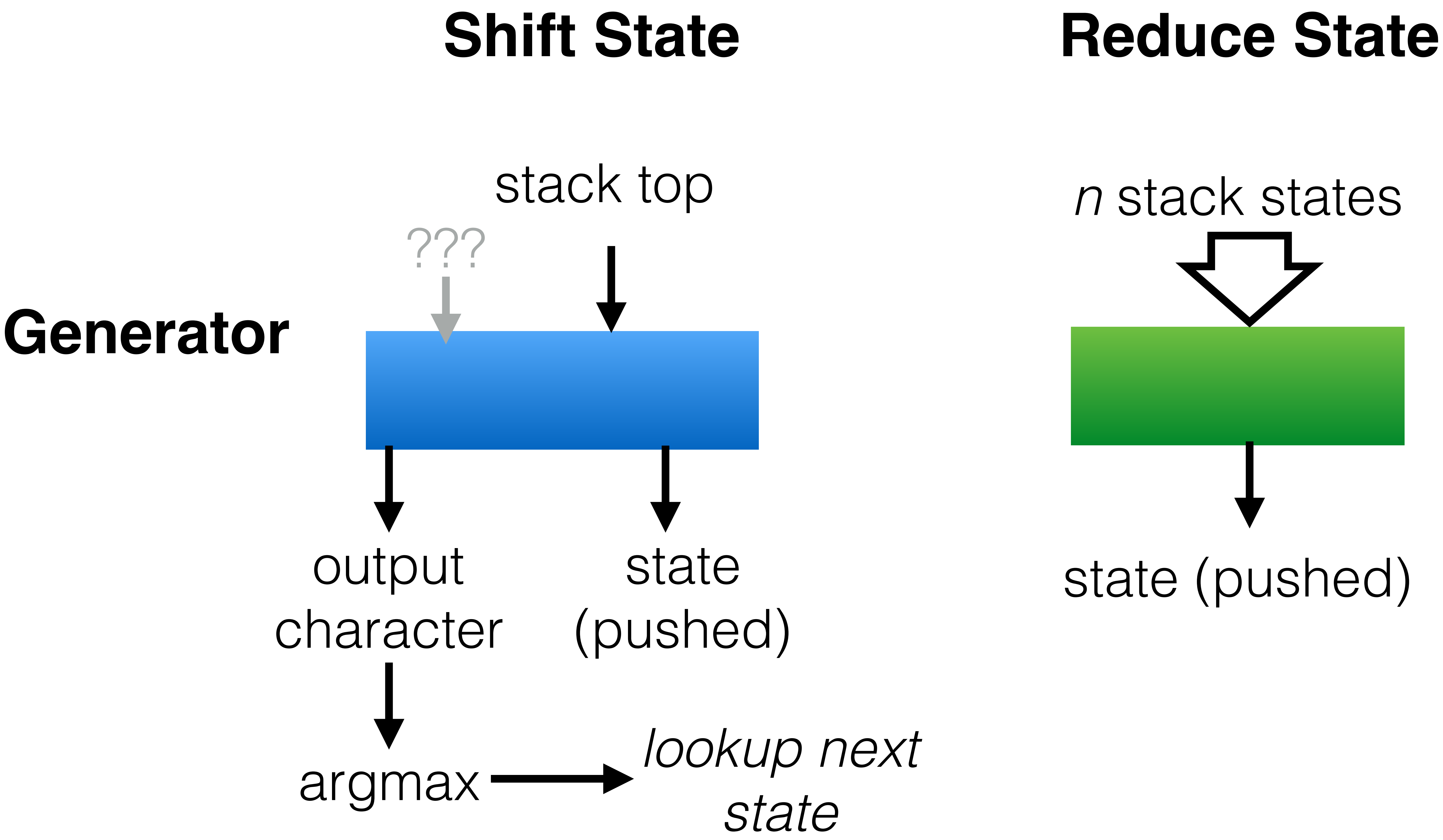
The NN Part

Implement every state as its own neural network that calculates a function of the input in the context of the parse (encoder) or outputs a syntactically correct string according to the rules of the grammar (generator)

