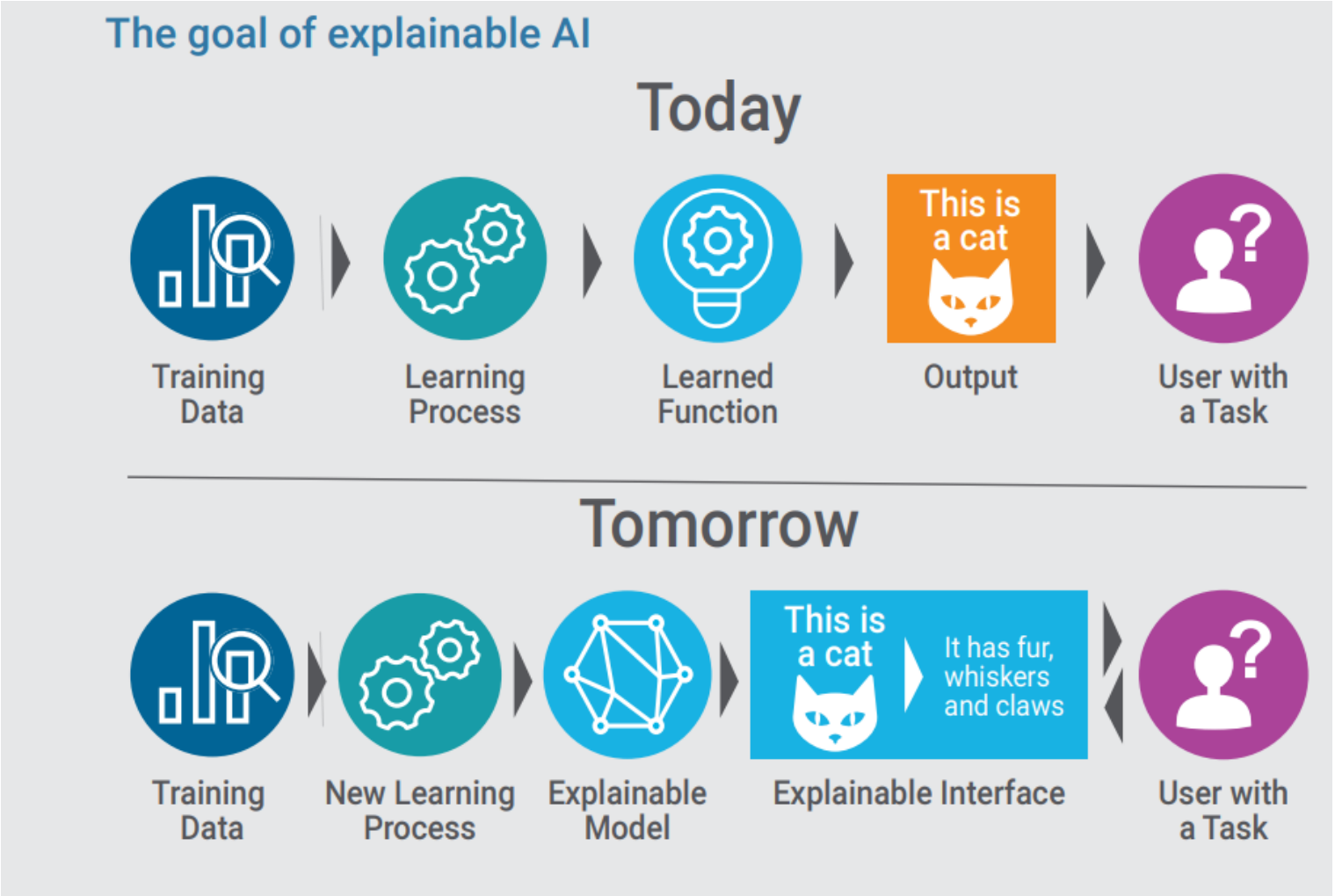
Explainable 3D-CNN for Protein-ligand Binding

Summer Intern Work in LLNL

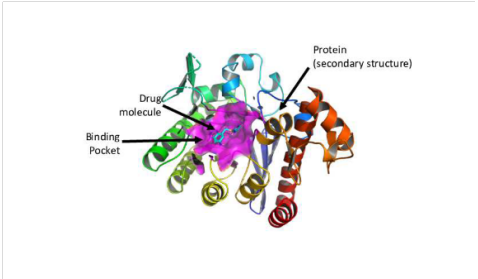
Zhi Zhang

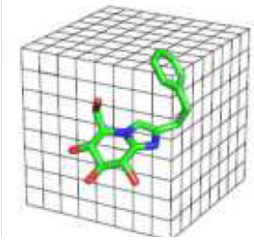
December 13, 2020

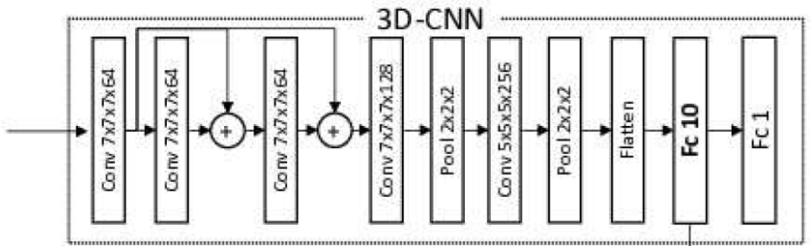
Background


The goal of explainable AI
Today
Training Data
Learning Process
Learned Function
This is a cat
Output
User with a Task
Tomorrow
Training Data
New Learning Process
Explainable Model
This is a cat
It has fur, whiskers and claws
Explainable Interface
User with a Task

3D-CNN Model





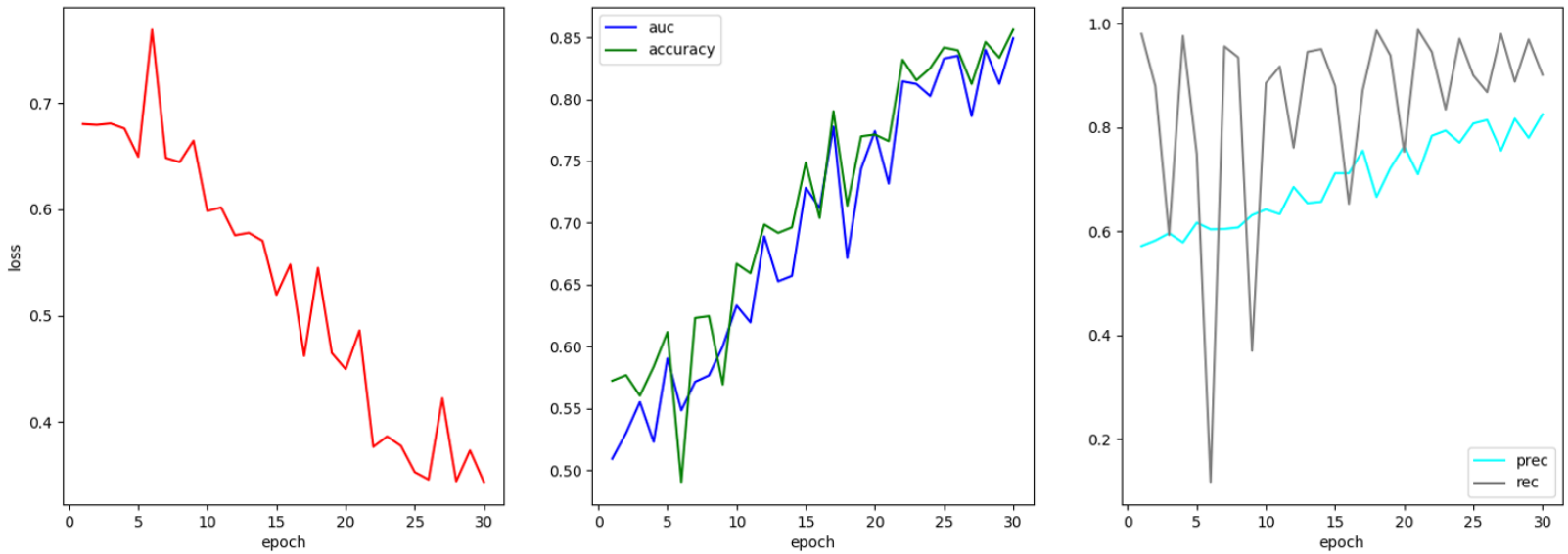


- The 3D-CNN representation implicitly models pairwise relationships between atoms through the relative positioning of atoms in a 3D voxel grid.
- Task: Binary Classification of correct (rmsd < 2) or incorrect (rmsd >4) binding based on protein docking data
- Data: $\{x, y\}_{\mathbf{1}}^{n}$, x, protein docking data, each of protein id and pose combination contains about 500 atoms, build a $48 \times 48 \times 48 \times 22$ voxel, where first 3 of last dimension are used to identify subspace in $\mathcal{R}^{\mathbf{3}}$ to bound boxes and last 19 are features. y: labeling of rmsd (0,1)
- loss function: softmax cross-entropy

$$\xi(T, Y) = -\sum_{i=\mathbf{1}}^{n}\sum_{c=\mathbf{1}}^{C} t_{ic} \cdot log(y_{ic}) \tag{1}$$

