

Download PDF

Article | [Open access](#) | Published: 12 July 2021

Efficient water desalination with graphene nanopores obtained using artificial intelligence

[Yuyang Wang](#), [Zhonglin Cao](#) & [Amir Barati Farimani](#) 

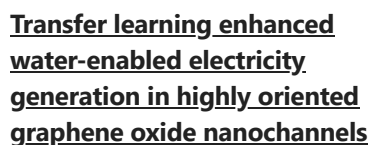
[npj 2D Materials and Applications](#) **5**, Article number: 66 (2021)

11k Accesses | **55** Citations | **45** Altmetric | [Metrics](#)

Abstract

Two-dimensional nanomaterials, such as graphene, have been extensively studied because of their outstanding physical properties. Structure and topology of nanopores on such materials can be important for their performances in real-world engineering applications, like water desalination. However, discovering the most efficient nanopores often involves a very large number of experiments or simulations that are expensive and time-consuming. In this work, we propose a data-driven artificial intelligence (AI) framework for discovering the most efficient graphene nanopore for water desalination. Via a combination of deep reinforcement learning (DRL) and convolutional neural network (CNN), we are able to rapidly create and screen thousands of graphene nanopores and select the most energy-efficient ones. Molecular dynamics (MD) simulations on promising AI-created graphene nanopores show that they have higher water flux while maintaining rival ion rejection rate compared to the normal circular nanopores. Irregular shape with rough edges geometry of AI-created pores is found to be the key factor for their high water desalination performance. Ultimately, this study shows that AI can be a powerful tool for nanomaterial design and screening.

Similar content being viewed by others



Graphene oxide-based membranes for water desalination and purification

3D hierarchical aquaporin-like nanoporous graphene membrane with engineered tripartite nanochannels for...

Article | Open access
15 February 2024

Single-layer graphene, as an iconic two-dimensional (2D) material, has drawn much scientific attention in recent decades. Because of its ultrathin thickness and outstanding mechanical properties, graphene with artificial pores has been demonstrated to have great potentials in many engineering applications, such as effective hydrogen gas separator^{1,2,3}, next-generation energy storage or supercapacitor building^{4,5}, and high-resolution DNA sequencing^{6,7,8}. Given the potential imminent global water scarcity crisis, another important application for nanoporous graphene is energy-efficient water desalination^{9,10}. Equipped with nanoporous 2D material membranes like graphene, the reverse osmosis (RO) water desalination process can expect 2–3 orders improvement in water flux compared with traditional polymeric membranes^{9,10,11,12,13,14}. In RO, the geometry of nanopores in 2D materials plays a determinant role in water desalination performance^{9,11}. In general, a large pore that allows high water flux is likely to perform poorly in rejecting ions; a small pore that rejects 100% undesired ions, on the other hand, usually have limited water flux. Thus, an optimal nanopore for water desalination is expected to allow as high water flux as possible while maintaining a high ion rejection rate. However, finding the optimal nanopore geometry on graphene can be challenging due to high computational and experimental cost associated with extensive experiments, i.e., there are countless possible shapes for a pore on a 4 nm × 4 nm graphene membrane, but evaluating the water flux and ion rejection of a single pore using 10 ns MD simulation takes roughly 36 h on a 56-core CPU cluster. Given this time benchmark, evaluating the water desalination performance of 1000 graphene nanopores can take more than 4 years. Therefore, to discover the optimal graphene nanopore for water desalination, an efficient nanopore screening method with a fast nanopore water desalination performance predictor (performance predictor in short) is needed. Inspired by the recent success of deep learning¹⁵ and reinforcement learning (RL)¹⁶, we create an AI framework consists of the combination of the state-of-art deep reinforcement learning (DRL) algorithm with a convolutional neural network (CNN) to solve this challenge.

