Supporting information:

The Role of Clay Charge in The Mobility of Compensating Cations: An Approach from Molecular Dynamics

Información suplementaria

El Papel de la Carga de la Arcilla en la Movilidad de los Cationes de Compensación: Una Aproximación desde la Dinámica Molecular

C. D. Marrero-Pérez a, G. Rojas-Lorenzo[b], A. Lam[a†]

a) Zeolite Engineering Laboratory, Institute of Material Science and Technology (IMRE), University of Havana, Havana, CP. 10400. Cuba. krlodavid419@gmail.com, anabel@imre.uh.cu†

b) Instituto Superior de Tecnologías y Ciencias Aplicadas (InSTEC). Universidad de La Habana, La Habana - 10400, Cuba. german@instec.cu

† corresponding author

Supporting information

S1. Python Code for Generating Non-Homogeneous LiFh Models

The Python code provided in this section generates non-homogeneous LiFh models with a layer charge of −1.2e| starting from a homogeneous LiFh model with a charge of −2e.

```
import pandas as pd
import random

#  Code to generate Lithium Fluorhectorite clay models with a charge of -1.2
# from models with a charge of -2. The first part of the code generates 15 random
# numbers out of 75 (total number of cells). These 15 cells will be the ones with a -2
# charge (defects). In the second part of the code, the remaining 60 cells are processed,
# and for each one, a random decision is made on which Lithium atom to substitute.

# Read the original file with Pandas
df = pd.read_csv('CONFIG', header=None)

# Generate 15 unique random numbers between 1 and 75
random_numbers = sorted(random.sample(range(1, 76), 15))
print(random_numbers)

# Create a list to store the lines to copy
lines_to_copy = []

# For loop to iterate through numbers from 1 to 75
for num in range(1, 76):
    if num in random_numbers:  # Check if the number is in the random numbers list
        start_line = num * 152 - 152 + 5  # Calculate the lower limit
        end_line = num * 152 + 5   # Calculate the upper limit
        lines_to_copy.extend(range(start_line, end_line))  # Add the lines to copy to the list

# Select the lines to copy and remove them from the original file
lines_selected = df.iloc[lines_to_copy]
df.drop(lines_to_copy, inplace=True)

# Write the selected lines to a new file
lines_selected.to_csv('celdas con defectos.txt', header=False, index=False)

# Save the original file without the selected lines
df.to_csv('celdas sin defectos.txt', header=False, index=False)

print("The selected lines have been copied to the new file and removed from the original file.")

# Read the txt file
df = pd.read_csv('celdas sin defectos.txt', header=None)

a = 0

for i in range(69,9040 , 150):
    # Generate random number between 0 and 1
```

```
    aleatorio = random.randint(0, 1)
    print(aleatorio)


    if aleatorio == 0:
        # Change 'Li' to 'Mg' in line i
        df.iloc[i] = df.iloc[i].str.replace('Li', 'Mg')

        # Delete lines i+14 and i+15
        df = df.drop([i+14+2*a , i+15+2*a])

    else:

        # Change 'Li' to 'Mg' in line i+2
        df.iloc[i+2] = df.iloc[i+2].str.replace('Li', 'Mg')

        # Delete lines i+12 and i+13
        df = df.drop([i+12 + 2*a, i+13 + 2*a])
        # Swap lines i and i+1 with lines i+2 and i+3
        temp = df.iloc[i].copy()
        df.iloc[i] = df.iloc[i+2]
        df.iloc[i+2] = temp

        temp = df.iloc[i+1].copy()
        df.iloc[i+1] = df.iloc[i+3]
        df.iloc[i+3] = temp
    a=a+1

# Save the modified DataFrame to a new txt file
df.to_csv('celdas sin defectos (balanceadas).txt', header=False, index=False)

# Read the text files
df1 = pd.read_csv('celdas sin defectos (balanceadas).txt', header=None)
df2 = pd.read_csv('celdas con defectos.txt', header=None)

# Concatenate the dataframes
df_final = pd.concat([df1, df2],ignore_index=True, axis=0)

# Save the result in a new text file with multiple space separator
df_final.to_csv('Arcilla con carga -1.2.txt', index=False)
```

