

# Designing Machine learning surrogates for molecular dynamics simulations

JCS Kadupitiya, GC Fox, V Jadhao

Intelligent Systems Engineering

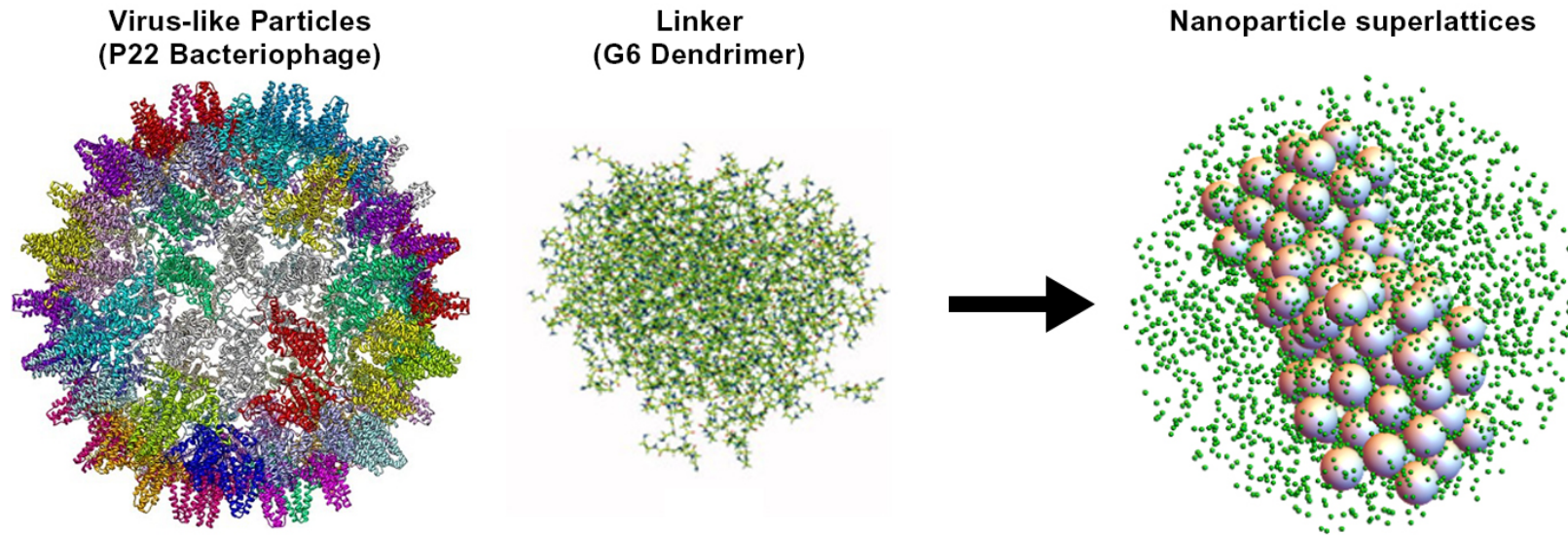
Luddy School of Informatics, Computing, and Engineering

Indiana University, Bloomington



ENGINEERED  
**nanoBIO**  
AN INDIANA UNIVERSITY RESEARCH NODE

# Introduction



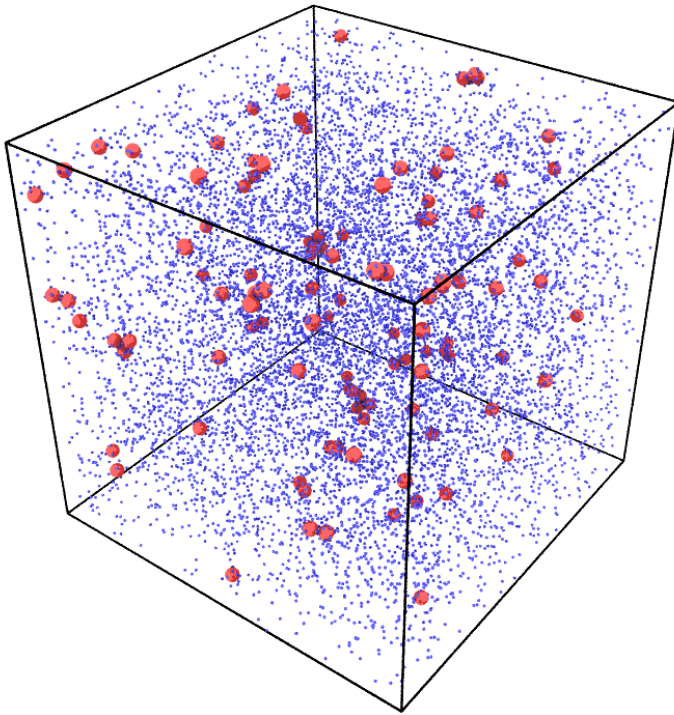
- Molecular dynamics (MD) simulations are powerful tools for investigating the microscopic origins of the behavior of a wide range of materials, including soft matter
  - Simulations are used everywhere; physics, chemistry, bioengineering, and materials science.
- Simulations enable the understanding of microscopic mechanisms underlying the macroscopic material and biological phenomena.
- Parallel computing techniques (OpenMP, MPI) are often used with complex systems.