The main idea of RL¹⁷ is to train an agent to find an optimal policy that maximizes the expected return in the future through actively interacting with the environment to achieve a goal. Recently, DRL^{16,18}, which models the RL agent with artificial neural networks, has proven to be an efficient tool in material-related engineering fields, such as material design^{19,20,21} and molecule optimization²². In this work, we designed and implemented an artificial intelligence framework consisting of DRL, which is capable of creating a nanopore on a single-layer graphene membrane to reach optimal water desalination performance. By a series of decisions on whether or not to remove carbon atoms and which atom to be removed, the DRL agent can eventually create a pore that allows the highest water flux while maintaining ion rejection rate above an acceptable threshold. Such precisely controlled atom-by-atom removal nanopore synthesis can be conducted by electrochemical reaction^{23,24}. Perforation technologies can also offer the opportunities to control the formation of pores, gaps, and bridges with nano-meter dimensions on 2D materials such as graphene experimentally^{25,26,27,28}. During training, the DRL agent learns from the feedback based on the water desalination performance (e.g. reward for high water flux and penalty for lower ion rejection). However, conventional methods to calculate desalination performance, like MD simulation, are too time-consuming to be implemented in our DRL model. To evaluate DRL-designed nanopores fast and accurately, we implemented a CNN-based^{29,30,31,32} model that uses the geometry of porous graphene membrane to directly predict the water flux and ion rejection rate under certain external pressure. To this end, a ResNet³² model is trained on the dataset we collected through MD simulation of water desalination using various graphene nanopores. With the CNN-accelerated desalination performance prediction, the DRL model can rapidly discover the optimal graphene nanopore for water desalination. MD simulations on top-performing DRL-created graphene nanopores prove that they have higher water flux while maintaining a similar ion rejection rate compared to the circular nanopores. Further investigation of molecular trajectories reveals the reason that DRL-created nanopores outperform the conventional circular nanopores and provides insights for energy-efficient water desalination. Lastly, our AI-driven framework can be potentially applied to various application areas³³ of 2D materials besides water desalination, such as gas permeation and separation, battery and supercapacitor applications, and biomolecular translocation^{34,35}.