S2 Force fields

In the CLAYFF force field, the total energy of the system is determined by evaluating the appropriate energy terms for every atom-atom interaction [1]. This force field includes contributions from Coulombic (electrostatic) interactions, short-range interactions (often referred to as the van der Waals term), and bonded interactions for the O-H terminal.

$$E_{total} = E_{Coul} + E_{VDW} + E_{bondstretch} \quad (1)$$

The O-H (Oh-H) terminal bonding interactions are only present in the 010 model, where periodicity is disrupted and the Si and O valences are completed by OH and H groups. The Oh-H bond was modeled using Morse potential terms for edge surfaces, as proposed by Pouvreau et al [3]. However, as explained in Reference [5], the angle term and water potential proposed by Pouvreau et al. could not be fitted to our model. Water molecules were modeled using the SPC force field [4] as implemented in CLAYFF.

Lennard-Jones (L-J) potential was used to represent the van der Waals term in the total energy of the system.

$$E_{LJ} = \sum_{i \neq j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right]$$

This potential combines the short-range repulsion associated with atom-atom overlap and the short-range attraction associated with electron dispersion and it is calculated in CLAYFF force field by the following expresion:

$$E_{VDW} = \sum_{i \neq j} \varepsilon_{ij}^{*} \left[\left(\frac{\sigma_{ij}^{*}}{r_{ij}} \right)^{12} - 2\left(\frac{\sigma_{ij}^{*}}{r_{ij}} \right)^{6} \right]$$

In CLAYFF, ε_{ij}^{*} is determined using standard geometric combination rule and u σ_{ij}^{*} sing standard arithmetic combination rule:

$$\varepsilon_{ij}^{*} = \left(\varepsilon_i^{*} \varepsilon_j^{*} \right)^{1/2} \text{ and } \sigma_{ij}^{*} = \frac{\sigma_i^{*} + \sigma_j^{*}}{2}$$

Table SI1 shows the charges and the atomic parameters used for the force field used in our simulations.

Table SI1. Charges and atomic parameters for van der Waals interactions used in the simulations.

Clay	symbol	charge (*e*)	ε_i (kcal/mol)	σ_i (Å)
	Si[1]	2.1000	1.8405×10^{-6}	3.3020
	Mg	1.415	9.0298×10^{-7}	5.2643
	Li	0.415	9.0298×10^{-7}	4.2101
	Li+[2]	1.0000	0.16029	2.3370
	F	-0.8109	0.1802	3.1170
	O[1]	-1.0500	0.1554	3.1655
	Oo[1]	-1.1808		
	Oh[3]	-0.95		

	H[3]	0.425		
water SPC	HW[4]	0.4100		
	OW[4]	-0.8200	0.1553	3.5533
Oh-H Morse Potential[3] $E_{bondstretch} = E_0[\{1 - \exp(-k\,(r_{ij} - r_0))\}^2 - 1]$				
	E_0 (kcal/mol)	k (Å^{-1})	r_0 (Å)	
	132.2491	2.1350	0.9450	

[1] R. T. Cygan, J.-J. Liang, and A. G. Kalinichev, J. Phys. Chem. B **108**, 1255 (2004).
[2] S. Koneshan, J. C. Rasaiah, R. M. Lynden-Bell, and S. H. Lee, J. Phys. Chem. B **102**, 4193 (1998).
[3] M. Pouvreau, J. A. Greathouse, R. T. Cygan, and A. G. Kalinichev, J. Phys. Chem. C **123**, 11628 (2019).
[4] H. Berendsen, J. P. M. Postma, W. van Gunsteren, and J. Hermans, *Interaction models for water in relation to protein hydration.* In *Intermolecular Forces*, (Pullman, B., Ed.; D. Reidel: Amsterdam, 1981) pp. 331.
[5] A. Lam and G. Rojas-Lorenzo Rev. Cubana Fis. **41**, 10 (2024).

S3. Thermal Energy Contribution in the Clay-Water System

The thermal energy of the system (comprising 2940 clay particles and 900 rigid SPC water molecules at 300 K) was calculated using the equipartition theorem, which assigns an energy of $1/2k_BT$ per degree of freedom. The particles of the clay (2940), exhibits 6 degrees of freedom (3 translational + 3 vibrational). While, the water SPC model, where each rigid molecule contributes 6 degrees of freedom (3 translational + 3 rotational).