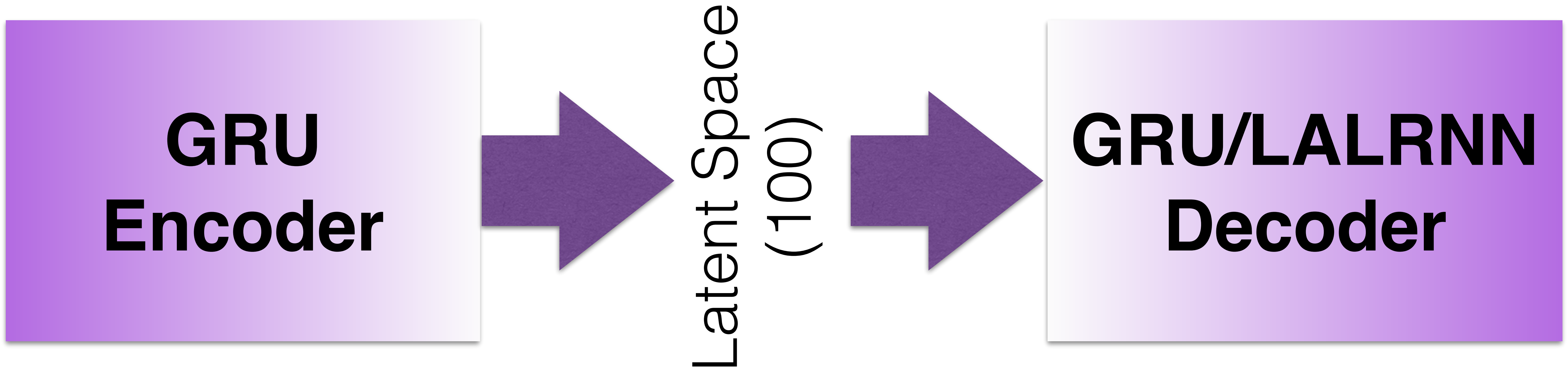


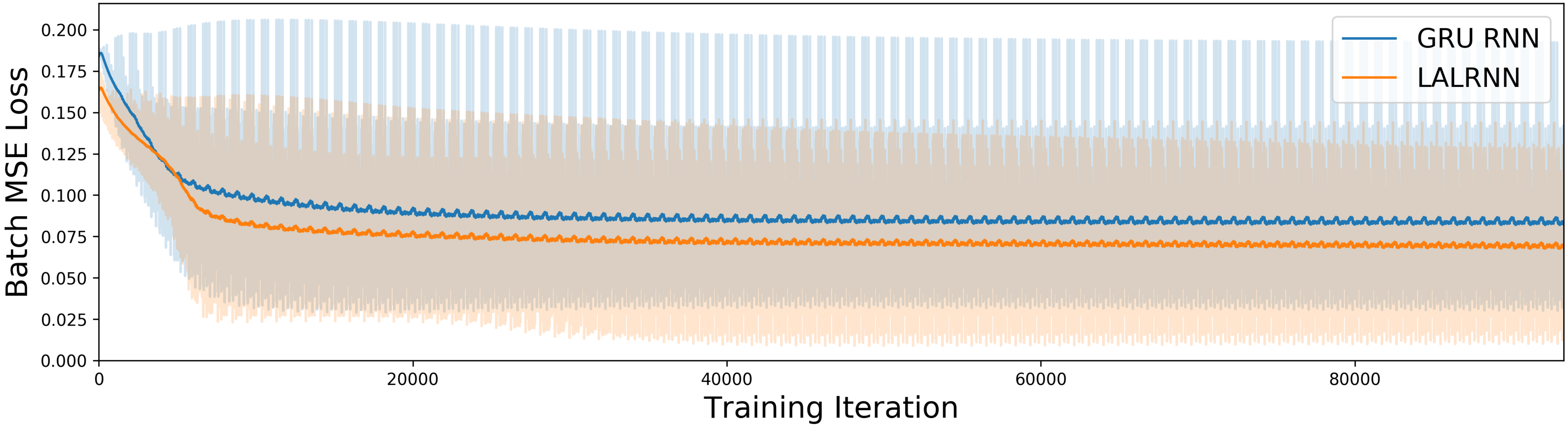
The NN Part

Implement every state as its own neural network that calculates a function of the input in the context of the parse (encoder) or outputs a syntactically correct string according to the rules of the grammar (generator)