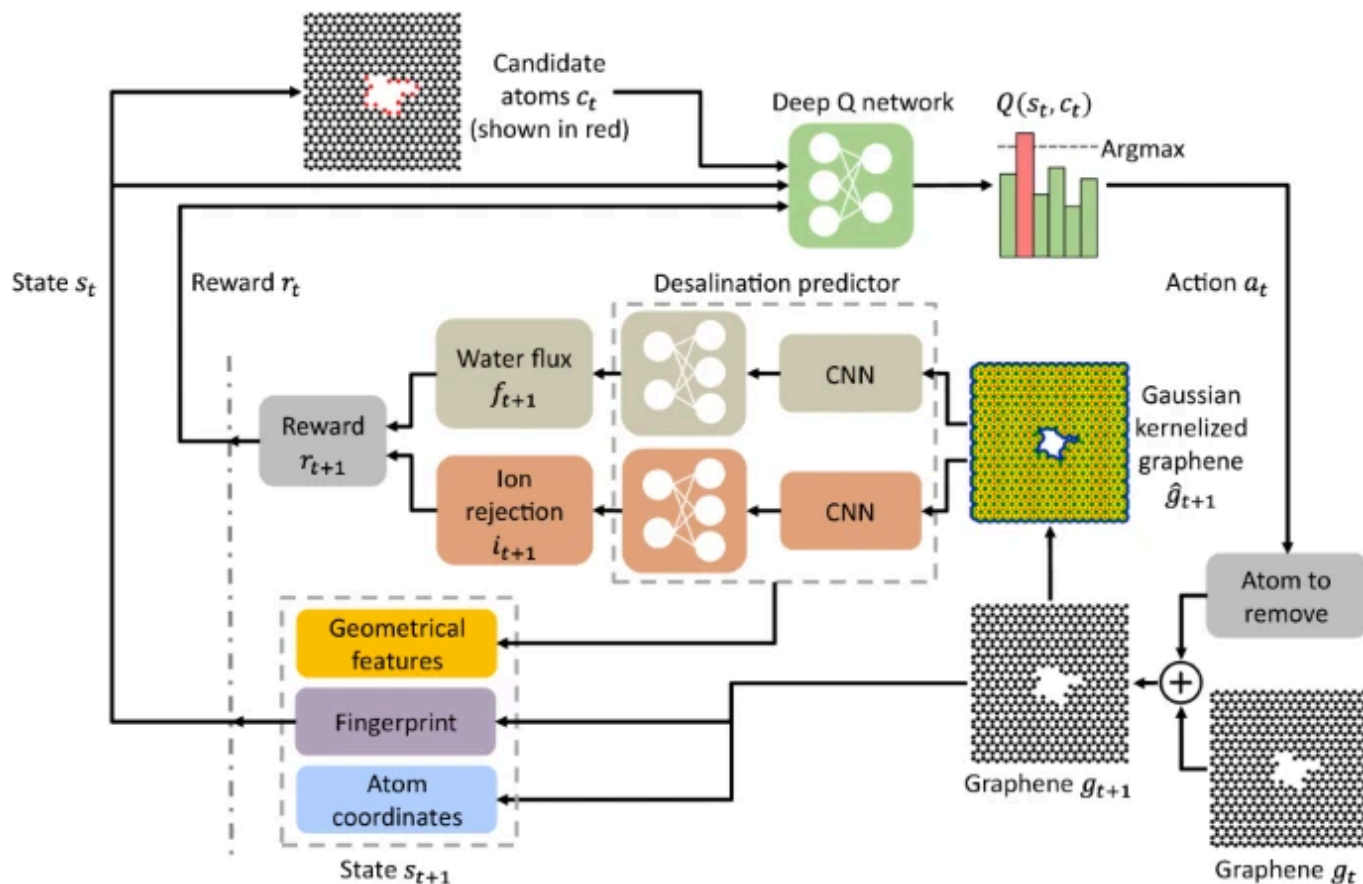
Results

AI framework

The framework (Fig. 1) of water desalination for efficient water desalination consists of a DRL agent and a CNN-based performance predictor network. At each timestep, the DRL agent generates a updated nanopore by removing at most one atom from the graphene, and the CNN-based performance predictor network predicts the water flux/ion rejection rate of the nanopore, such that the DRL agent can get instantaneous feedback on its action. Given the featurized information of the nanoporous graphene sheet (Morgan fingerprint, Cartesian coordinates of each atom, and geometrical features of graphene membrane from the CNN model) and predicted water flux and ion rejection, the DRL agent (details of DRL agent shown in Supplementary Fig. 1) was trained to create a pore on graphene sheet with the goal to maximize its

performance in the water desalination process. The dataset used to train CNN performance predictor is generated by MD simulations of various graphene nanopores for water desalination.

Fig. 1: Overview of AI creating nanopores for efficient water desalination via the integration of CNN and DRL.