The Thermal Energy Calculation:

For each component, the total thermal energy is:

$$E_{thermal} = \frac{f}{2} N k_B T$$

where f is the number of degrees of freedom, N is the number of particles, k_B is the Boltzmann constant and T=300K.

- For clay particles:

$$E_{clay} = \frac{6}{2} x\, 2940\, x k_B\, x 300 = 0.888\, kcal/mol$$

- For water molecules:

$$E_{water} = \frac{6}{2} x\, 900\, x k_B\, x 300 = 0.270\, kcal/mol$$

Finally, the Thermal Energy is:

$$E_{total} = E_{clay} + E_{water} = 1.158\, kcal/mol$$

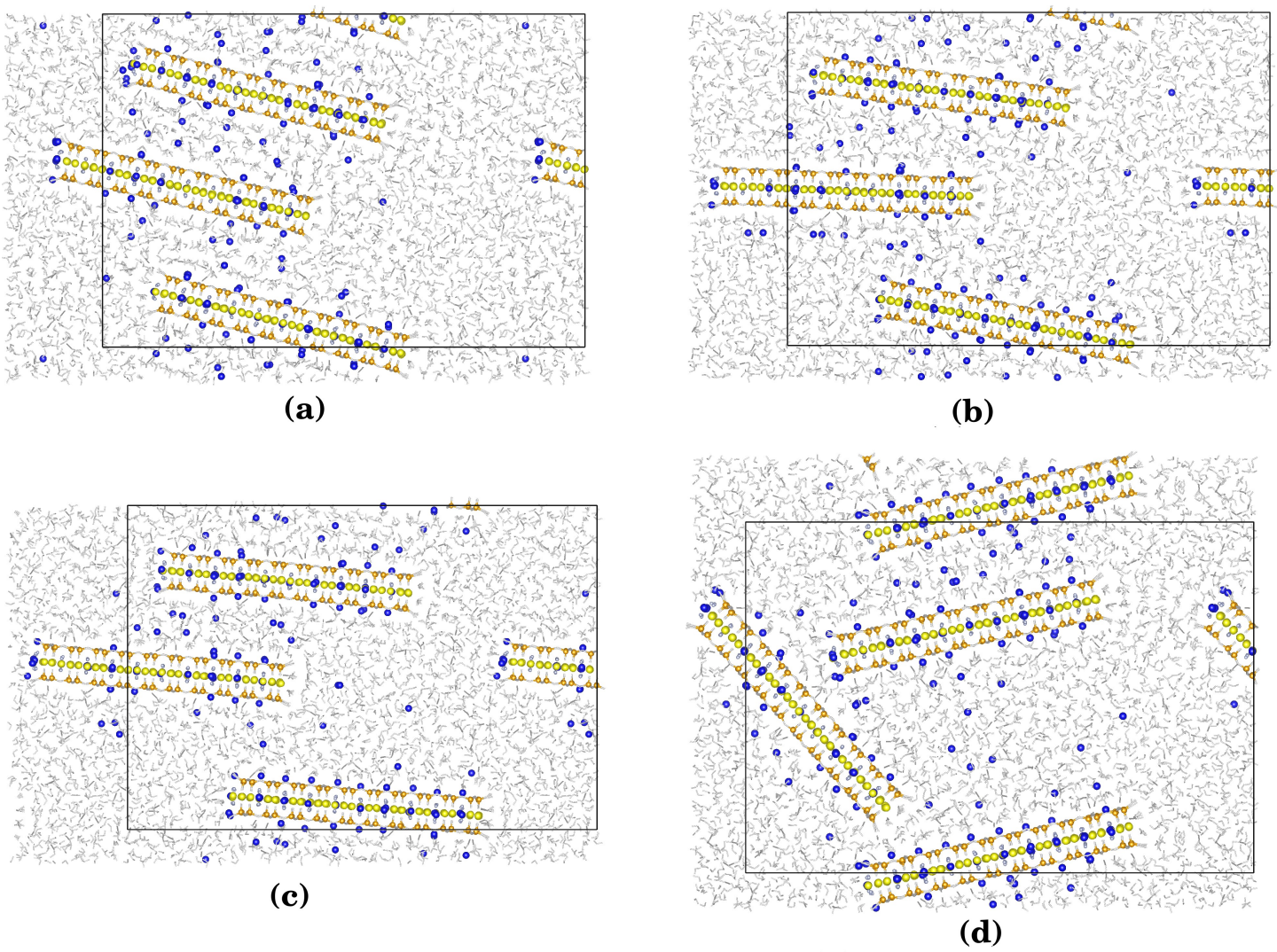

Figure SI1. Different snapshots of the simulation in the 010 model. (a) 0.65 ns, (b) 3.29 ns, (c) 4.05 ns and (d) 5.31 ns. The simulation box is indicated within the black rectangle, and boundary conditions were applied to enhance visualization. The orange, grey, blue, yellow, pink and white spheres correspond to Si, O, Li, Mg, F and H atoms respectively.