# Few examples for MD simulations

## Virus like particle - Linker superlattices simulation

wall time: ~4 hr

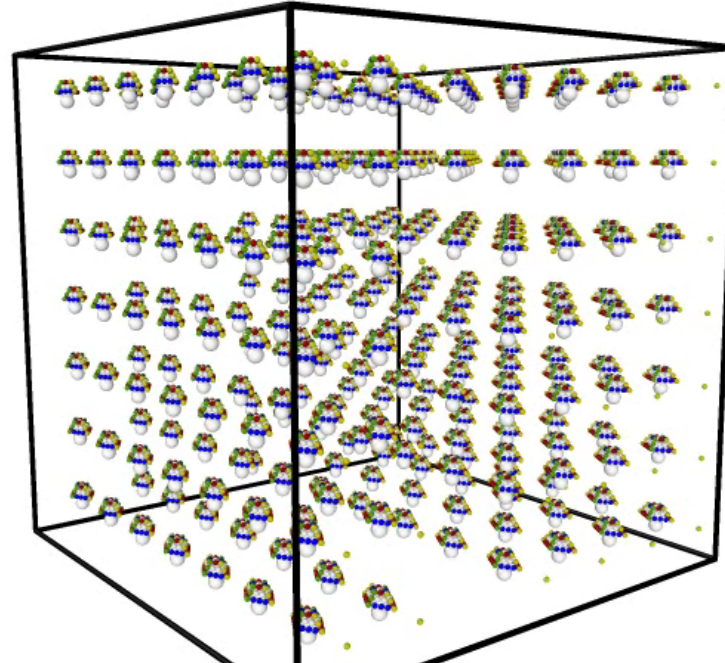
# of cores: 4 nodes 24 cores



## 12-capsomere virus nanoparticle assembly

wall time: ~ days

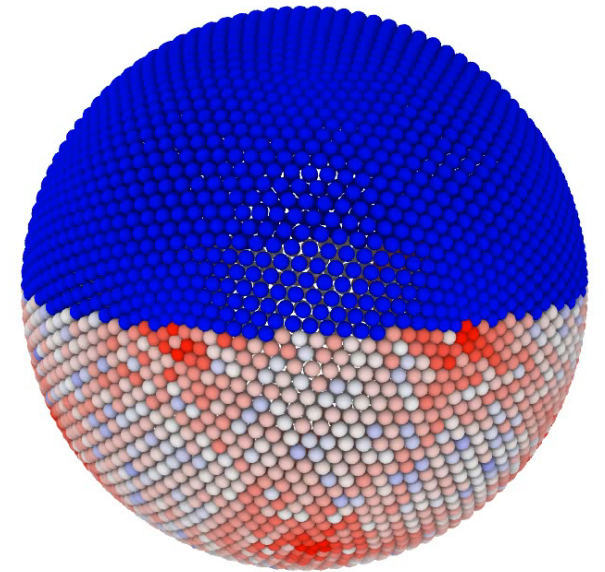
# of cores: 4 nodes 24 cores



## Shape control of charge- patterned nano-containers

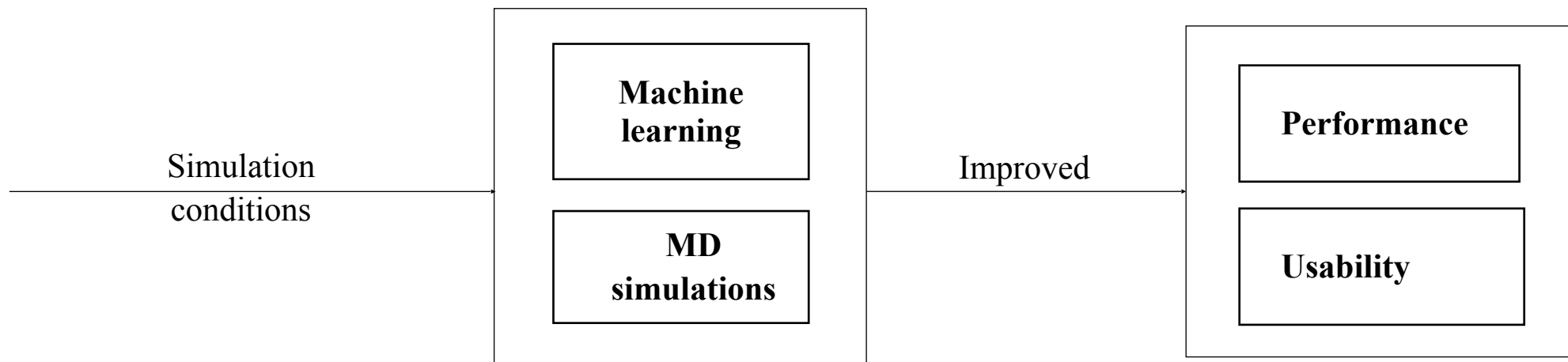
wall time: ~1 hr

# of cores: 2 nodes 24 cores



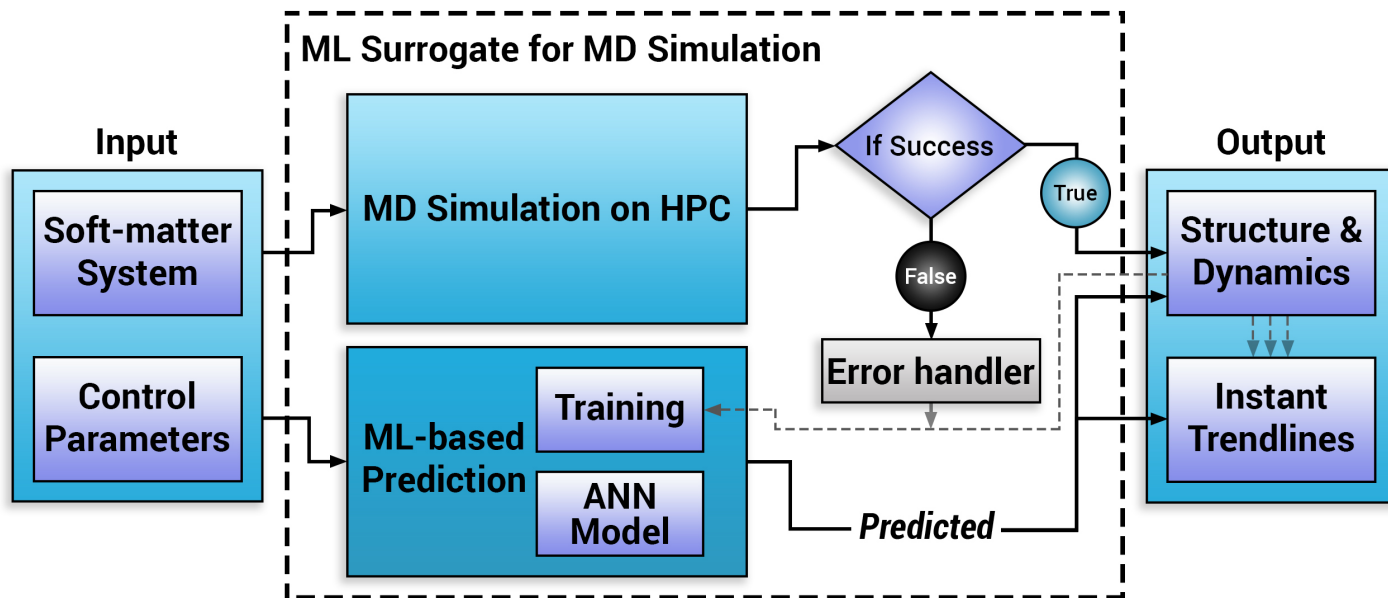
# Why Machine Learning cont.

- In classroom usage, fastest simulations can take about 10 minutes to 3 hours.
- Similarly, for research applications, not having a rapid access to expected overall trends can make the process of starting new investigations unwieldy and time-consuming (waiting + runtime).
- we explore the idea of integrating machine learning (ML) layer with simulations to enhance the performance and improve the usability of simulations for both research and education.



# Machine Learning Surrogates for MD Simulations

- The “ML surrogates for MD simulations” framework is an approach to use ML to learn from MD simulations and produce learned surrogates for MD simulations.
- ML surrogates for MD simulations enable several capabilities:
  - learn pre-identified (desired) features associated with the simulation outputs
  - generate accurate predictions for unseen design space parameters
  - enable instantaneous predictions and improve interactivity



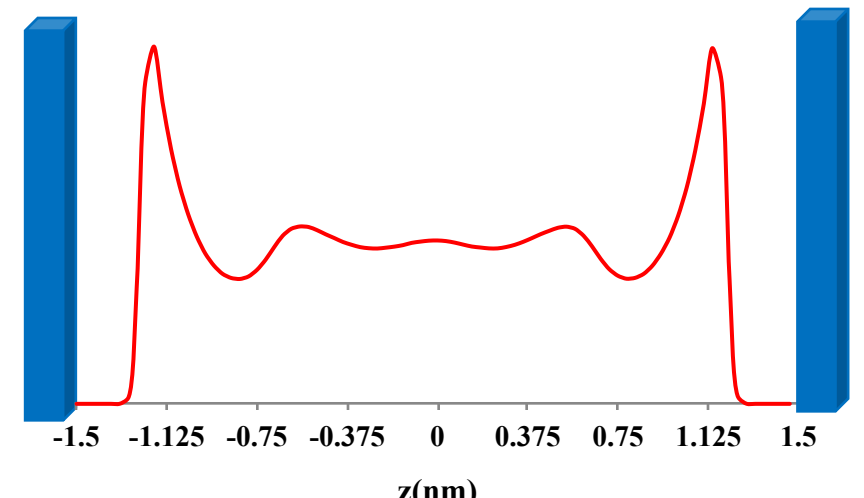
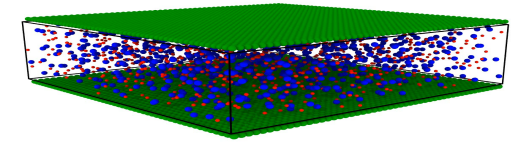
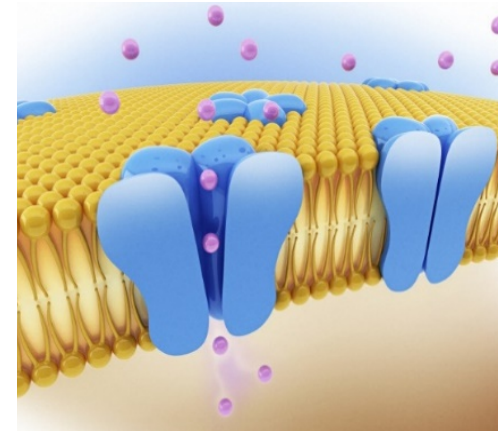
## Related Work:

- SorbNet (2019): DNN surrogate for adsorption equilibria
- Chem. Sci. (2019): ANN to predict dissociation timescale of compounds in ab initio MD simulations
- arXiv:2001.08055 (2020): CNN based “emulators” to predict outcomes of simulations in biochemistry

# Application

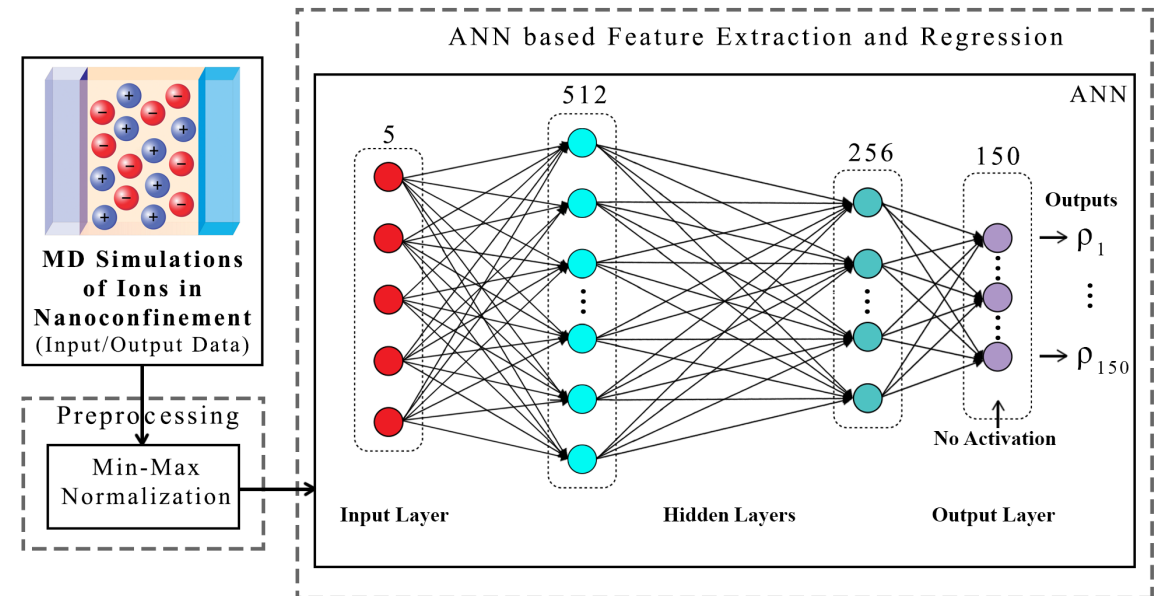
- Apply the ML surrogate idea to the case of MD simulations of ions in nanoconfinement created by uncharged material surfaces.
- Goal: bypass simulations and use ML to extract the distribution of confined ions.
- Inputs: confining length, salt conc., ion valencies, ion diameter
- Outputs: the density profiles of ions

Ion channels



# Artificial Neural Network (ANN) Model for Regression

- ANN-based regression model used in the ML surrogate to predict output ionic density profile
- Generated dataset having **6,864 simulation configurations** for training and testing (0.7:0.3)
- ANN was trained to predict  $\sim 150$  points characterizing (half of) the ion density profile
- ❖ Technologies: TensorFlow, Keras and Sklearn
- ❖ Implementation details: Adam optimizer, Xavier normal distribution, mean square loss function, dropout regularization.



- **Inputs:** confining length, salt conc., positive ion valency, negative ion valency, and ion diameter
- **Outputs:** the density profiles of ions

# Results

- ANN based regression model predicted Contact density  $\rho_c$  , mid-point (center of the slit) density  $\rho_m$  , and peak density  $\rho_p$  accurately with a success rate of 95:52% (MSE  $\sim$  0:0000718), 92:07% (MSE  $\sim$  0:0002293), and 94:78% (MSE  $\sim$  0:0002306) respectively outperforming other non-linear regression models

Model	Contact Density		Midpoint Density		Peak Density	
	<i>Success %</i>	<i>MSE</i>	<i>Success %</i>	<i>MSE</i>	<i>Success %</i>	<i>MSE</i>
Polynomial	61.04	0.0129300	60.84	0.0187700	61.87	0.0100400
Kernel-Ridge	78.86	0.0030900	76.57	0.0041200	75.93	0.0049800
Support Vector	80.11	0.0012700	79.55	0.0024900	81.98	0.0010600
Decision Tree	68.44	0.0084600	64.54	0.0094900	62.47	0.0110700
Random Forest	74.15	0.0045700	70.85	0.0078900	75.09	0.0040800
ANN based	95.52	0.0000718	92.07	0.0002293	94.78	0.0002306



# Results: Accuracy Comparison

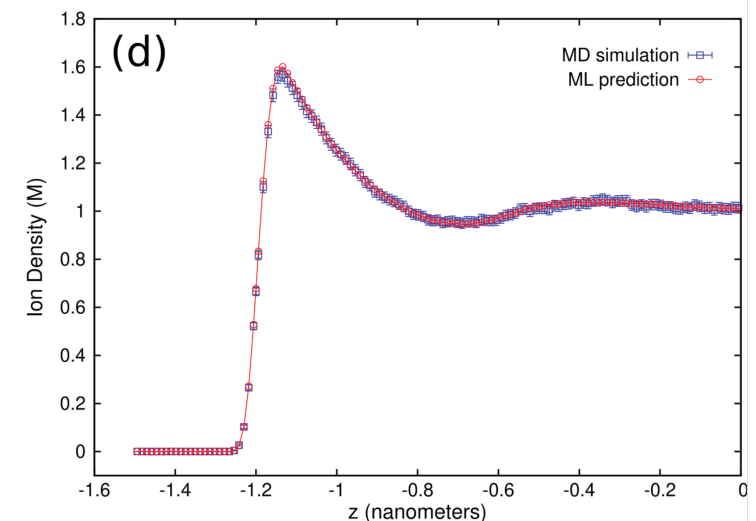
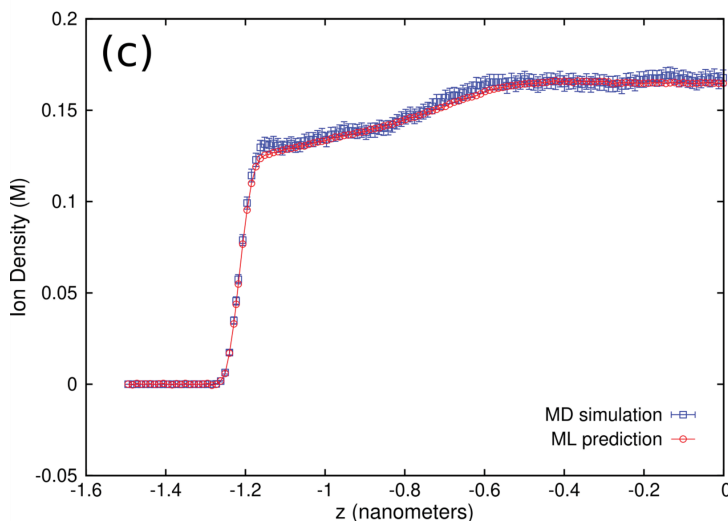
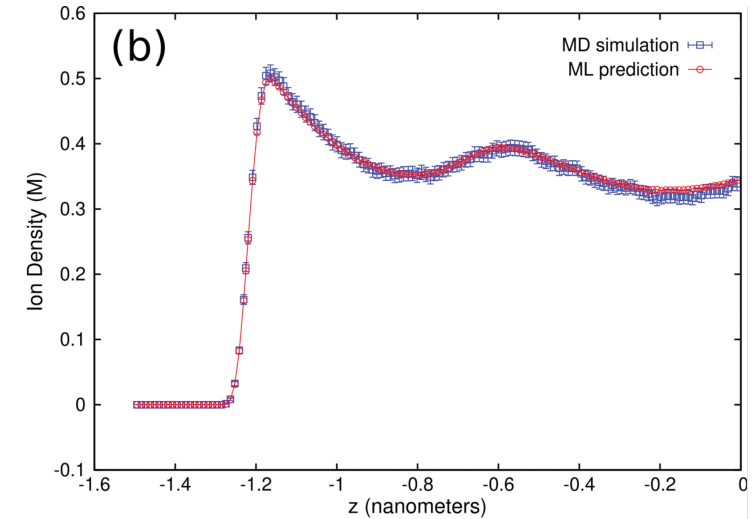
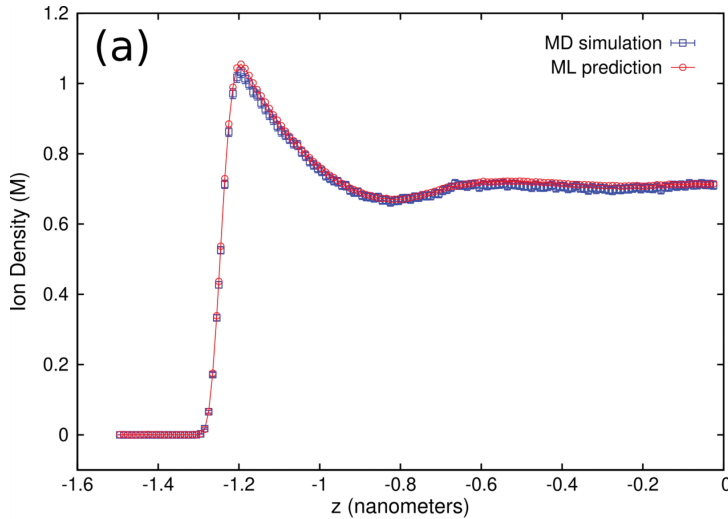
- The success rates  $A_i$  for systems a, b, c, and d are found to be 0.98, 0.91, 0.78, 0.89 respectively.