LALRNN vs GRU





Thoughts

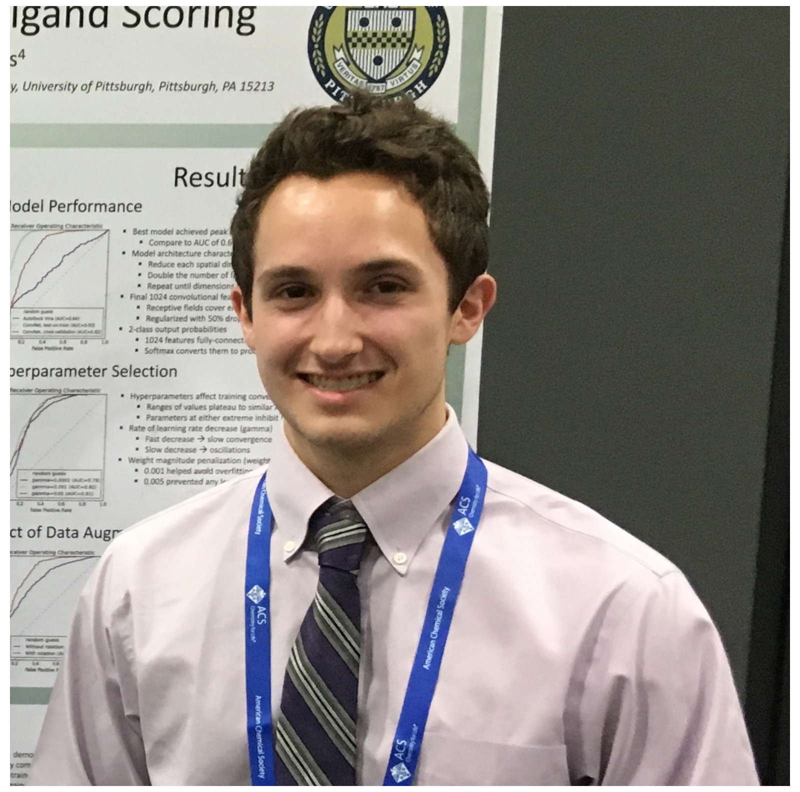
Deep Learning: not just for supervised learning

- can *generate* examples from data distribution

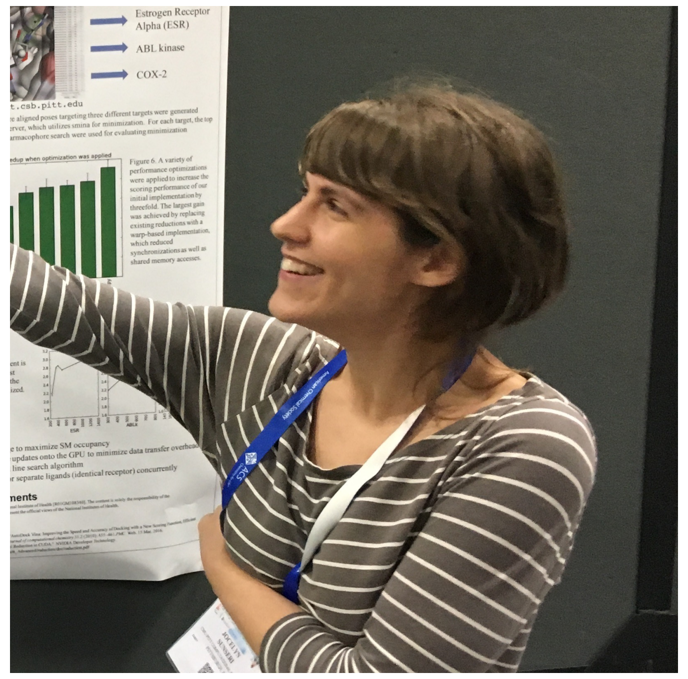
Teach the network through data **and** instruction

- how can we productively impose physics in the network?