3D-CNN Model

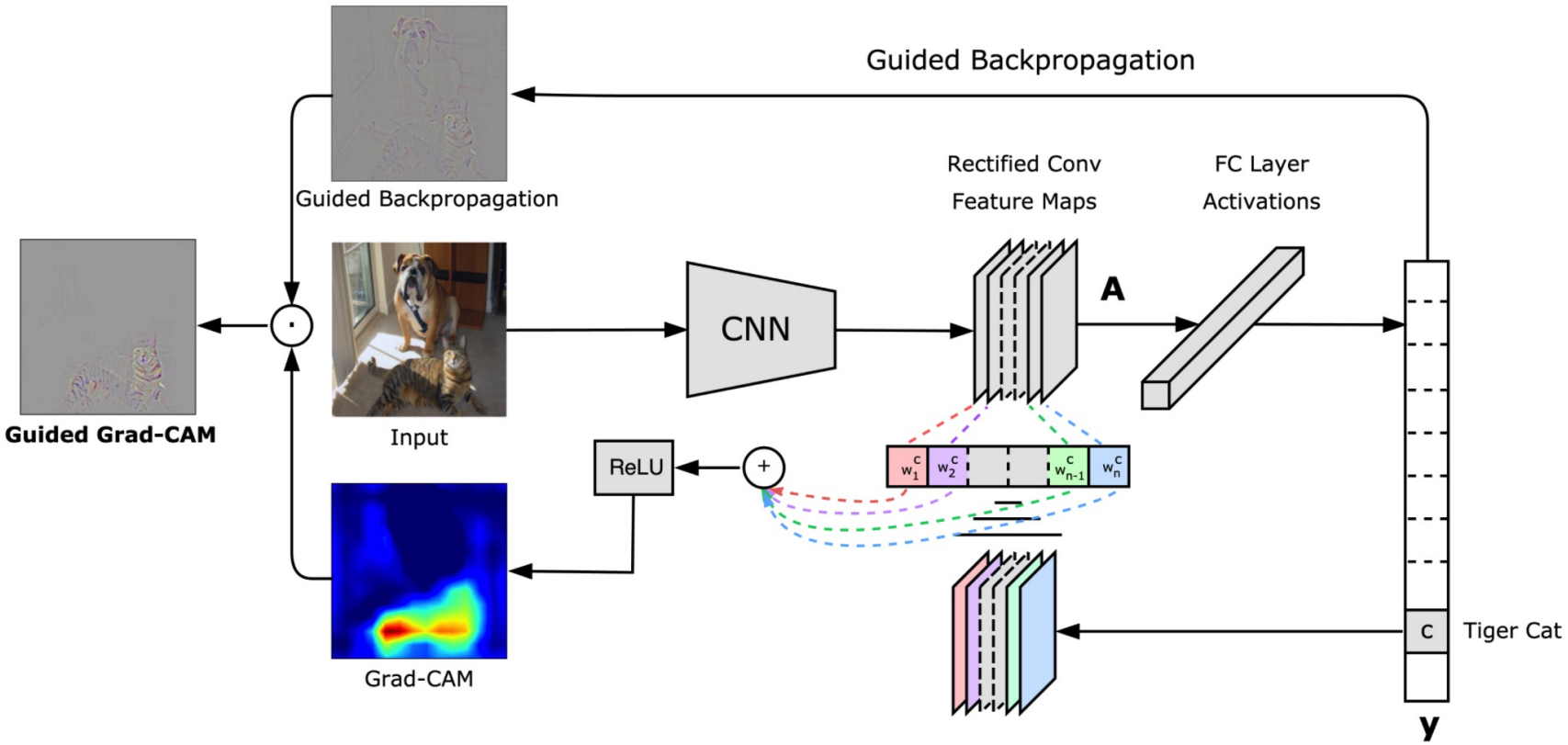
The model was trained in the LLNL-LC high-performance PCs for 12 hours.



	Error	Accuracy	AUC
Testing Performance	0.1401	0.8599	0.8560

Grad-CAM

- CAM (Class Activation Mapping)
- Grad-CAM



GCAM-++ Method

GCAM-++ pixel-wise weighting of the gradients of the output w.r.t. a particular spatial position in the final convolutional feature map of the CNN. Chattopadhyay *et al.* [2018]

$$w_k^c = \sum_i \sum_j [\frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}}] \circ relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

$$\frac{\partial Y^c}{\partial A_{ij}^k} = Y^c[\frac{\partial S^c}{\partial A_{ij}^k} - \sum_k Y^k \frac{\partial S^k}{\partial A_{ij}^k}]$$

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = Y^c[\frac{\partial S^c}{\partial A_{ij}^k} - \sum_k Y^k \frac{\partial S^k}{\partial A_{ij}^k}] - Y^c(\sum_k \frac{\partial Y^k}{\partial A_{ij}^k} \frac{\partial S^k}{\partial A_{ij}^k})$$

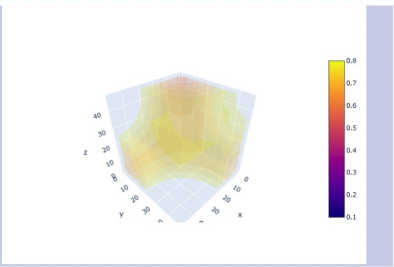
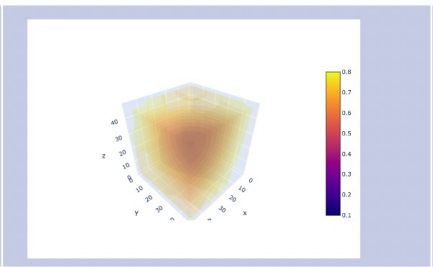
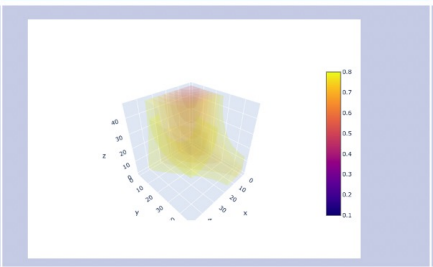
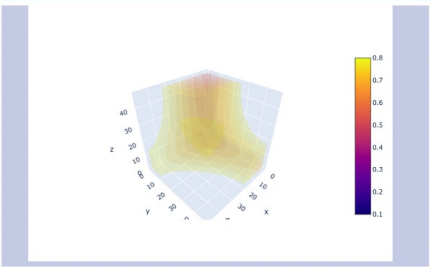
$$\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} = Y^c[\frac{\partial S^c}{\partial A_{ij}^k} - \sum_k Y^k \frac{\partial S^k}{\partial A_{ij}^k}] - 2Y^c(\sum_k \frac{\partial Y^k}{\partial A_{ij}^k} \frac{\partial S^k}{\partial A_{ij}^k}) - Y^c(\sum_k \frac{\partial^2 Y^k}{\partial A_{ij}^k}^2 \frac{\partial S^k}{\partial A_{ij}^k})$$

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k}$$

$$L_{ij}^c = \sum_k w_k^c A_{ij}^k \quad \textbf{Final Class Discriminative Salinecy Map}$$