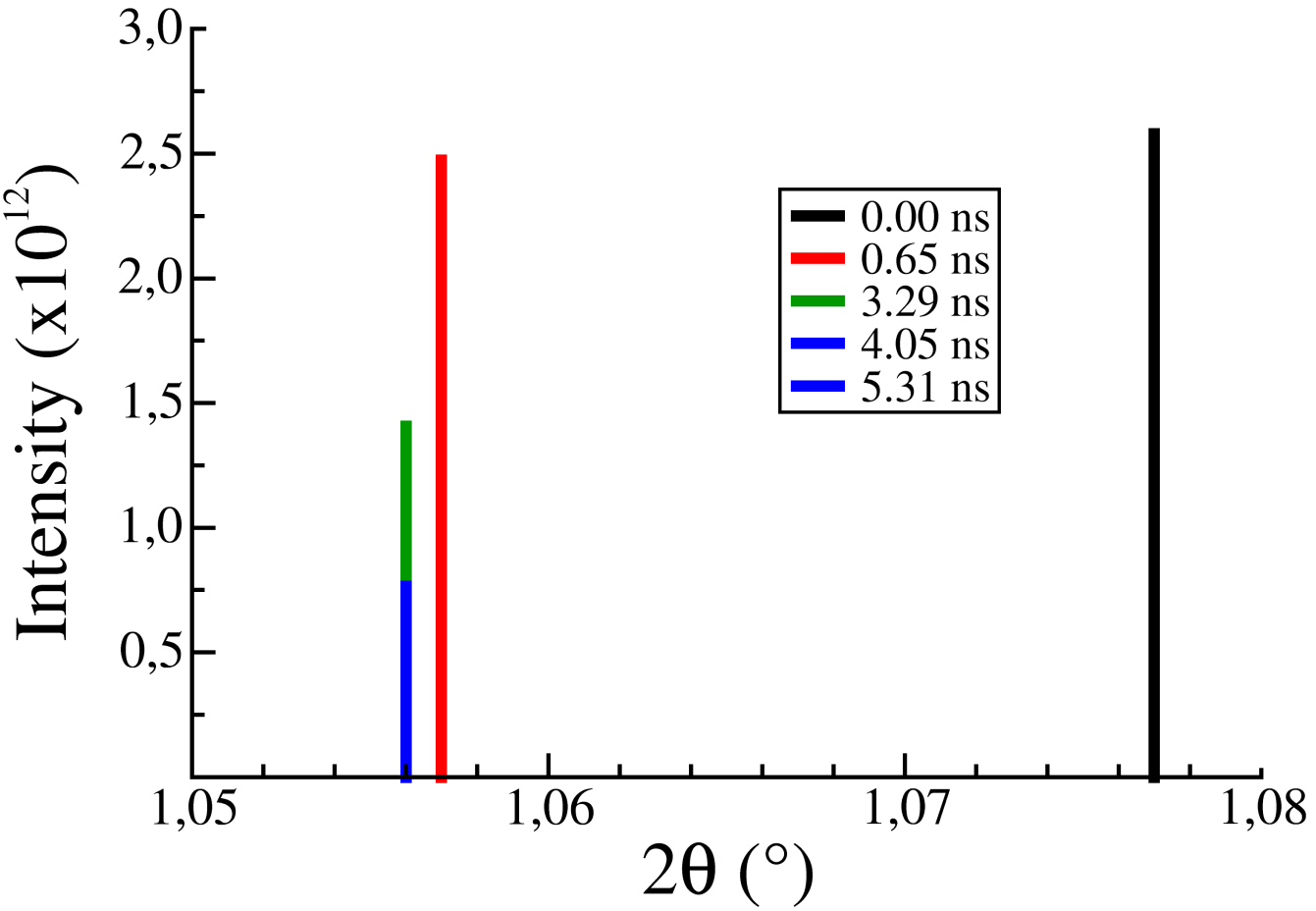

Figure SI2. 010 reflection of the X-ray diffraction pattern of the 010 model at different stage of the simulation.

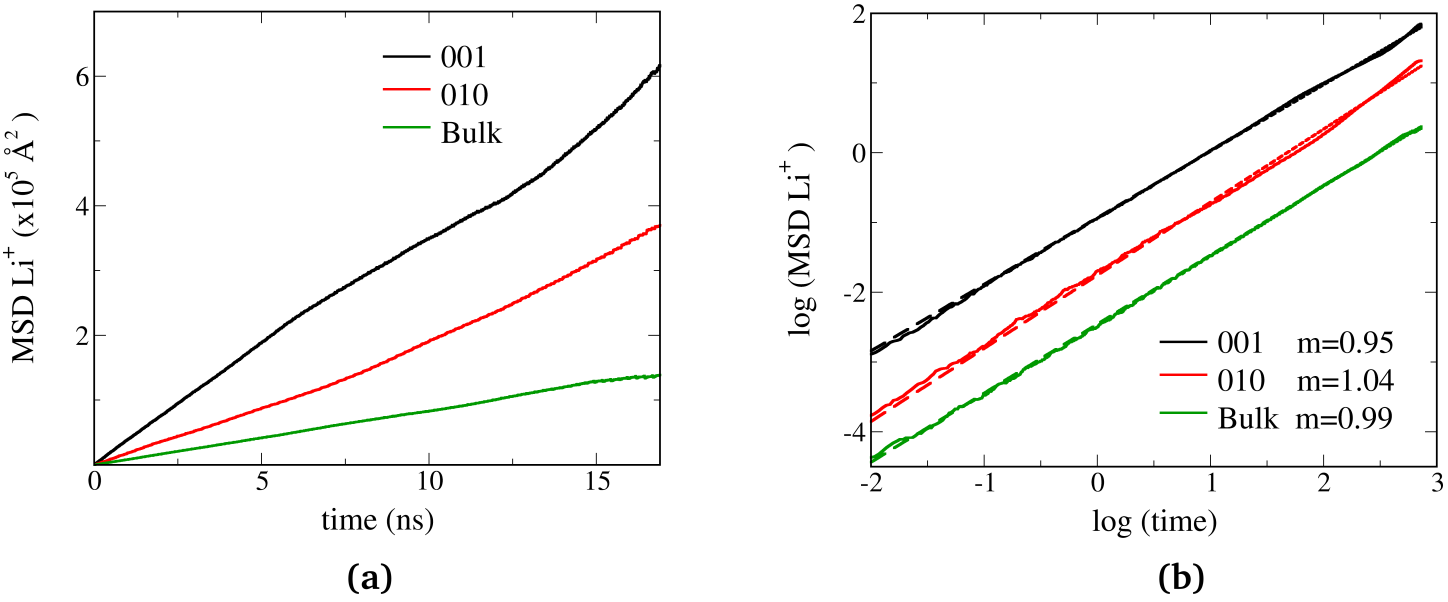

Figure SI3. Lithium cations diffusion characteristics in the bulk (green), 010 (red) and 001 (black) models. **(a)** Ensemble-averaged mean square displacement (MSD) of *all* Li+ cations as a function of time, calculated from DL_POLY trajectories, **(b)** Logarithmic scaling (log(MSD) vs log(time)) of the data in (a), with dashed lines indicating slope (m). If m= 1 (Fickian difussion) and if m= 0.5 (single-file diffusion).The observed ensemble-averaged slopes (m ≈ 1) confirm normal diffusion at macroscopic scales, despite nanoconfinement. This behavior is consistent with the jump-diffusion mechanism operating at atomic scales (see Figure SI4).