$$A_i = \frac{1}{P} \sum_{n=1}^P \Theta \left( \left| \rho_{n,i}^{\text{ML}} - \rho_{n,i}^{\text{MD}} \right|, \epsilon_{n,i} \right)$$

- where  $i$  indicates the simulation index,

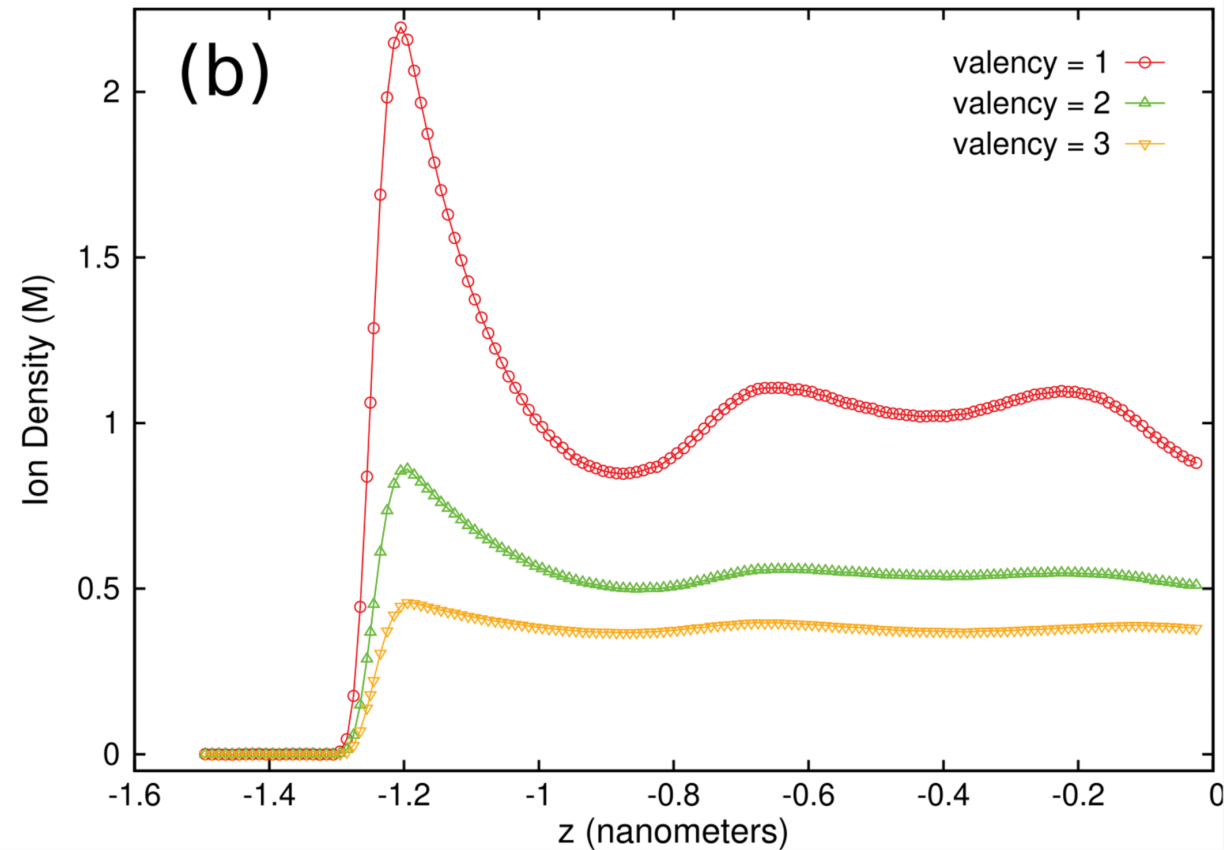
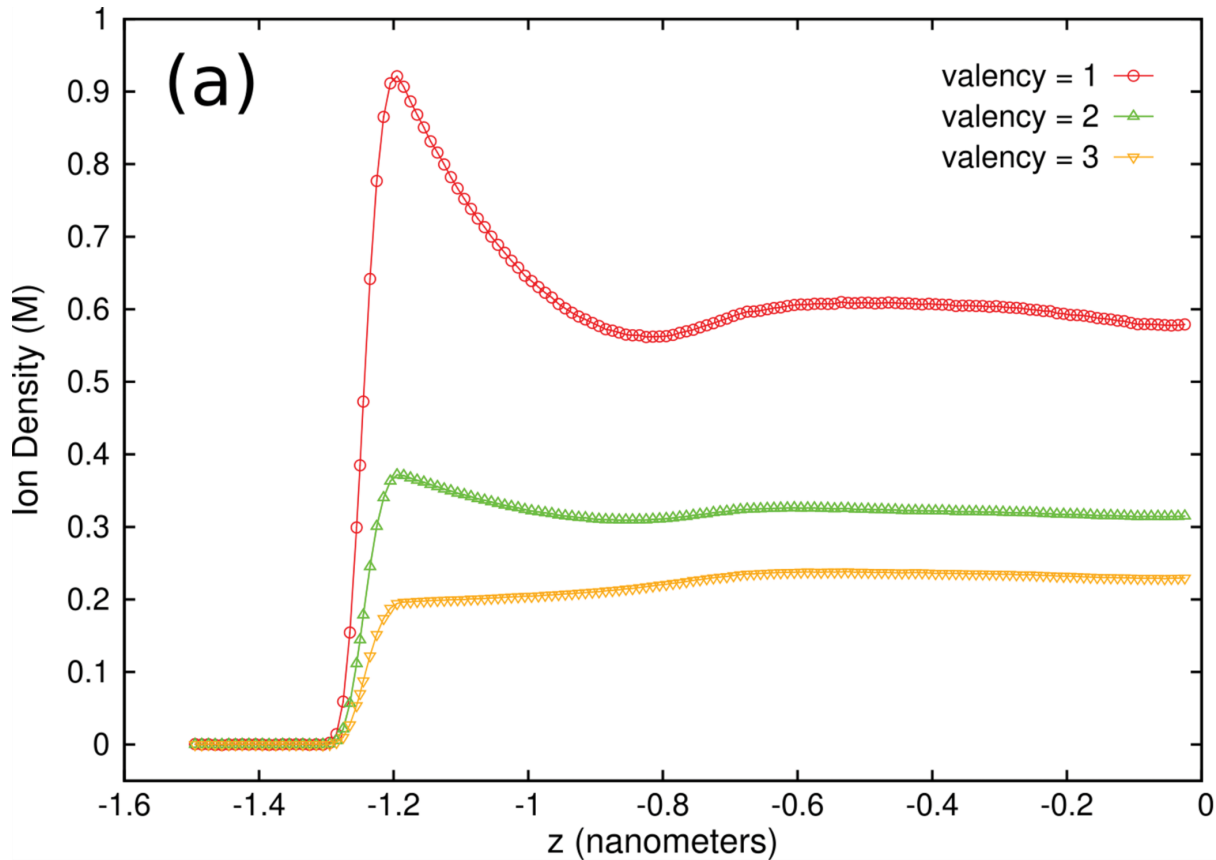
- $P$  is the number of predictions made