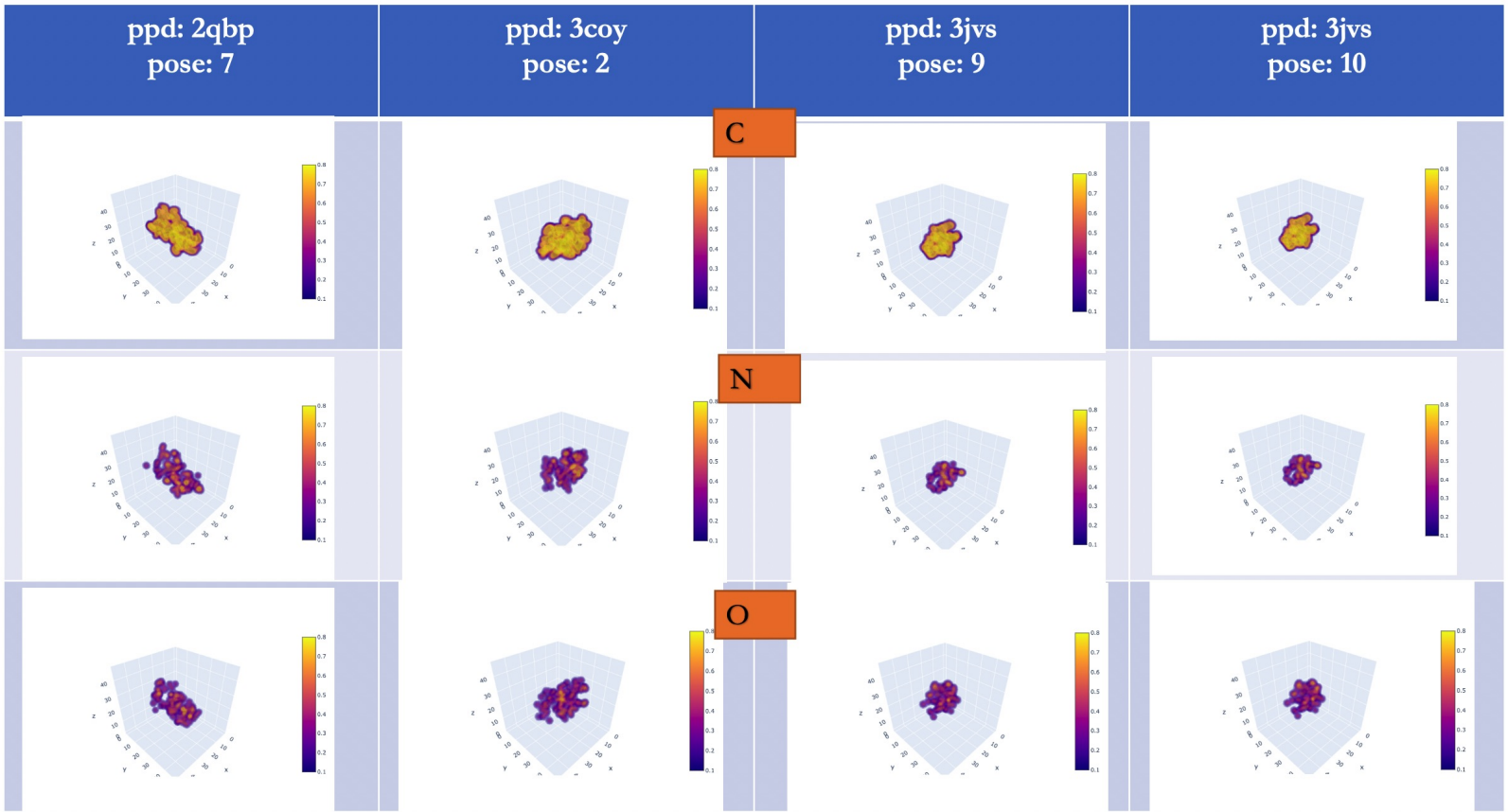
GCAM-++ result

Which part of last convolutional layer the classifier is most sensitive to?

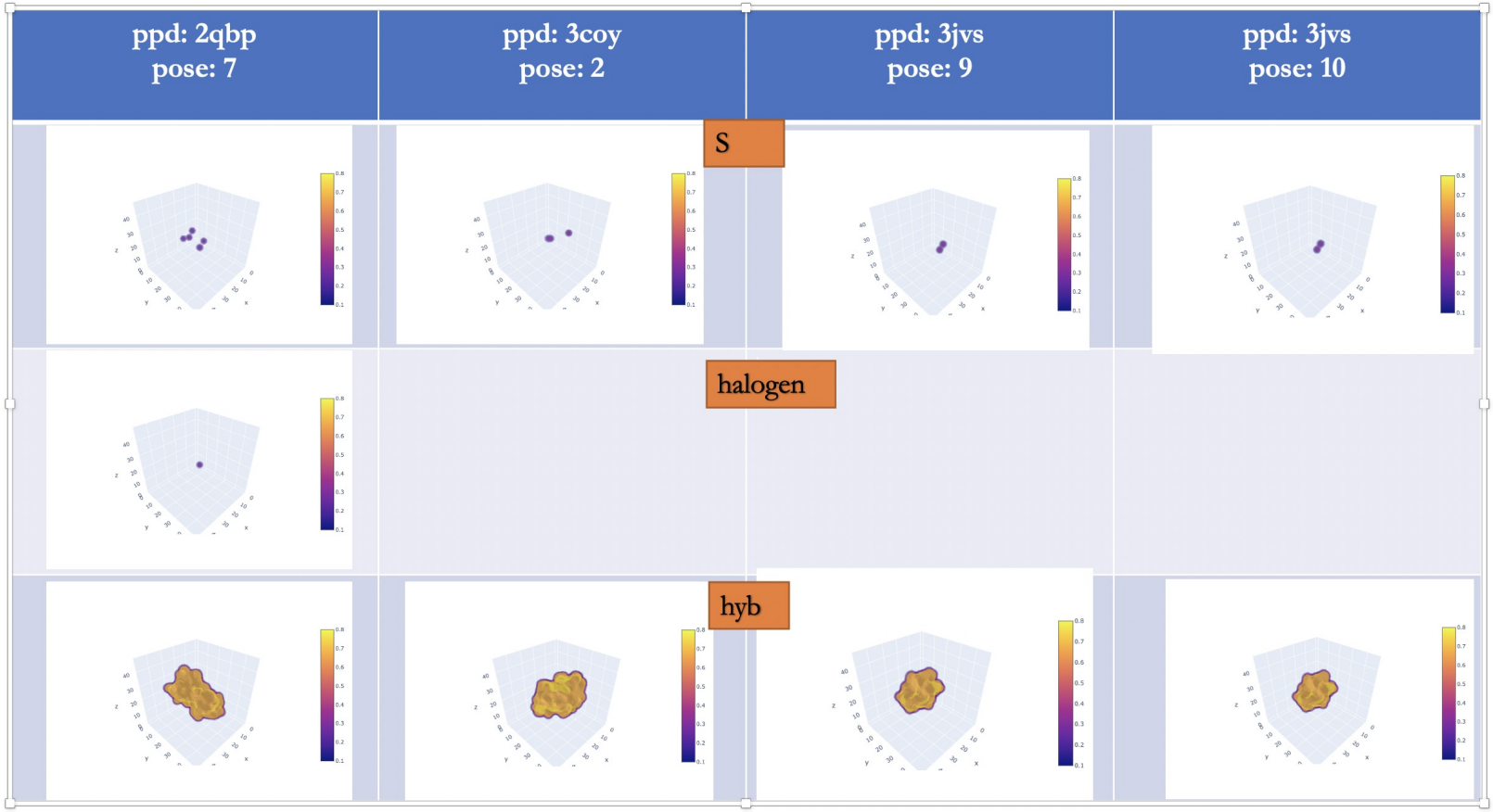
ppd: 2qbp pose: 7	ppd: 3coy pose: 2	ppd: 3jvs pose: 9	ppd: 3jvs pose: 10
			