Acknowledgements



Matt Ragoza



Jocelyn Sunseri

Paul Francoeur

Department of Computational and Systems Biology

AI GRANT

National Institute of General Medical Sciences R01GM108340

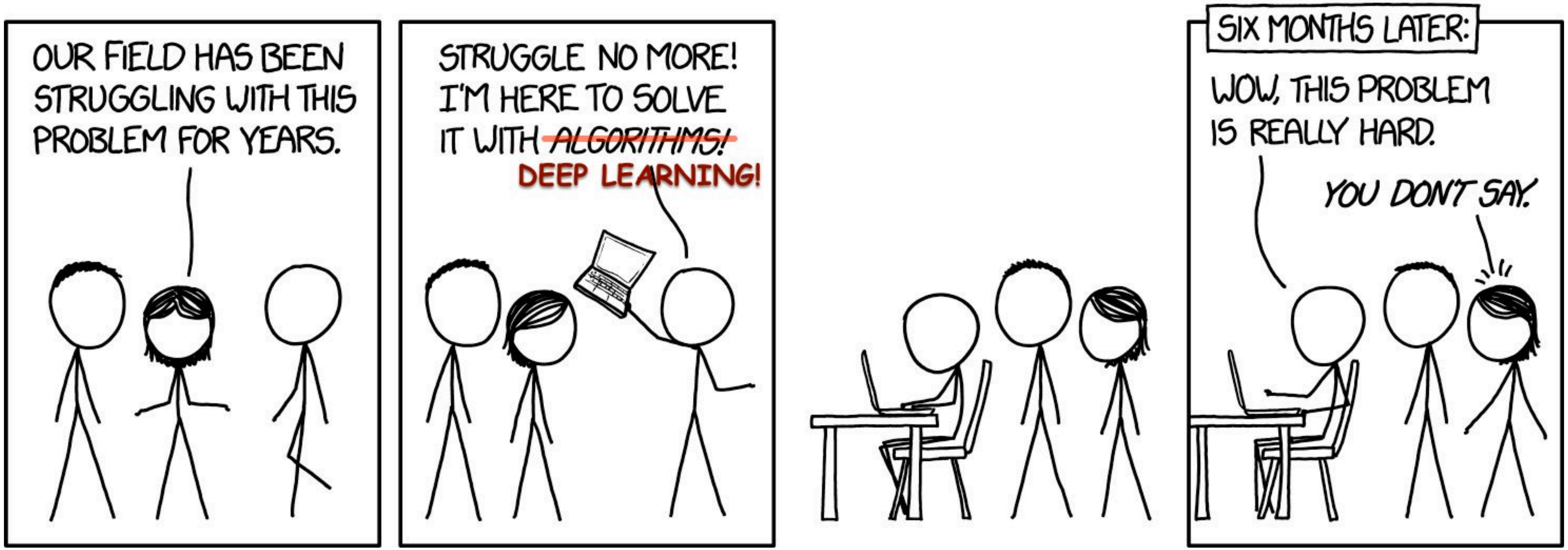
NVIDIA

Google Cloud

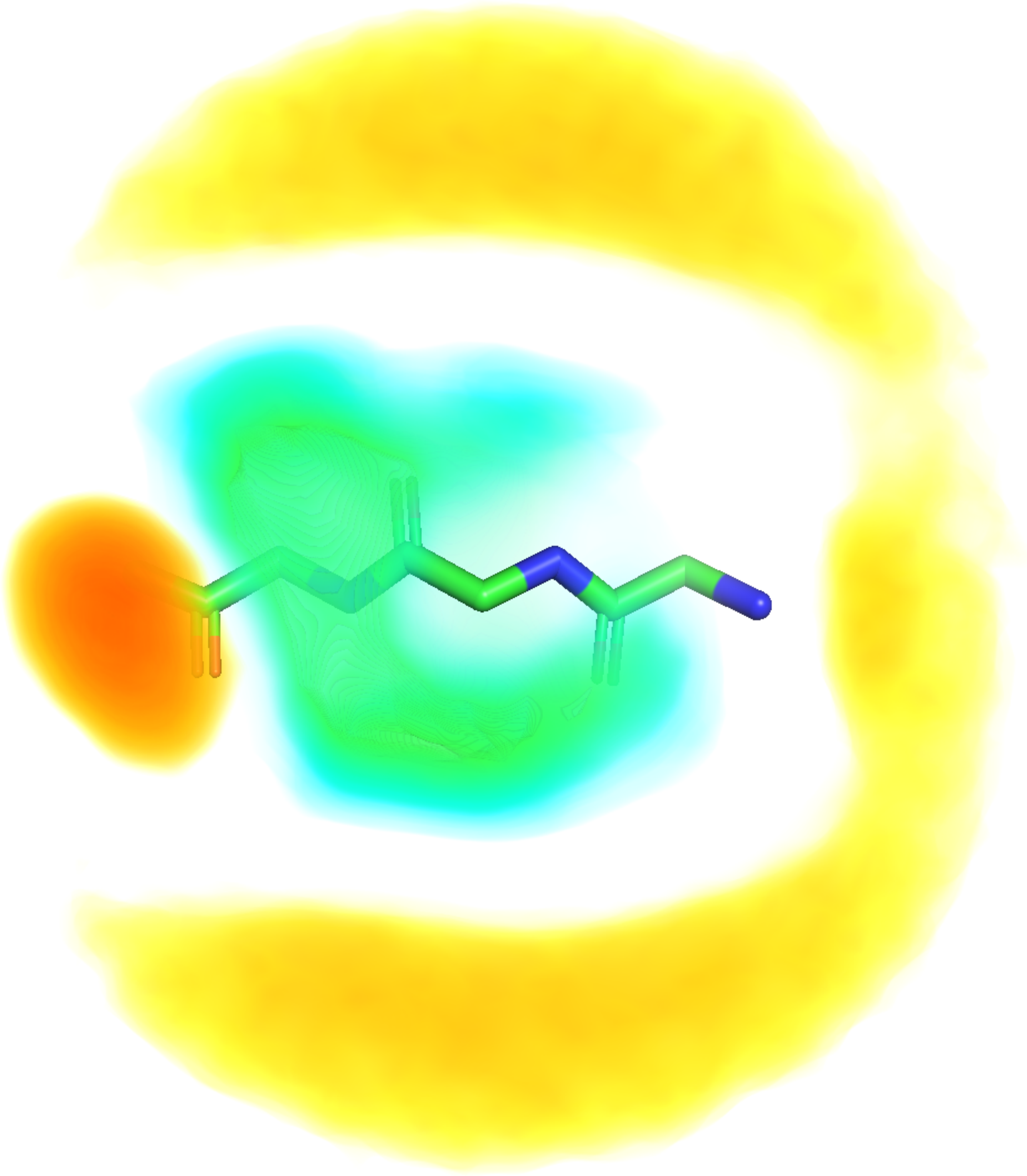
github.com/gnina