- $\Theta(\mathbf{x}, \boldsymbol{\epsilon})$  is a step function given by  $\Theta(\mathbf{x}, \boldsymbol{\epsilon}) = 1$  for  $\mathbf{x} < \boldsymbol{\epsilon}$ , and  $\Theta(\mathbf{x}, \boldsymbol{\epsilon}) = 0$  for  $\mathbf{x} \geq \boldsymbol{\epsilon}$ .



# Results: Trendlines generated using ML surrogate

- Trendlines generated using ML surrogate to examine variation in ionic density with positive ion valency at salt concentration (a)  $c = 0.5$  M and (b)  $0.9$  M.



# Results: Speedup (S)

- Traditional speedup formulae associated with parallel computing methods need to be adapted for evaluating the speedup associated with the ML-enhanced simulations.

$$S_{ML} = \frac{t_{sim}}{t_p + t_{tr} \cdot N_{tr} / N_p}$$

- S rises with increasing  $N_p$
- When  $N_{tr} \ll N_p$ , number of S becomes  $t_{sim} / t_p$  ; For our MD simulations ( $t_{sim} \sim 12$  hours) and ANN model ( $t_p \sim 0:25$  seconds), we find this ratio to be over **10<sup>5</sup>**
- When  $N_{tr}$  (4K)  $>$   $N_p$ (1), For our MD simulations  $t_{tr}$  is 1000s, so **S  $\sim$  10<sup>-2</sup>**
- **S<sub>T</sub> =  $t_{sim} / t_{tr}$**  and  $t_p \sim 0$ , so **S<sub>T</sub> /  $N_{tr}$  = S<sub>ML</sub> /  $N_p$**
- $t_{sim}$  is sequential run time
- $t_p$  is time to do forward propagation/ inference per instance
- $N_p$  number of predictions/looked ups
- $N_{tr}$  number of training samples
- $t_{tr}$  is average MD simulation walltime to create one training sample + average training time per sample
- $N_{tr} * t_{tr}$  represents total time to create the training dataset and the TensorFlow training time.

# Uncertainty Quantification Using ML surrogate


- Uncertainty Quantification (UQ) is a new feature in the nanoconfinement tool.
- Inputs to simulations often have some uncertainty in their values
  - Measurement error, Variations in samples, Etc.
- We need to know how these uncertainties propagate to the output(s)
- It would also be nice to know which uncertainties affect the output(s) the most.
  - This is known as “Sensitivity Analysis”.
- We use a pretrained ML surrogate to run the simulations required for UQ.
- This runs the tool with different combinations of input values using the ML surrogate.
  - Since we are using the ML surrogate, running many runs will be instantaneous.

# UQ Inputs


- Supported Probability Distributions
  - Exact (no UQ) values
  - Gaussian (or Normal) distribution
  - Uniform distribution
- If any number values were set to probability distributions (Gaussian or Uniform), UQ is performed.

## Input controls:


**Salt concentration (M):**

 Distribution: gaussian  Mean: 0.6  Std.: 0.001

**Ion diameter (nm):**

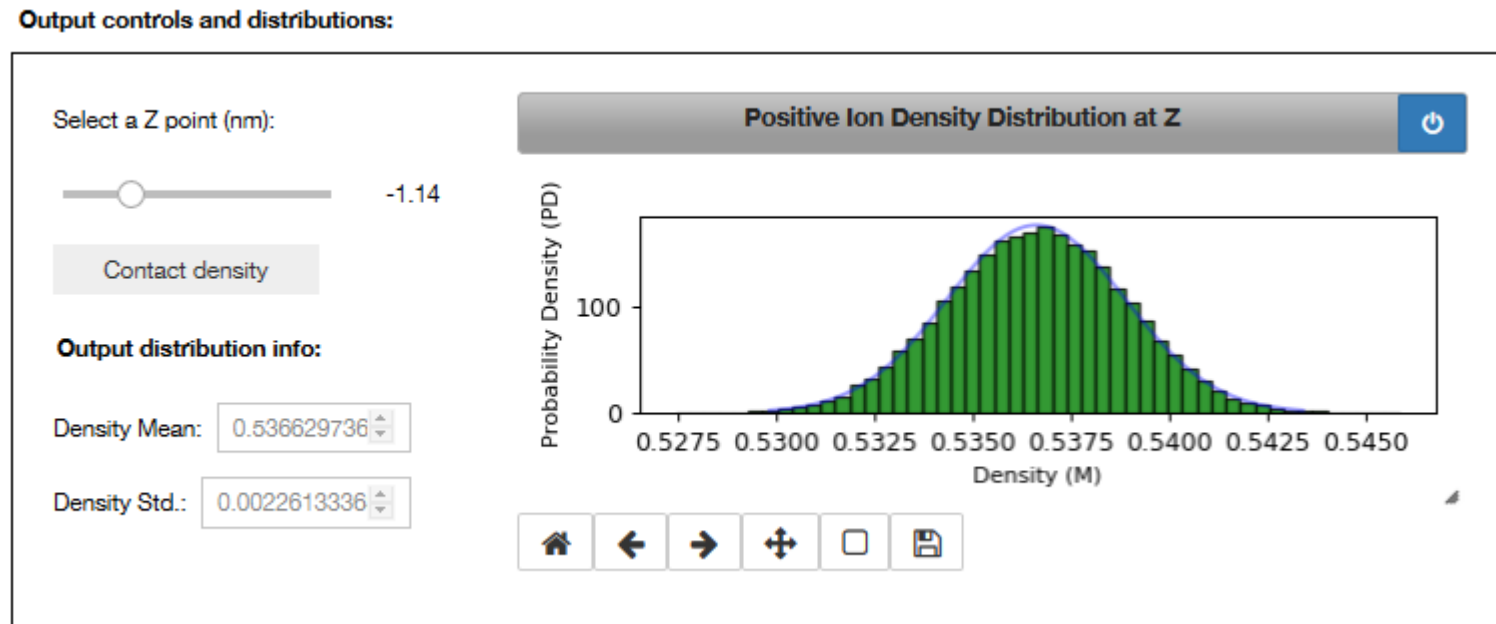
 Distribution: gaussian  Mean: 0.714  Std.: 0.001

Samples per input:

 Simulate

# UQ Outputs-1

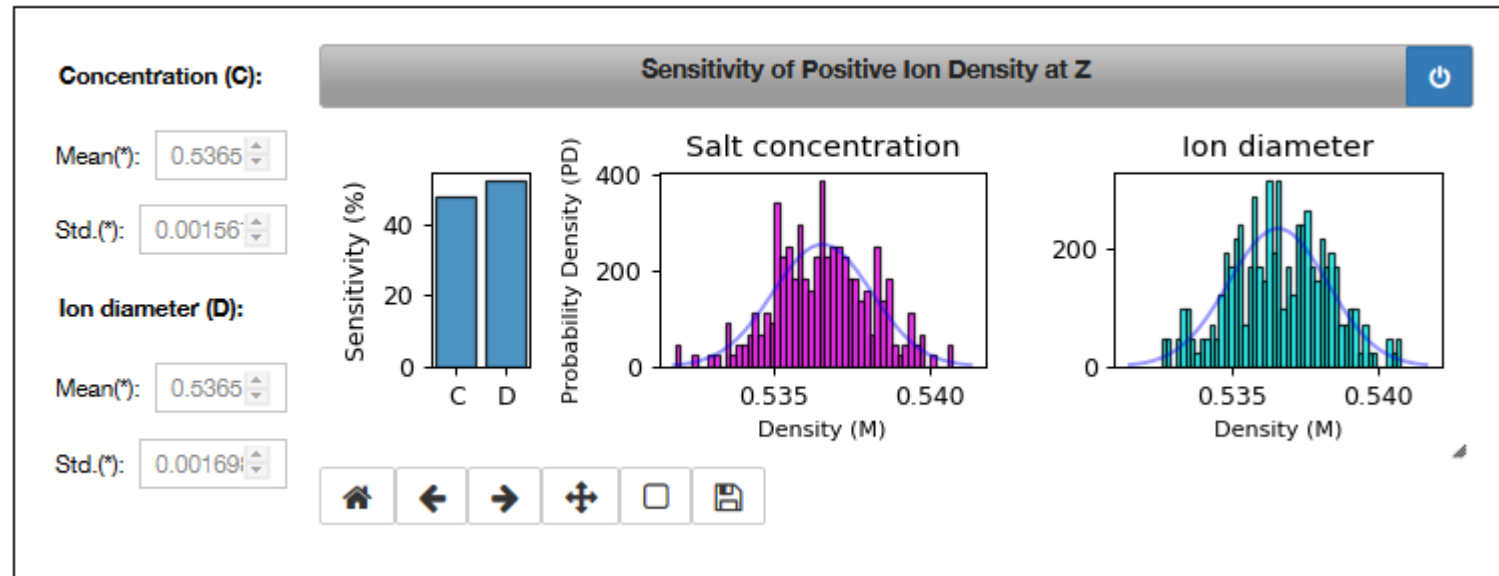
- This output allows users to see the output probability density at one point in the 2D density plot.
- Users can change the “Z point slider” to see density probability distribution at different points



# UQ Outputs-2: Sensitivity analysis

- When there is more than one UQ input, sensitivity analysis will be performed. The values indicate the relative amount the output changes over the range of each input parameter's distribution. So, for example, changing the deviation of a gaussian input parameter will impact the sensitivity.
- Users can change the “Z point slider” to see sensitivity analysis at different points

Sensitivity analysis:

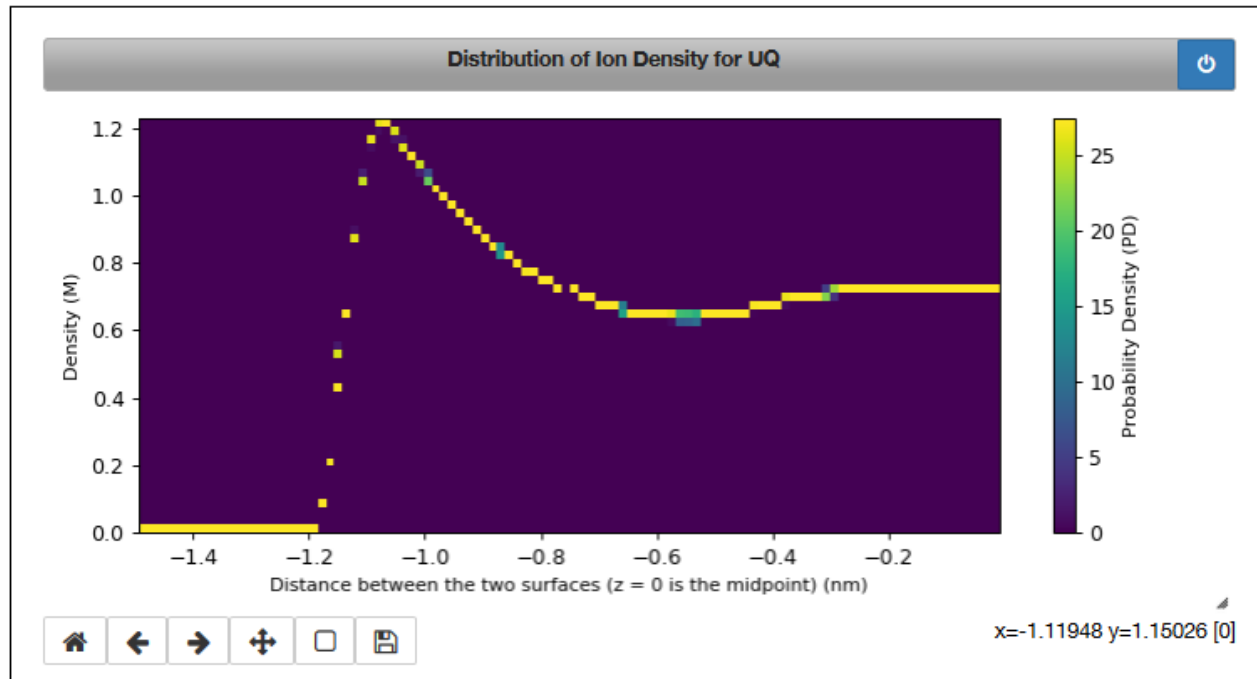


# UQ Outputs-3: Response

- All the output responses are plotted as a 2D probability density. Color represents the probability density.

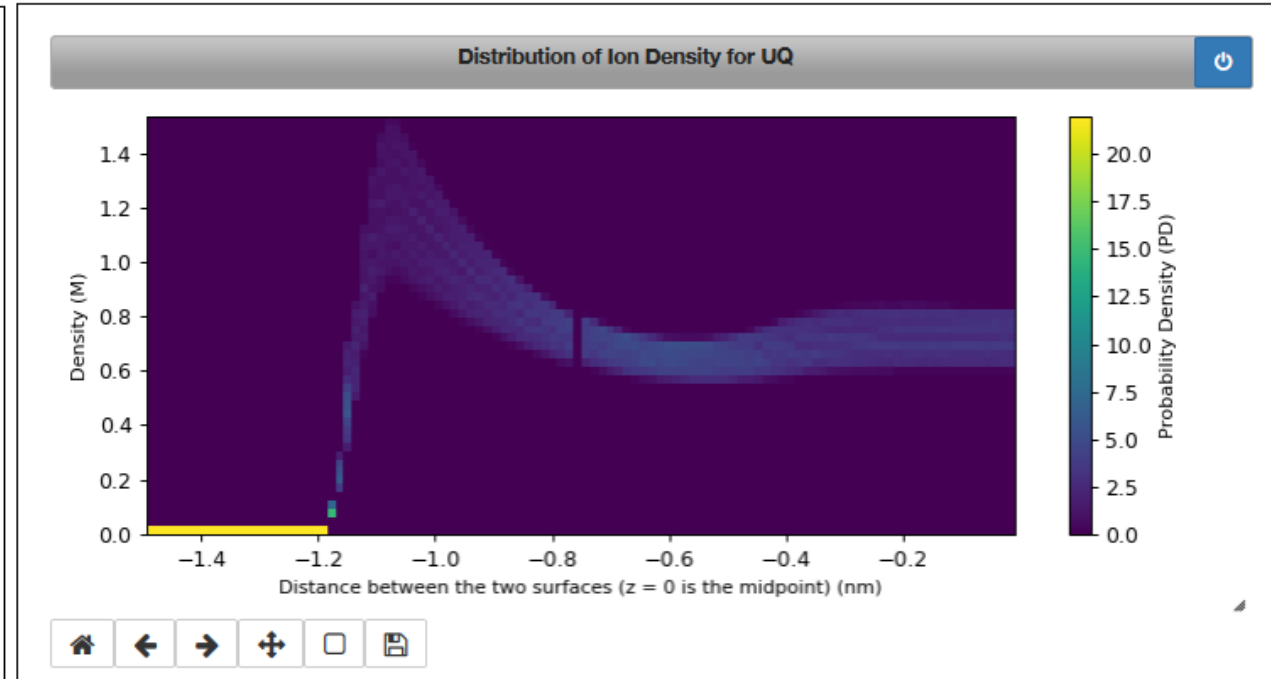
**Salt concentration, and Ion diameter both are gaussians.**