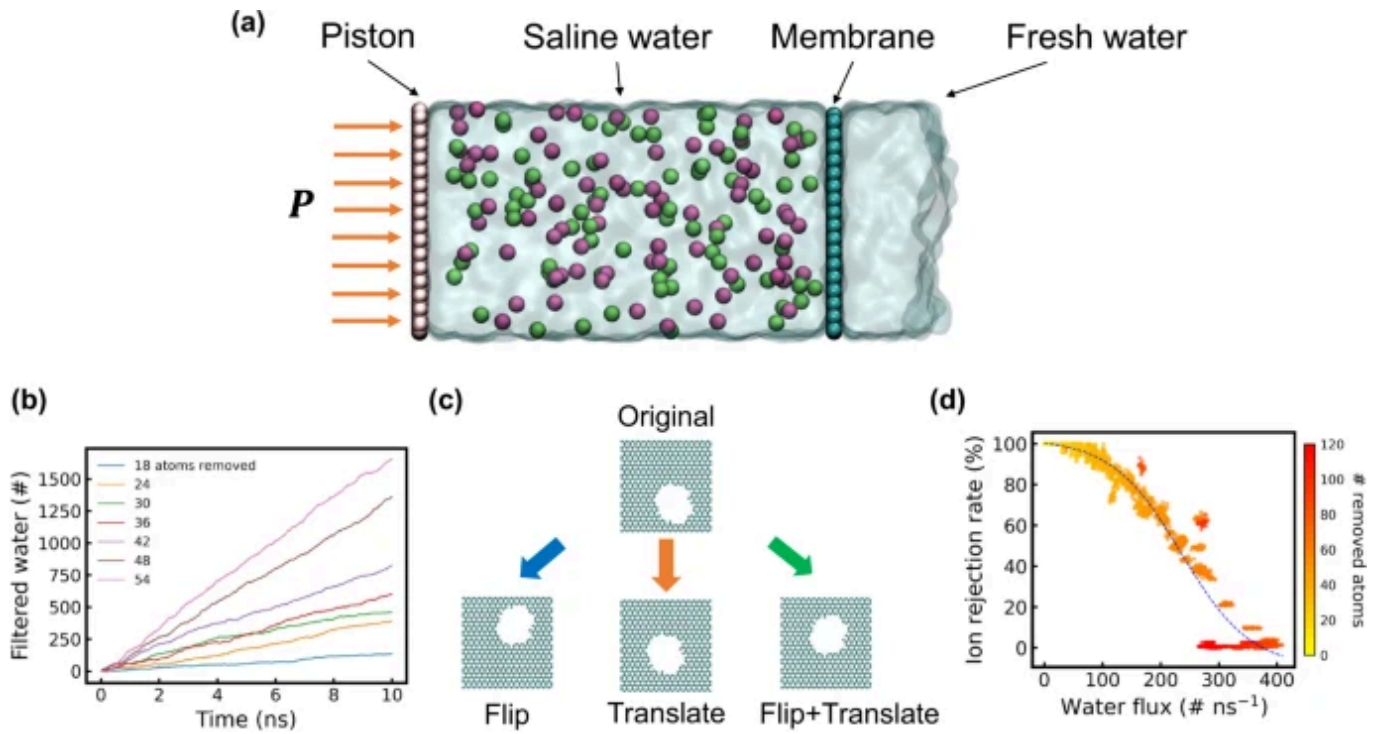
The whole framework runs by removing atoms sequentially. At each timestep t , at most one candidate atom (colored as red) is removed from the current graphene nanopore g_t to generate a updated nanopore g_{t+1} . Any dangling atoms caused by the removal of candidate atom are also removed from g_t . g_{t+1} is fed into a CNN-based performance predictor to predict water flux f_{t+1} and ion rejection rate i_{t+1} . Meanwhile, the geometrical feature is extracted from the CNN. The reward is then calculated from the predicted i_{t+1} and f_{t+1} . The geometrical feature is concatenated with the fingerprint and atom coordinates as the state s_{t+1} . Given g_{t+1} , candidate atoms to remove are picked from those located at the edge of the nanopore. The DRL agent constructed upon deep Q-network takes the reward, candidate atoms, and state as input to determine the next atom to remove from the graphene.

Graphene nanopore dataset

We consider the graphene nanopore system as illustrated in Fig. 2a, which consists of four different sections: a graphene piston that applies constant external pressure; a saline water section containing potassium chloride

as solute; a single-layer graphene membrane with the pore of different geometries; and a freshwater section which functions as a reservoir of filtered water. The molarity of the saline water in this work is ~ 2.28 M, which is higher than normal seawater for the sake of computational efficiency. The dimension of the simulation box is approximately $4\text{ nm} \times 4\text{ nm} \times 13\text{ nm}$ in x , y , and z -directions, respectively. A periodic boundary condition was applied to all three dimensions.

Fig. 2: Dataset generation using MD simulation and data processing.



a Graphene nanoporous membrane water desalination system in MD simulation. **b** Number of filtered water molecules with respect to simulation time for pores with a different number of atoms removed. The slope of the least-squared regression line of each curve is the water flux. **c** Two data augmentation techniques: flip and translation. **d** Water flux and ion rejection rate distribution of the final training dataset for predictive CNN model. The blue dashed line represents a reverse sigmoid curve fitted in terms of water flux and ion rejection rate.

The two major performance indicators of a membrane in water desalination: water flux and ion rejection rate, were calculated by post-processing the MD simulation trajectories. The slope of the fitted least-square regression line on filtered water with respect to the simulation time curve was calculated to be the water flux of each membrane (Fig. 2b). The ion rejection rate of each membrane was calculated by dividing the number of ions in the freshwater section by the total number of ions.