GCAM-++ result

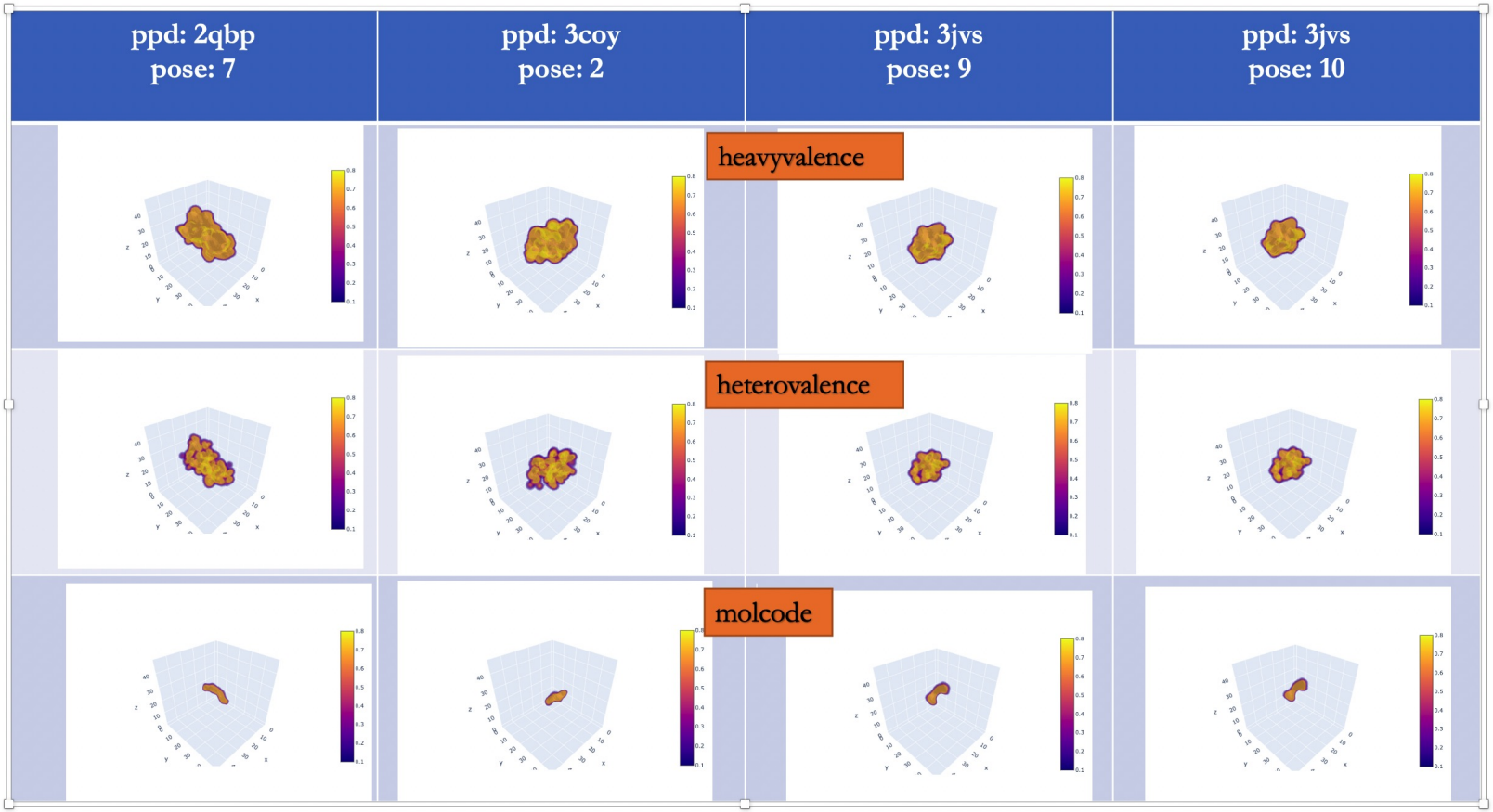
Segment by input features



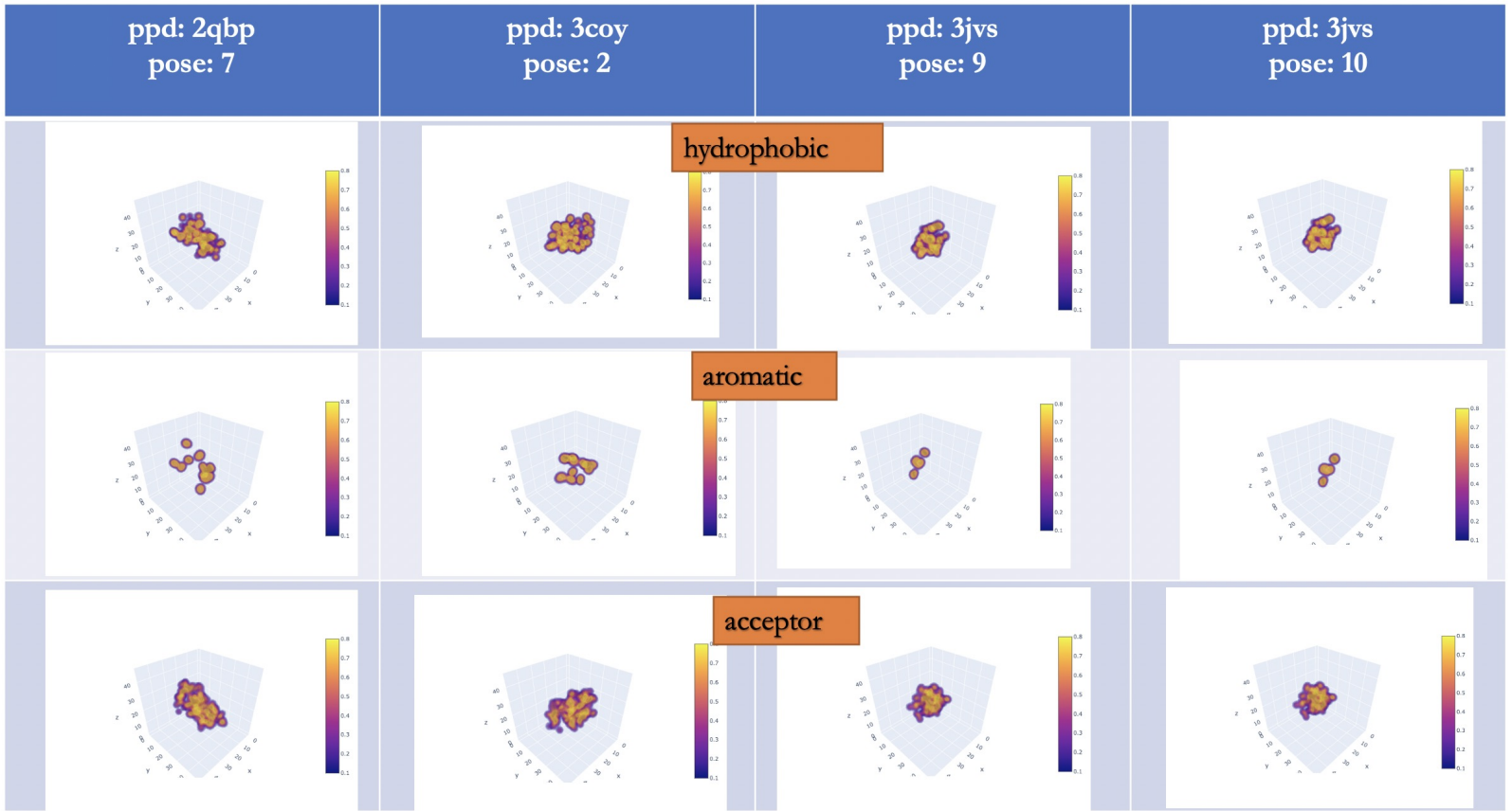
GCAM-++ result


ppd: 2qbp
pose: 7
ppd: 3coy
pose: 2
ppd: 3jvs
pose: 9
ppd: 3jvs
pose: 10
S
halogen
hyb