All simulations in a 2D histogram plot :



**Salt concentration, and Ion diameter are uniform and gaussians, respectively.**

All simulations in a 2D histogram plot :





# Conclusion

- Using MD simulations of ions in confinement as an example, we illustrated the idea of using ML-based methods to enhance the performance and usability of scientific simulations.
- ML model predicted critical density features with  $\sim 94\%$  success rate
- The results demonstrated that the performance gains of parallel computing can be further enhanced by using machine learning.
- ML enhancement can extend the usability of simulations for both research and education
- Explore extensions to other fields and core issues such as determination of error bars

# Acknowledgements

## ○ Jadhao Group:

- Vikram Jadhao
- Nicholas Brunk
- Lauren Nilsson
- Nasim Anousheh

## ○ Computing Resources:

- Big Red II
- NCN-hub

○ This work is supported by the National Science Foundation through Award #1720025.



Nanoconfinement: <https://nanohub.org/tools/nanoconfinement>

The app enables users to simulate ions confined between nanoparticle (NP) surfaces in aqueous media. Nanoparticles can be synthetic (such as gold NPs) or natural (e.g. proteins) and the length of confinement is of the order of nanometers. Example systems include ion channel proteins of the cell membrane, adsorbed ions near surfaces of porous electrodes, and ions confined by NPs and/or colloidal particles. NP surfaces are assumed to be unpolarizable and are modeled as planar interfaces considering the large size difference between the ions and the NPs. The app facilitates investigations for a wide array of ionic and environmental parameters. Users can extract the ionic structure (density profile) and study its dependence on salt concentration (c), ion valency (z), and other physical attributes. Users can explore interesting effects by changing the c parameter from 0.3 to 0.9 M. This increase in density leads to crowding of the channel (confinement) with a large number of ions. The effect of symmetry breaking caused by the surfaces is seen: to avoid being pushed by ions from both the sides, an ion prefers the interface over the central region (bulk). The app enables users to explore this effect of ion accumulation near the interface, and make a quantitative assessment of ionic structure in strong confinement. Another rich avenue to explore is to tune the valency of positive ions (parameter z) from 1 to 3. A positively-charged multivalent ion (+3 Fe or +2 Ca) near an interface is pulled away from the interface by oppositely charged ions with a stronger force relative to the bulk where the symmetry allows for no preferred movement. Thus, stronger electrostatic interactions (as in the case of multivalent ions) tend to cause depletion of the ions from the

**Physical Parameters**

Salt concentration (M): 0.5

Positive ion valency (e): 1 (monovalent)

Negative ion valency (e): -1

Confinement length (nm): 3

Positive ion diameter (nm): 0.714

Negative ion diameter (nm): 0.714

Predict using ML

**Computing Parameters**

Simulation steps: 5000000

Cluster mode

Progress:

**Output Controllers**

Images:  5000000

Slide to navigate the simulation snapshots

**Positive Ion Density**

Negative Ion Density Prediction Graph Simulation Snapshot Downloads

Distribution of positive ions

Distribution of positive ions confined within the nanoparticle surfaces

Density (M)

Distance between the two surfaces (z = 0 is the midpoint) (nm)

ML prediction MD simulation

Output

LAMMPS Output

Last Run: OK. Run Time: 00:26:52

Run



Thank you!

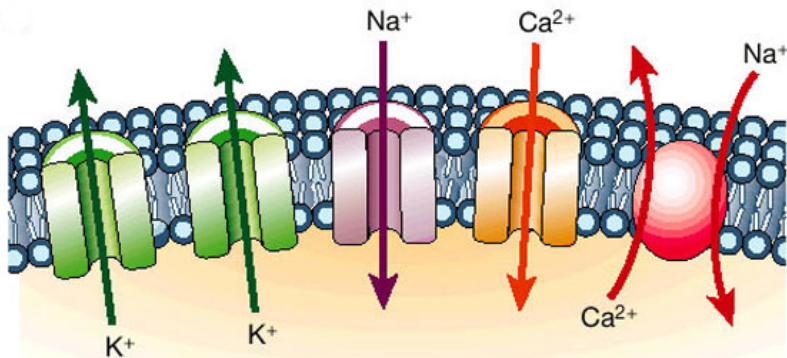
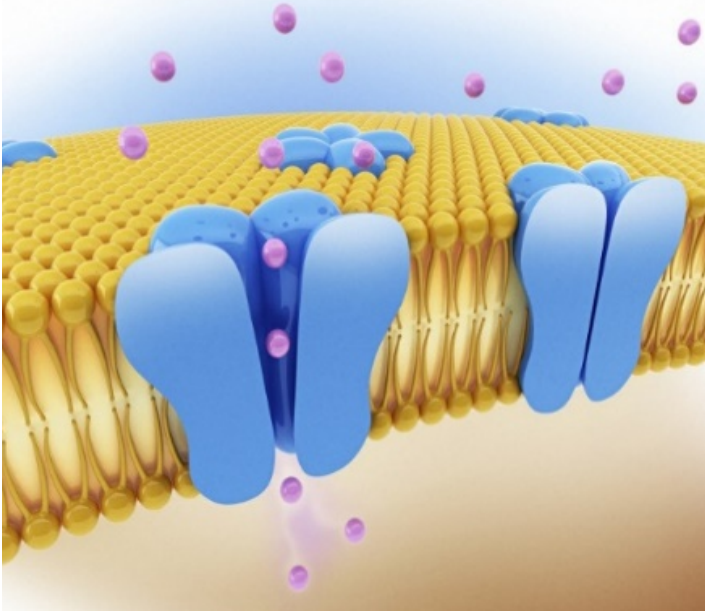
# References

- [1] Allen, R., Hansen, J. P., & Melchionna, S. (2001). Electrostatic potential inside ionic solutions confined by dielectrics: a variational approach. *Physical Chemistry Chemical Physics*, 3(19), 4177-4186.
- [2] Boda, D., Gillespie, D., Nonner, W., Henderson, D., & Eisenberg, B. (2004). Computing induced charges in inhomogeneous dielectric media: application in a Monte Carlo simulation of complex ionic systems. *Physical Review E*, 69(4), 046702.
- [3] Botu, V., & Ramprasad, R. (2015). Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry*, 115(16), 1074-1083.
- [4] Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., & Walsh, A. (2018). Machine learning for molecular and materials science. *Nature*, 559(7715), 547.
- [5] Ferguson, A. L. (2017). Machine learning and data science in soft materials engineering. *Journal of Physics: Condensed Matter*, 30(4), 043002.
- [6] Häse, F., Kreisbeck, C., & Aspuru-Guzik, A. (2017). Machine learning for quantum dynamics: deep learning of excitation energy transfer properties. *Chemical science*, 8(12), 8419-8426.
- [7] Jadhao, V., Solis, F. J., & De La Cruz, M. O. (2012). Simulation of charged systems in heterogeneous dielectric media via a true energy functional. *Physical review letters*, 109(22), 223905.
- [8] Liu, J., Qi, Y., Meng, Z. Y., & Fu, L. (2017). Self-learning monte carlo method. *Physical Review B*, 95(4), 041101.
- [9] Luo, G., Malkova, S., Yoon, J., Schultz, D. G., Lin, B., Meron, M., ... & Schlossman, M. L. (2006). Ion distributions near a liquid-liquid interface. *Science*, 311(5758), 216-218.
- [10] Spellings, M., & Glotzer, S. C. (2018). Machine learning for crystal identification and discovery. *AIChE Journal*, 64(6), 2198-2206.