The total number of different simulated porous graphene is 185. Since the reward of DRL agent in our model was calculated based on the water flux/ion rejection prediction of performance predictor (Eqs. (1) and (2);

Supplementary Fig. 2), highly accurate predictions must be achieved to ensure the quality of DRL training. A much larger training dataset was necessary for the optimization of CNN model. The method employed in our study to substantially increase the size of the dataset was data augmentation^{36,37}. Given that the water desalination performance of a graphene pore depended on its size and geometry, we could assume that a flipped or translated pore on the same graphene membrane would demonstrate identical water flux/ion rejection rate of the original pore (proven by MD simulations in Supplementary Fig. 3). Therefore, copies of original pores were created by being flipped along x- or y-axis and/or translating in -4 to 4 Å in x and y directions (Fig. 2c). The water desalination performance of pore copies is a random variable of normal distribution (μ = original pore performance, σ = 1% of original pore performance). In order to improve CNN's prediction accuracy on the performance of pores created by the DRL agent, we augmented DRL-generated pores 32 times. Among the other pores, the ones with zero water flux (too small to allow water transport) were augmented 6 times, and the rest of the pores were augmented 24 times. The final dataset used for CNN training contains 3937 samples (Fig. 2d). A reverse sigmoid function was fitted to the distribution of samples to show the general relationship between the water flux and ion rejection rates.

Water desalination performance prediction

To facilitate the efficient estimation of water desalination performance in our AI-driven framework, a CNN model was trained to make an instantaneous prediction of water flux and ion rejection rates given a specific graphene nanopore. CNN is widely known as a universal feature extractor. Given that the water desalination performance of a graphene nanopore depends on its geometrical features, CNN can be the most suitable model to recognize geometrical features and make predictions based on them. The CNN models were implemented based on VGG³¹ and ResNet³², and a multi-layer perceptron (MLP) was built on top of the convolutional layers to project the CNN-extracted features to the predicted water desalination performance (i.e., flux and ion rejection rate).

We compared the performance of CNN-based deep learning models with XGBoost³⁸, a widely used shallow machine learning model, which was also trained to predict the water flux/ion rejection rate. The advantage of XGBoost model is that it requires much less time for training compared to CNN. Before the training of the XGBoost model, the graphene membrane was featurized into a one-hot-encoded Morgan fingerprint³⁹ vector of dimension 1024 using RDKit package⁴⁰, with a cutoff distance of 5 Å. The Morgan fingerprint vector was then fed in the XGBoost regression model as input. A random search was conducted on the hyperparameter grid (Supplementary Tables 2 and 3) for model optimization.

The mean squared error (MSE) and coefficient of determination (R^2) are used as metrics to evaluate the performance predictions of models. The water flux and ion rejection labels are standardized before fed into the property prediction models. Thus the metrics tabulated are based on standardized water flux or ion rejection rate (Table 1). Since the accuracy of performance predictor directly influence how accurately the DRL

agent is rewarded/penalized during training, the model with the least MSE and highest R^2 values was chosen to be used for reward estimation. ResNet³² significantly outperformed other models on both metrics, and the fine-tuned ResNet50 model reaches the highest accuracy in predicting both water flux and ion rejection rate. Therefore, a ResNet50 (retrained using the whole dataset) is used to predict the water desalination performance of various graphene nanopores to accelerate the DRL training.

Table 1 Performance of different models for graphene property prediction.

DRL for discovering the optimal graphene nanopores

Our goal was to design the optimal geometry of graphene nanopore for energy-efficient water desalination, which simultaneously demanded high flux and high ion rejection under certain external pressure. In order to optimize the nanopore, an agent was expected to remove atoms sequentially until the desired pore geometry was developed. To this end, the agent was set to interact with graphene nanopores in a sequence of actions a_t , states s_t , and rewards r_t within an episode of length T . The goal of the agent was to select the action such that it could maximize the future discounted return $\mathbb{E}(\sum_{t=1}^T \gamma^{t-1} r_t)$ in the finite Markov decision process (MDP) setting. In our case, we set the discount factor γ to be 1.