http://bits.csb.pitt.edu

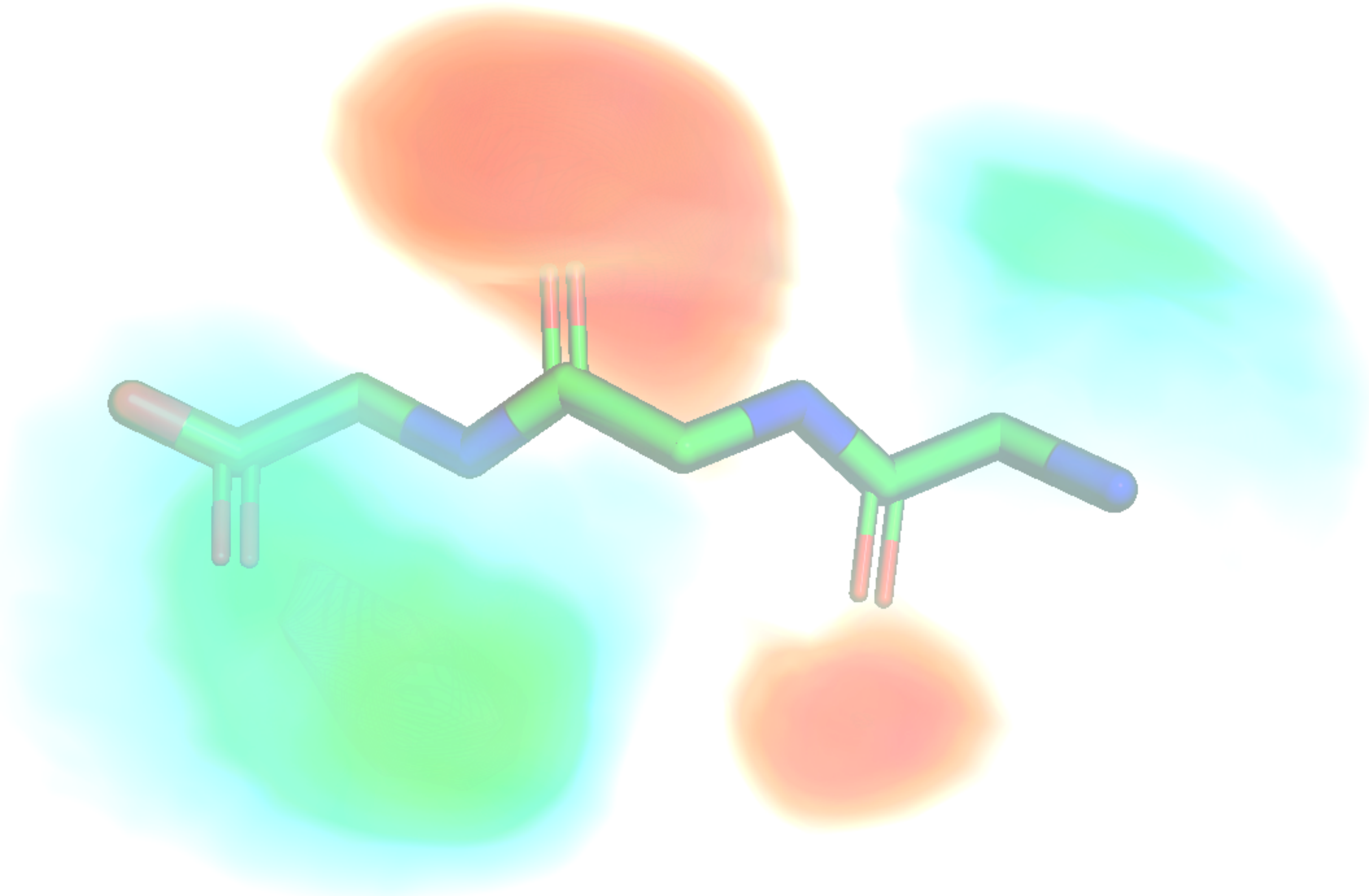
@david_koes



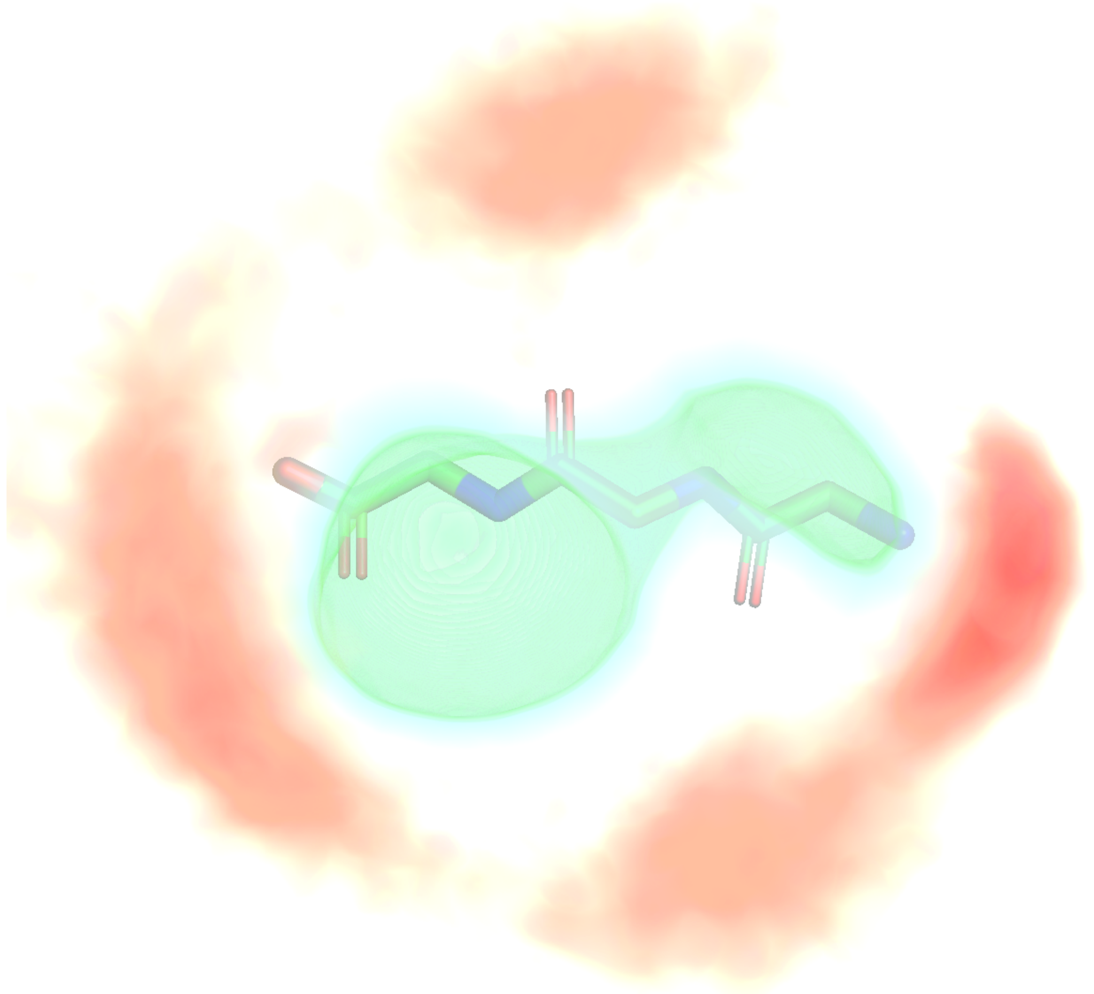
But what is it learning?