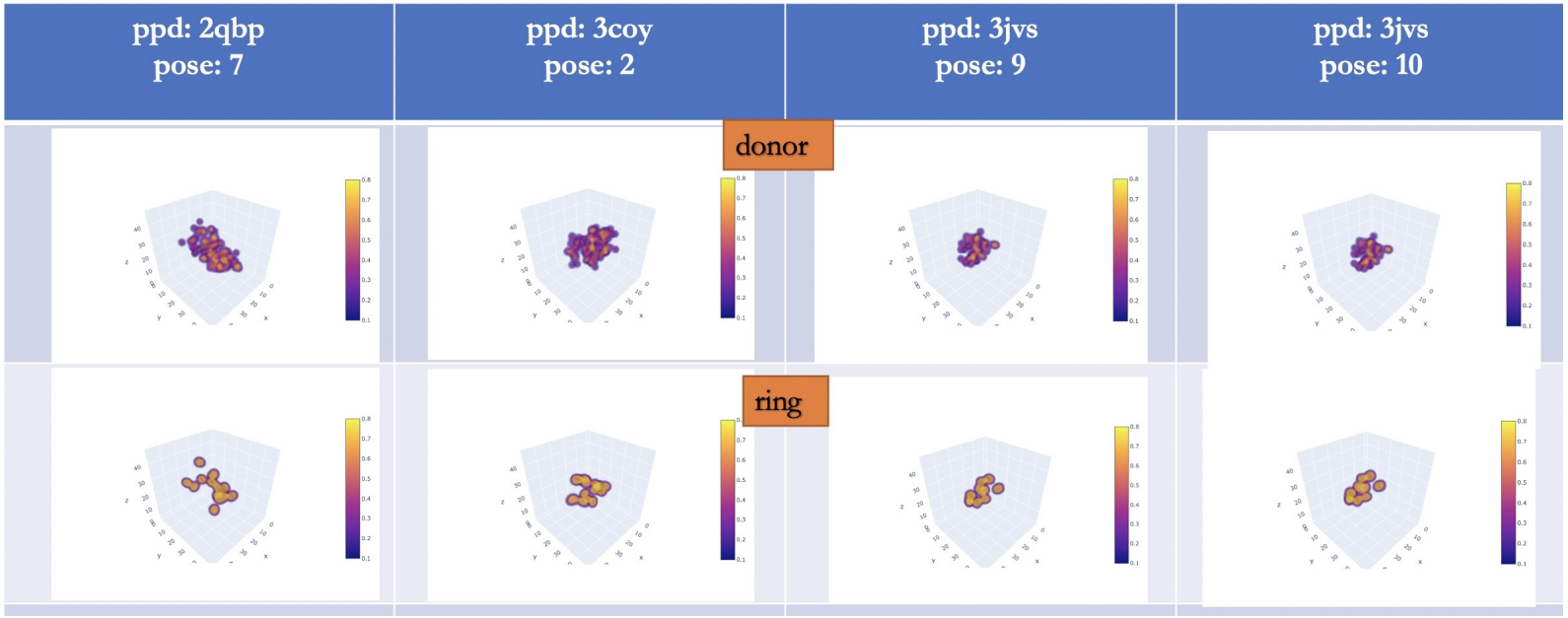
GCAM-++ result


ppd: 2qbp
pose: 7
ppd: 3coy
pose: 2
ppd: 3jvs
pose: 9
ppd: 3jvs
pose: 10
heavyvalence
heterovalence
molcode

GCAM-++ result

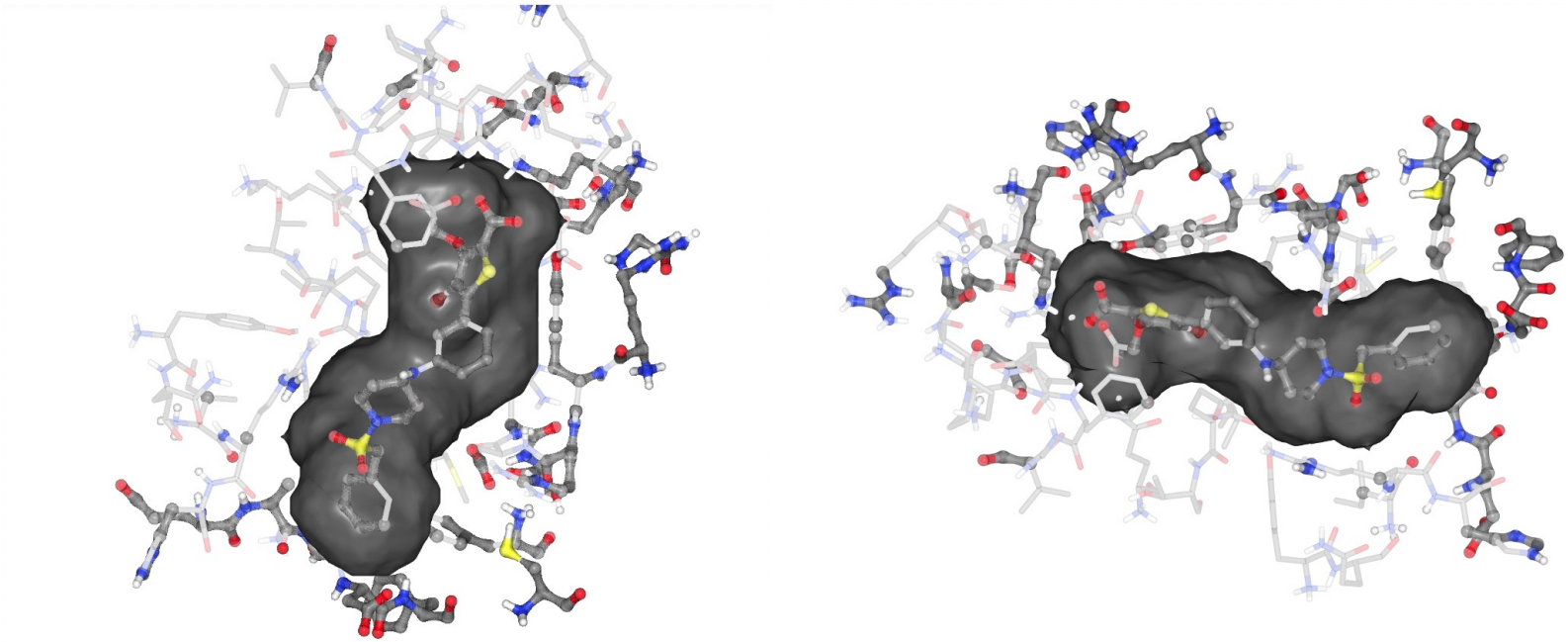

ppd: 2qbp
pose: 7
ppd: 3coy
pose: 2
ppd: 3jvs
pose: 9
ppd: 3jvs
pose: 10
hydrophobic
aromatic
acceptor

GCAM-++ result


ppd: 2qbp
pose: 7
ppd: 3coy
pose: 2
ppd: 3jvs
pose: 9
ppd: 3jvs
pose: 10
donor
ring

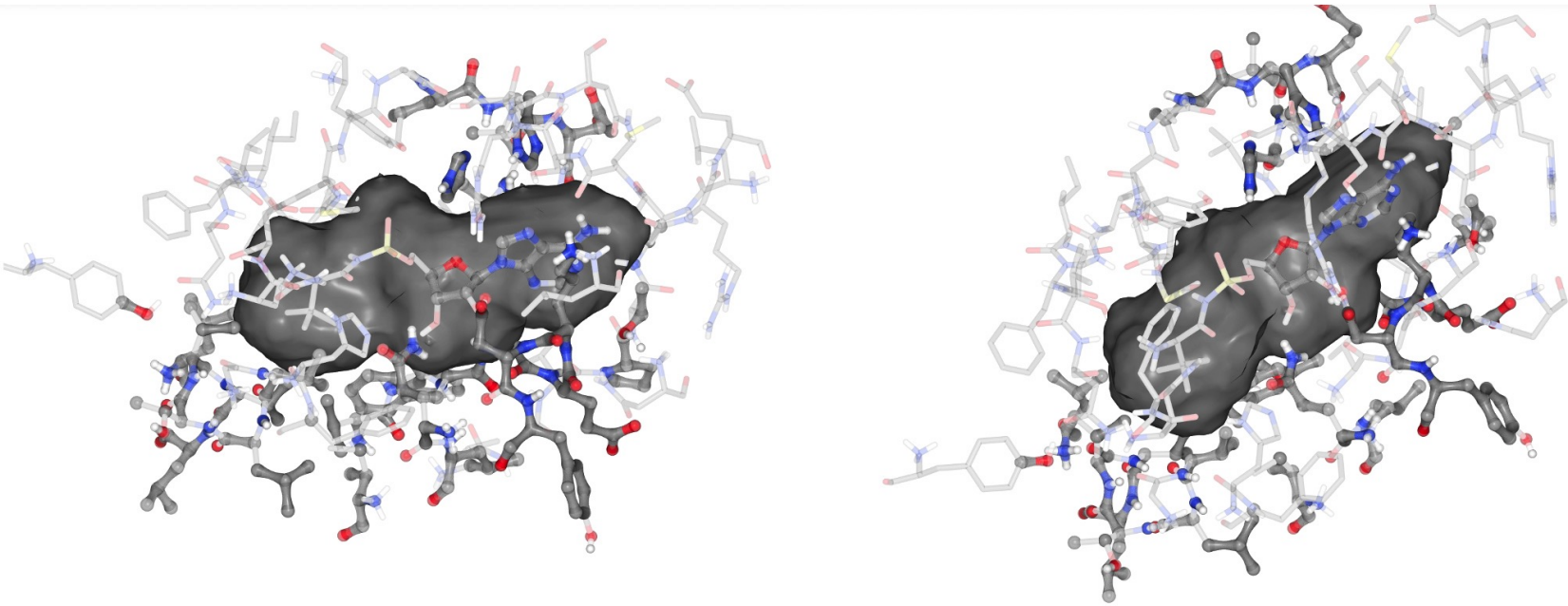
Overlay heatmap in NGLview

- gray surface area: "ligand'
- blue: nitrogen
- red: oxygen
- yellow: sulfur
- white: hydrogen
- gray: carbon