At timestep t , given the graphene nanopore G_t , the agent observed the state s_t , which was composed of Morgan fingerprint³⁹, coordinates of all the atoms, along with CNN-extracted graphene geometrical features. The graphene geometry $\{g_t\}$ was fed into the flux and ion rejection predictor, respectively. The geometrical features were the concatenation of last layer before output of the performance predictors. Once an atom was removed, its coordinate was set to the origin since MLP required a homogeneous input dimension. The predicted flux f_t and ion rejection i_t were leveraged to compute the reward signal r_t for the agent, as given in Eqs. (1) and (2):

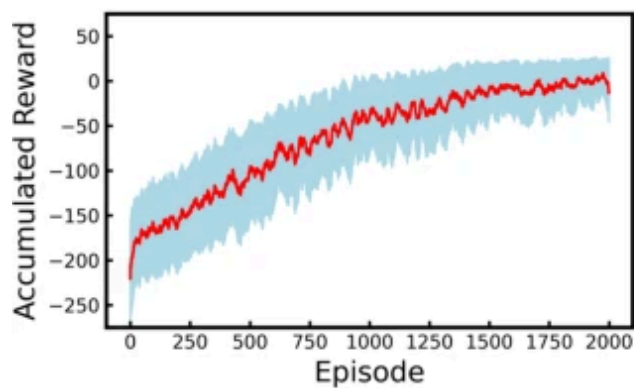
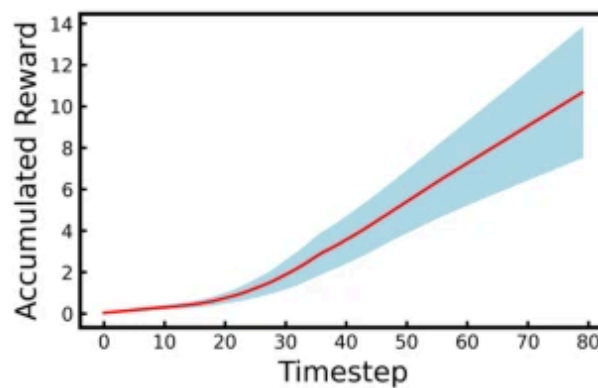
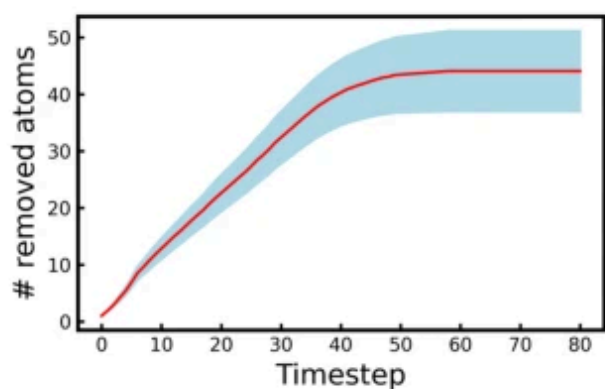
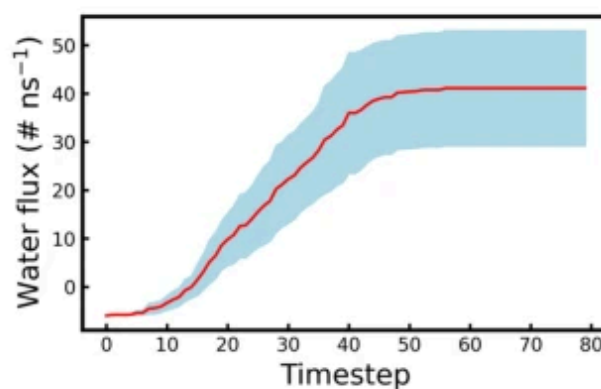
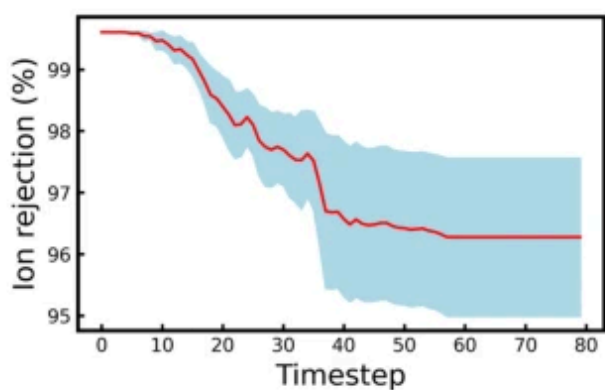
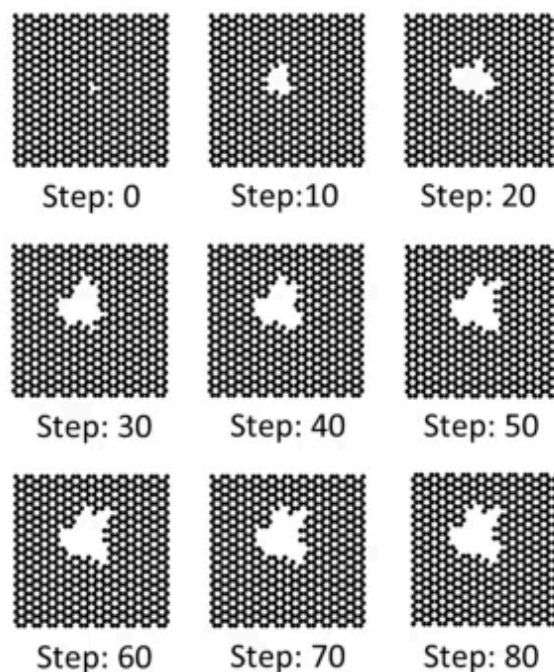
$$\sigma(x) = A + \frac{K-A}{(C+Qe^{-Bx})^{\frac{1}{\nu}}}, \quad (1)$$