Archived

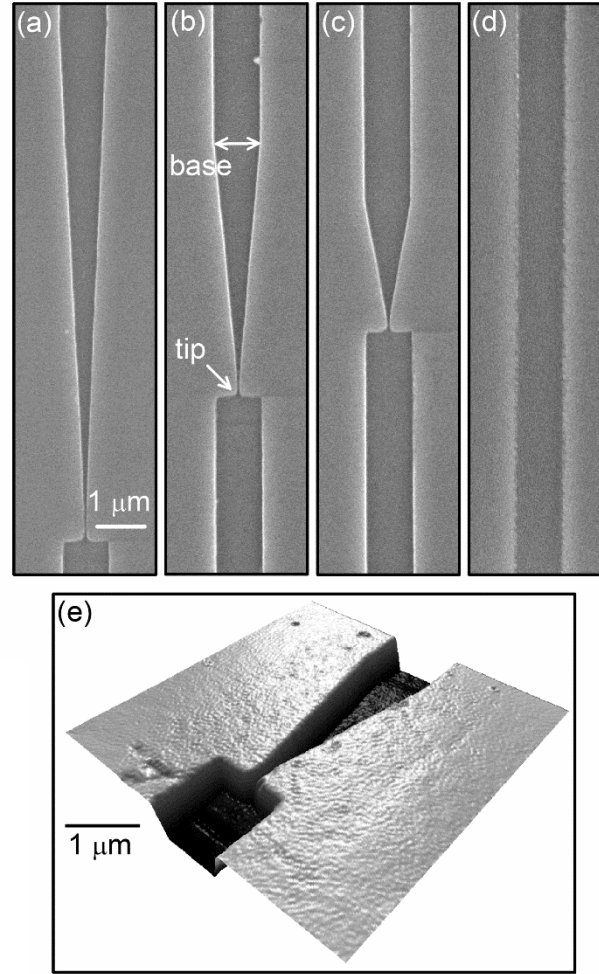
# Bio-inspired Design of Ion Separation Membranes

## Ion channels



Marbán, E. Nature 415, 213-218 (2002)

## Synthetic Ion channels

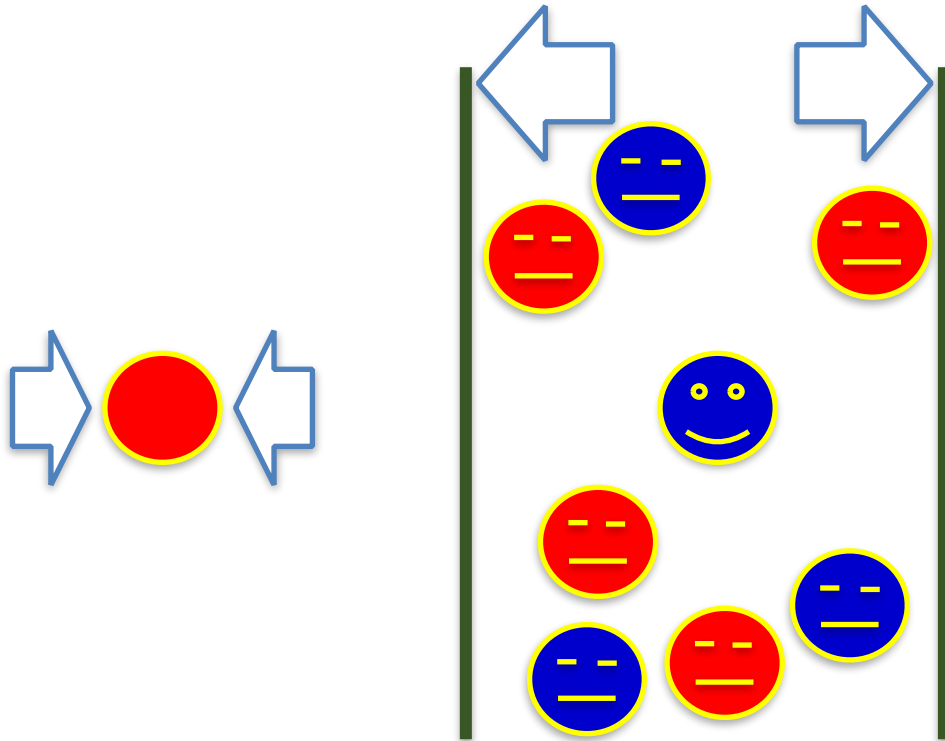


Jacobson Group at IU Chemistry

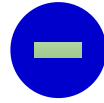
- Biological ion channels exhibit remarkable ion selection and separation efficiency, inspiring the fabrication of synthetic membranes and nanofluidic devices
- Accurate structural and dynamical information about ion behavior in confinement is critical to understand many biological and synthetic phenomena relevant to separation processes and energy storage applications
- In biological ion channels, concentrations of salt can be very high (excess of 1 M)

# Competition between Two Types of Correlations

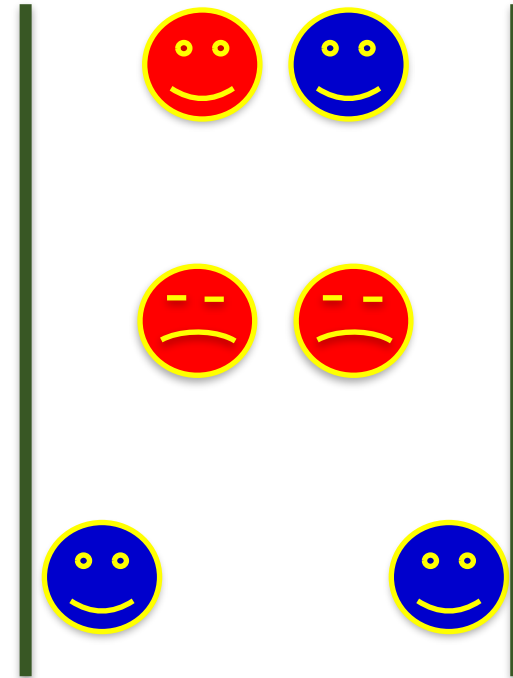
**Confinement**



Entropic (steric) forces push ions against the boundary



**Electrostatics**



Electrostatic forces lead to spatial preferences between ions

**Ionic Structure** is determined by the **competition** between **Steric** and **Electrostatic** correlations

# Why Machine Learning for Soft Materials Engineering?

- As the utility of simulations in the design of soft materials is further demonstrated, it will be necessary for **simulations to be performed at a faster speed and for a large set of parameters**.
- However, current simulations of soft materials and their data analysis incur **high computational costs despite the use of optimal parallel computing techniques**. We need new approaches to efficiently explore the high-dimensional material design space.
- On the other hand, advances in hardware have led to the **generation of “big” scientific computation data**. These datasets contain key information to advance the design of complex materials, but they are **high-dimensional** and difficult to analyze using standard approaches.
- Machine learning (ML) has the potential to directly address these needs. We employ ML to
  - leverage past simulations to **rapidly generate accurate predictions**
  - **accelerate molecular dynamics (MD)** simulations of soft matter
  - expedite simulation data analysis to **diagnose structure-property relationships** in materials