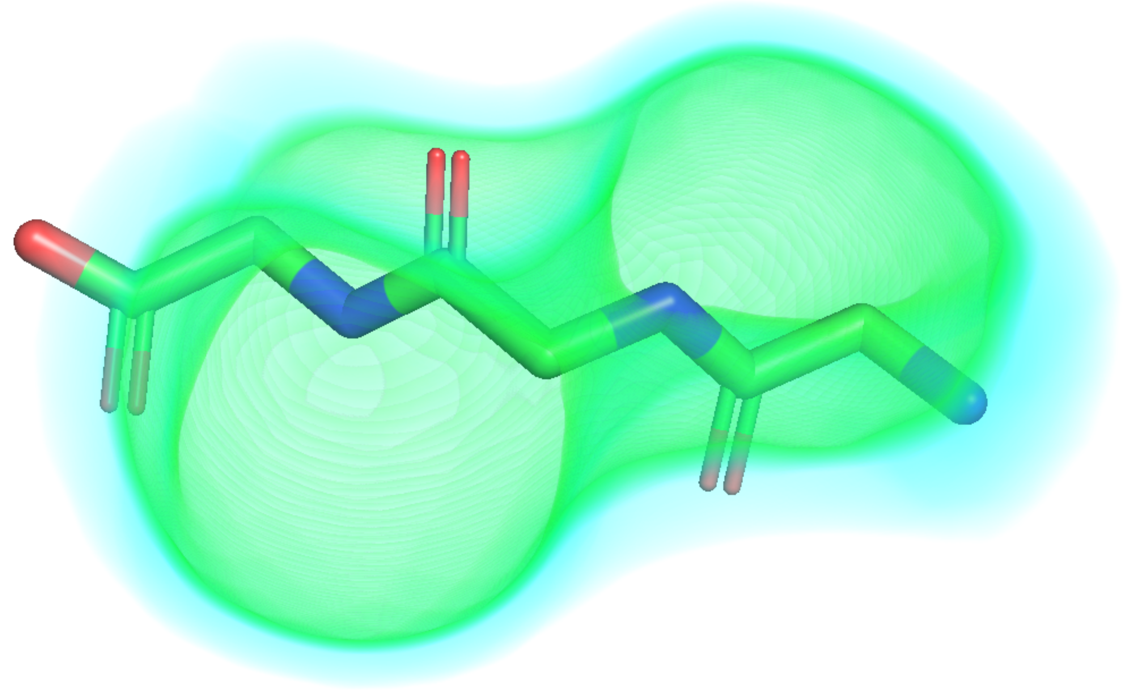
AliphaticCarbon



NitrogenDonor



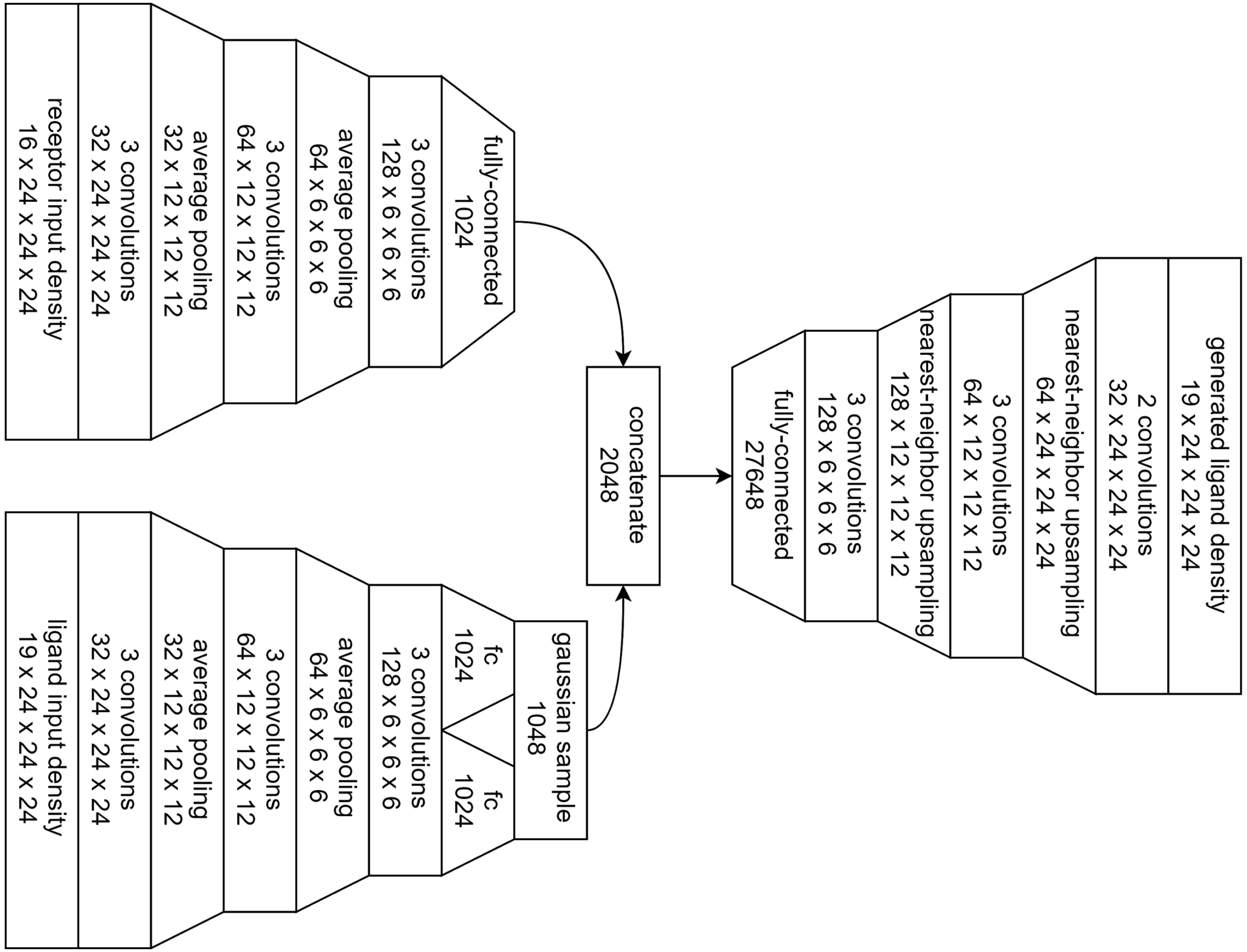
OxygenDonorAcceptor



OxygenAcceptor

Model Architecture

n_levels = 3
conv_per_level = 3
n_filters = 32
width_factor = 2
n_latent = 1024



Training Procedure

2016 PDBbind refined set
3765 crystal structures
Vina docking
RMSD < 2 Å from crystal pose
8648 poses (~2.3 per target)
random rotation & translation

Adam optimization
base_lr = 0.00001
momentum = 0.9
momentum2 = 0.999
max_iter = 100000
batch_size = 50