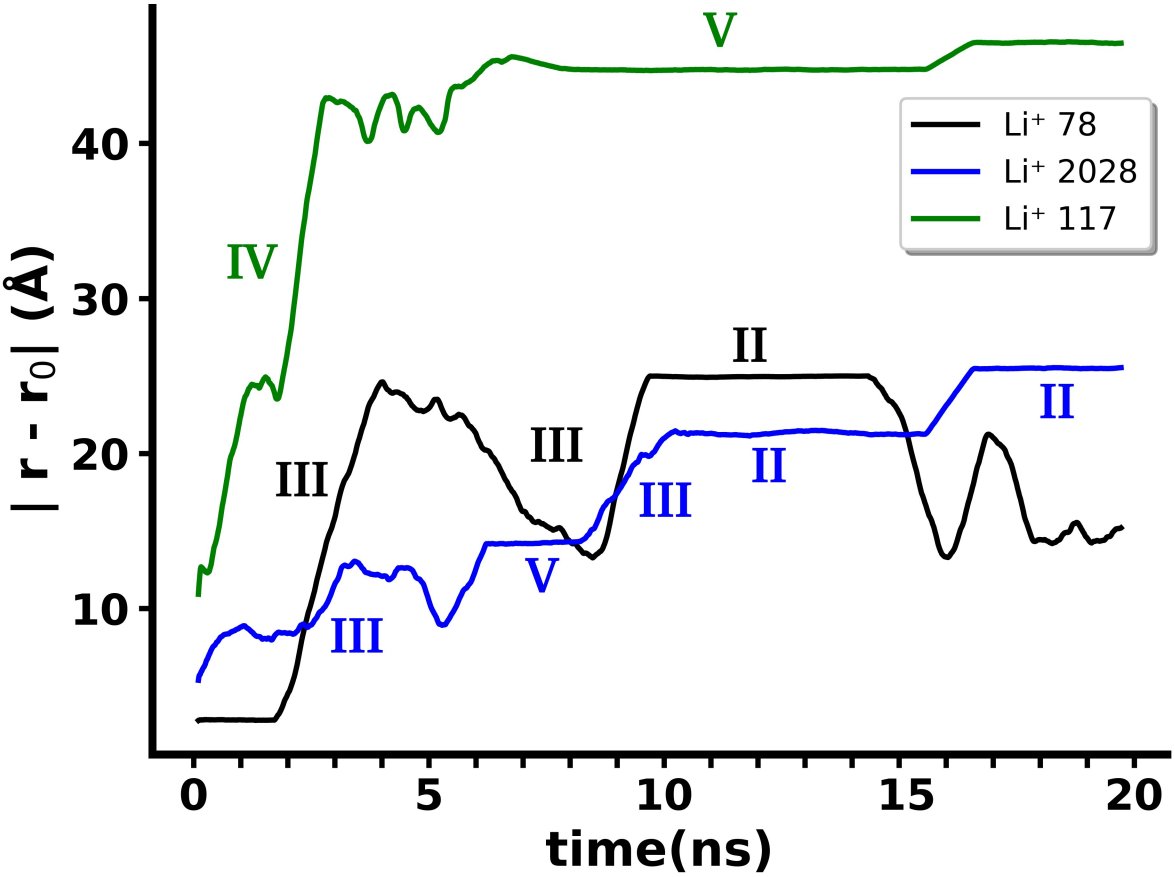

Figure SI4. Displacements during the simulation of representative Li^+ cations: 78 (black line), 2028 (blue line), and 117 (green line) in the 001 model. The movements of each Li^+ are shown and denoted by Roman numerals: II) move to another hexagonal cage within the same layer, III) move into the interlayer space, IV) diffuse into the outer water reservoir, V) cross the interlayer and occupy the opposite layer.

SI4. Videos of the simulations.

The videos of the simulations of the different models can be seen in:

Bulk: https://www.youtube.com/watch?v=ZU8JD5UaI5c&t=21s

Model 010: https://www.youtube.com/watch?v=d-gr0ayxDnQ

Model 001: https://www.youtube.com/watch?v=rwRm51y27wo