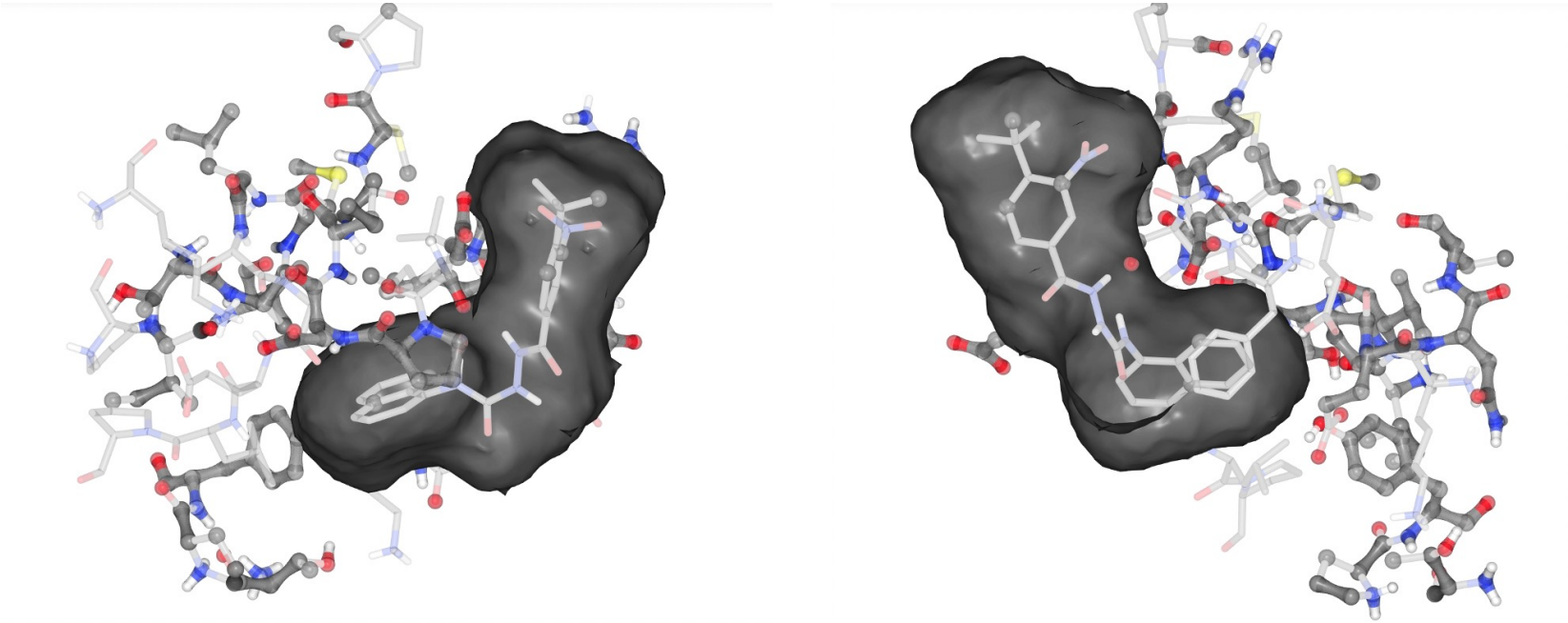
2qbp, 07 (label: 1, pred: 1)

Overlay heatmap in NGLview



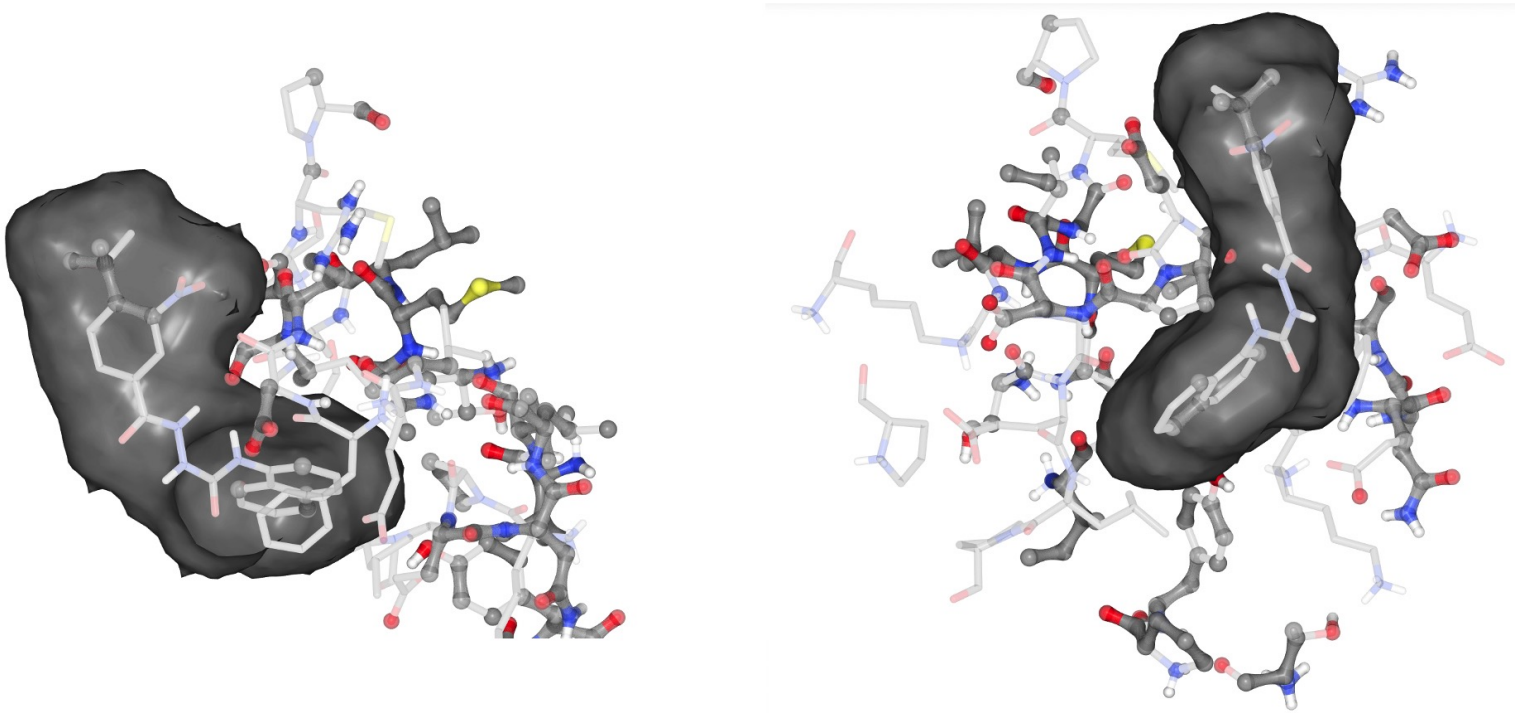
3coy, 02 (label: 1, pred: 1)

Overlay heatmap in NGLview



3jvs, 09 (label: 0, pred: 1)

Overlay heatmap in NGLview



3jvs, 10 (label: 0, pred: 1)

Quantitative Testing with Concept Activation Vectors

A CAV (concept activation vector) for a concept is simply a vector in the direction of the values (e.g., activation) of that concept's set of examples.

- **CAVs** are derived by training a linear classifier between a concept's examples and random counterexamples and then taking the vector orthogonal to the decision boundary.
- Consider neural network models with inputs $x \in R^n$ and a feedforward layer l with m neurons, such that input inference and its layer l activations can be seen as a function: $f_l : R^n \rightarrow R^m$
- A classifier $v_c^l \in R^m$ is a linear CAV for the concept C.
- We can gauge the sensitivity of ML predictions to changes in inputs towards the direction of a concept, at neural activation layer l.
- If $v_c^l \in R^m$ is a unit CAV vector for a concept C in layer l, and $f_l(x)$ the activations for input x at layer l, the "conceptual sensitivity" of class k to concept C can be computed as the directional derivative:

$$S_{s,k,l}(x) = lim_{\epsilon \rightarrow \mathbf{0}} \frac{h_{l,k}(f_l(x) + \epsilon v_c^l) - h_{l,k}(f_l(x))}{\epsilon}$$
$$= \nabla h_{l,k} f_l(x) \cdot v_c^l$$

- $S_{s,k,l}(x)$ can quantitatively measure the sensitivity of model predictions with respect to concepts at any model layer [Ghorbani *et al.*, 2019; Kim *et al.*, 2018]

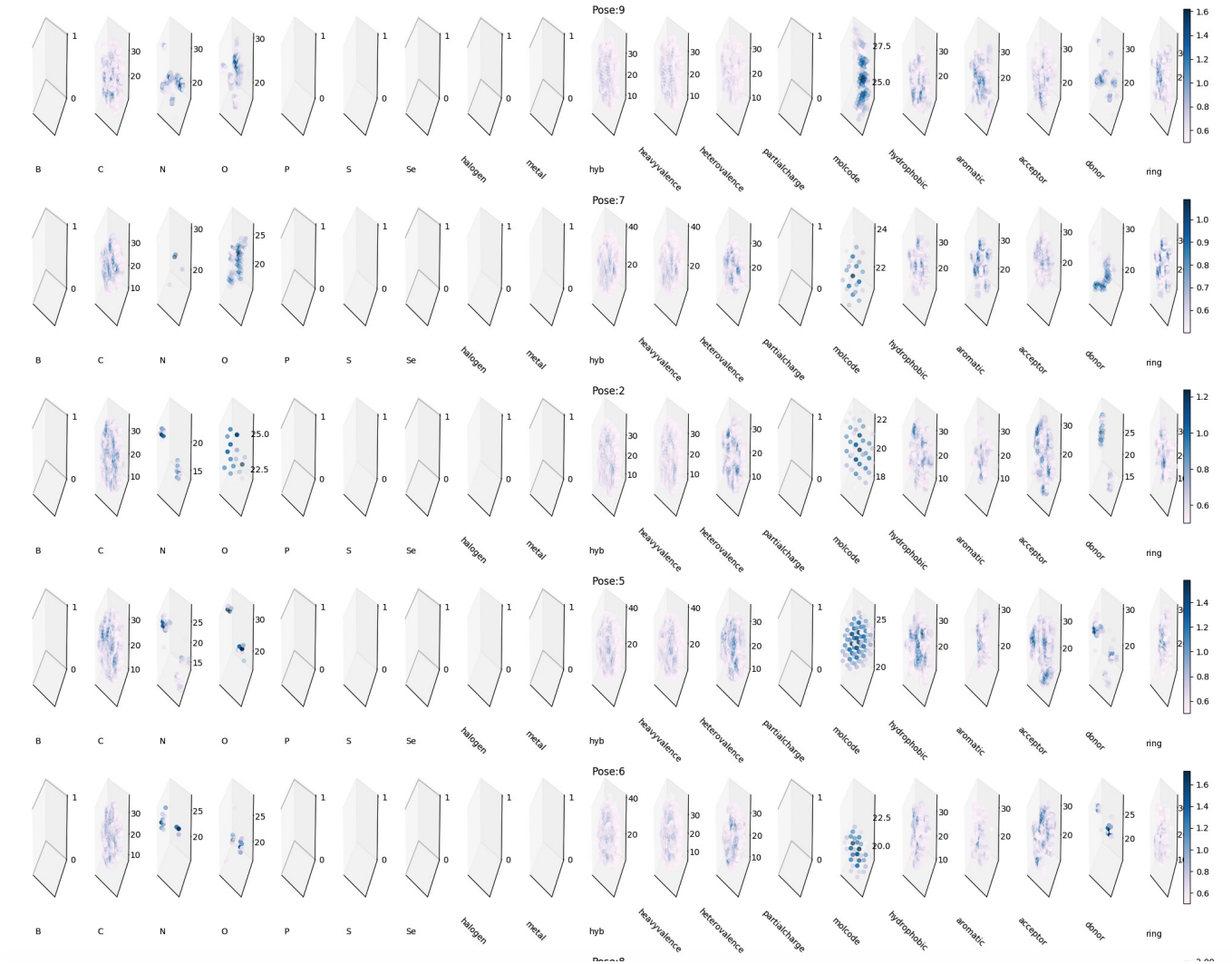
TCAV Method

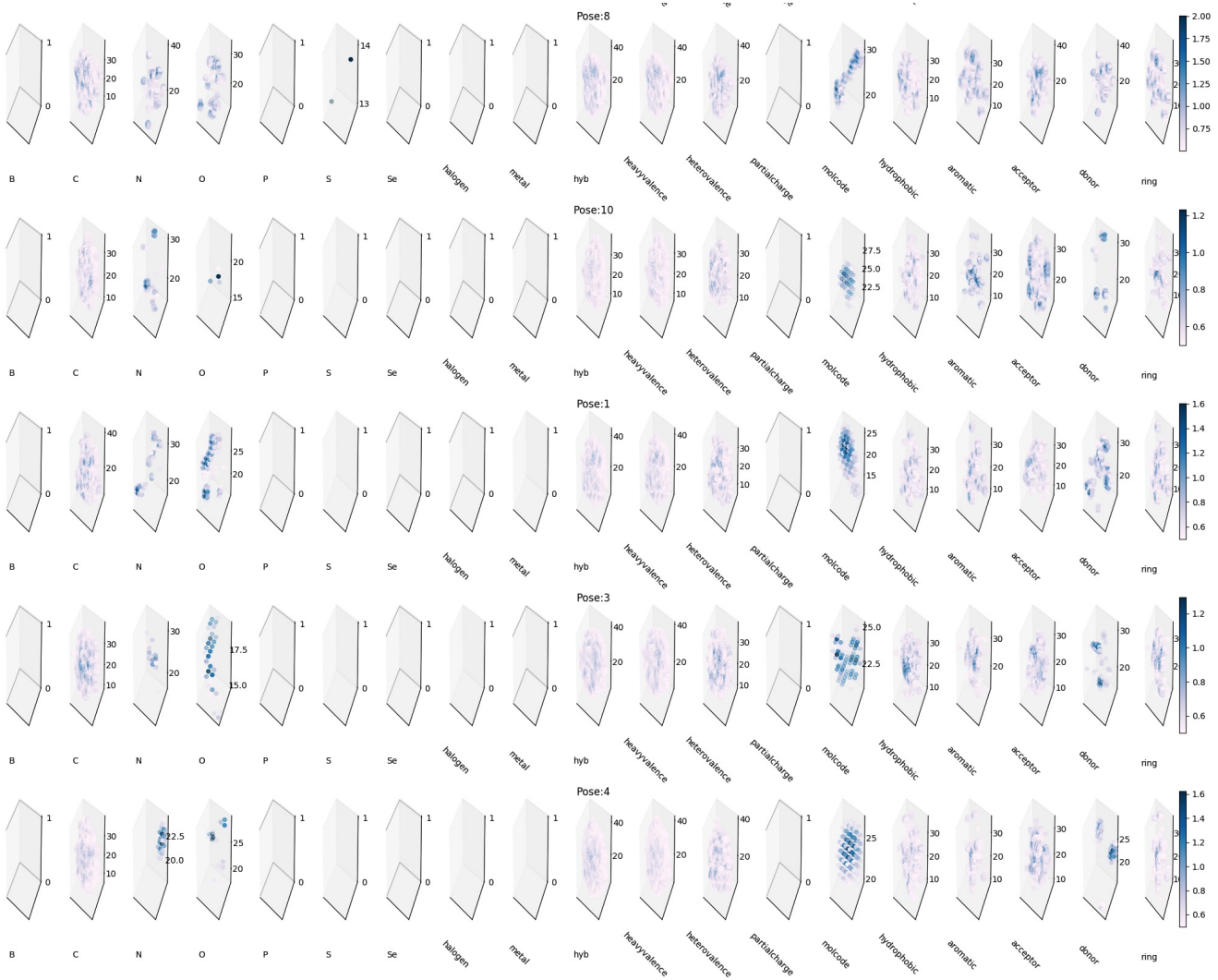
- Testing with TCAV: uses directional derivatives to compute ML models' conceptual sensitivity across entire classes of inputs.
- Let k be a class label for a given supervised learning task and let X_k denote all inputs with that given label. The TCAV score will be:

$$TCAV_{c,k,l} = \frac{|\{x \in X_k : S_{c,k,l}(x) > 0\}|}{|X_k|}$$

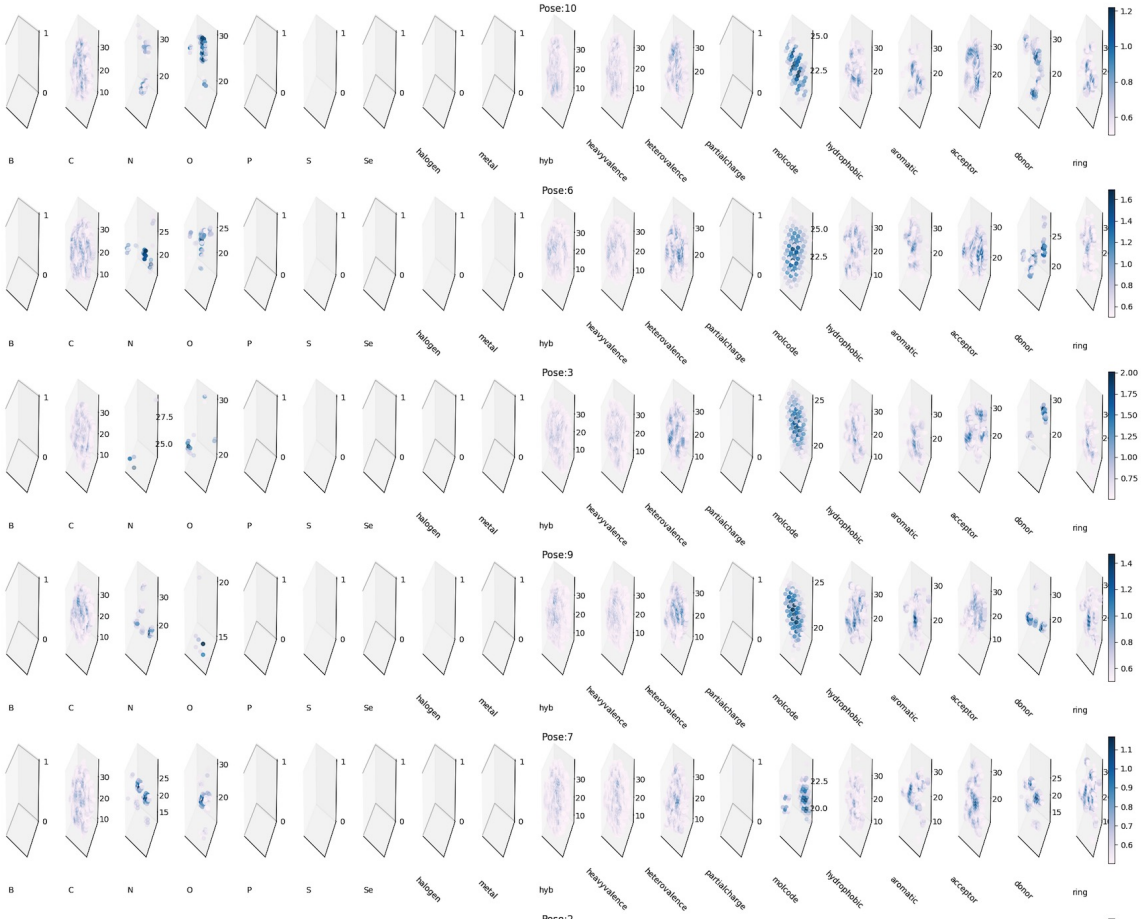
- Task: to identify the sensitivity of poses of protein to 3dcnn model for prediction of the protein binding at the last convolutional layer
- Concepts: different docking poses of protein
- Target class: binding correctly vs not correctly
- Model: 3D-CNN model

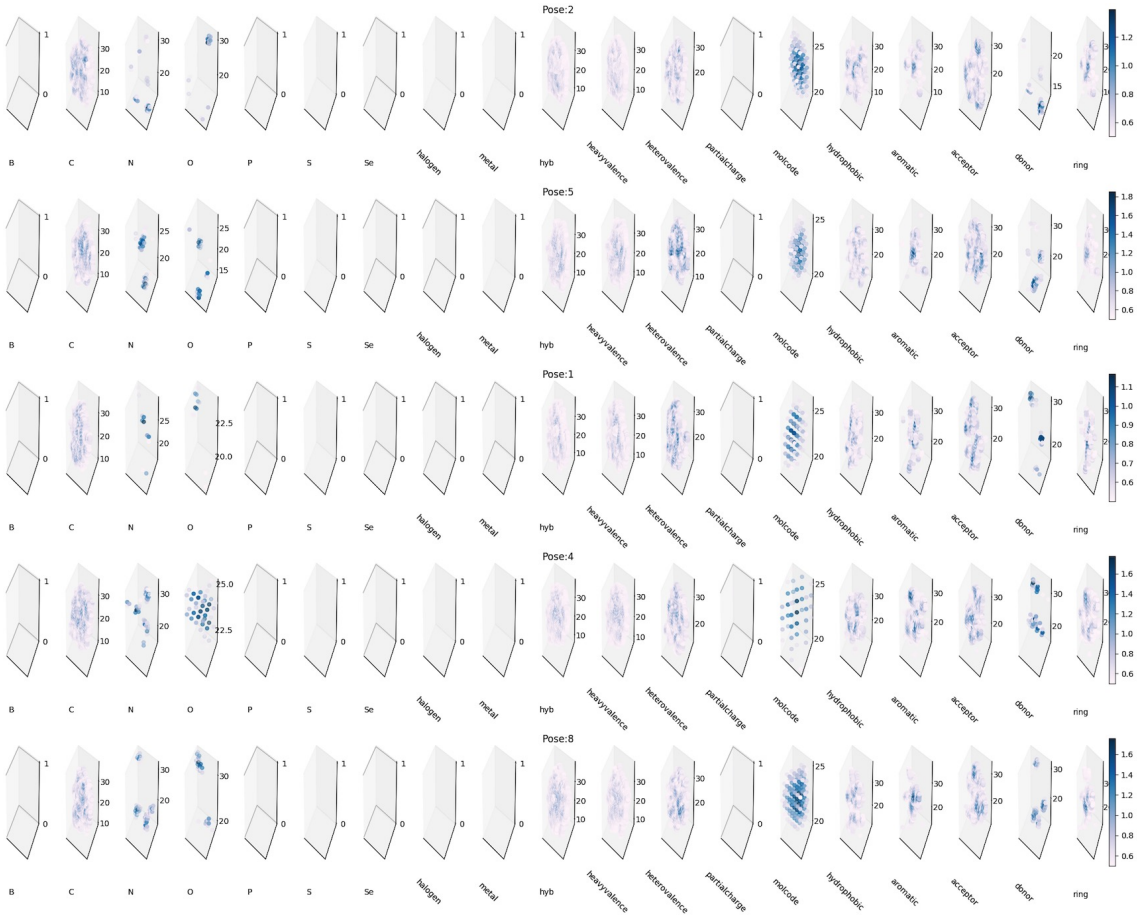
3D-CNN Last Convolutional Layer, $k := 1$




Pose:8
Pose:10
Pose:1
Pose:3
Pose:4
B
C
N
O
P
S
Se
halogen
metal
hyb
heavyvalence
heterovalence
partialcharge
molcode
hydrophobic
aromatic
acceptor
donor
ring

$k := 0$


Pose:10
Pose:6
Pose:3
Pose:9
Pose:7
B
C
N
O
P
S
Se
halogen
metal
hyb
heavyvalence
heterovalence
partialcharge
molcode
hydrophobic
aromatic
acceptor
donor
ring



	Error	Accuracy	AUC
Full Atom Features	0.1401	0.8599	0.8560
Selected Atom Features	0.1340	0.8660	0.8590

Reference

Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-CAM++: Improved Visual Explanations for Deep Convolutional Networks. *arXiv:1710.11063 [cs]*.

Ghorbani, A., Wexler, J., Zou, J., and Kim, B. (2019). Towards Automatic Concept-based Explanations. *arXiv:1902.03129 [cs, stat]*.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. (2018). Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *arXiv:1711.11279 [stat]*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, **128**(2), 336–359.