$$r_t = \alpha f_t + \sigma(i_t) - \sigma(1), \quad (2)$$

where $\sigma(\cdot)$ is the generalized logistic function⁴¹ and α is the coefficient for flux term. In our setting, α was set to be 0.01, and $A = -15$, $K = 0$, $B = 13$, $Q = 100$, $\nu = 0.01$, $C = 1$ for the logistic function. A linear term of flux reward encouraged the agent to expand nanopores, which would allow higher water flux. Since low ion rejection rate was not favored in water desalination, a generalized logistic function $\sigma(\cdot)$ was leveraged to penalize ion rejection term. When i_t was high, $\sigma(i_t)$ was close to zero, allowing the growth of the nanopores. However, when i_t was low, $\sigma(i_t)$ fiercely penalized the agent by outputting a large negative value (Supplementary Fig. 2). Besides, an extra 0.05 reward was given to the agent when it chose to remove an atom at timestep t to encourage pore growth at an early stage. Given state s_t and reward r_t , the agent intended to choose the action a_t for next step. However, due to the high dimensionality of possible action space (all the atoms in the graphene fragment), it was computationally expensive for the agent to efficiently and thoroughly explore the possible actions and to learn an optimal design. Therefore, only a subset of M atoms was selected as candidates c_t . Atoms on the edge of pore were picked based on the rank of their proximity to the pore center, if the number exceeds M , only the first M atoms closest to the center of pore were selected. However, when the number of edge atoms was less than M , non-edge atoms closest to the center of pore were selected as possible candidates to maintain the size of c_t . Given the state s_t , reward r_t , and candidate c_t , the agent learned to pick the action aiming to maximize future rewards.

We optimized the DRL agent via deep Q-learning¹⁶ with experience replay with 10 random seeds to generate various graphene nanopores. In the DRL agent training processes with different random seeds (Fig. 3), the red curves indicate mean values and the blue shadows represent standard deviations. The accumulated reward for each episode increases during training the DRL agent (Fig. 3a). Initially, the policy is noisy and the accumulated rewards are low because the DRL agent has not yet learned to stop expanding the pore before receiving an enormous penalty for a low ion rejection rate. During the training, the DRL agent gradually learns a stable policy through maximizing the rewards (balancing the trade-off between water flux and ion rejection rate). The performance of DRL agent after 2000 episodes of training is demonstrated in Fig. 3b–e. The DRL agent generates the nanopore which brings a positive reward at each timestep, and the agent also automatically learns to stop enlarging the nanopore to avoid a low ion rejection rate (Fig. 3b, c). For example, the evolution of a DRL-created pore (Fig. 3f, animated in Supplementary Movie) shows that DRL stops removing atom from the edge of graphene nanopore after 50th timestep because it determines that higher water flux reward brought by further removing atoms is not worth the penalty for low ion rejection rate. Based on the prediction of the performance predictor, the DRL-created graphene nanopores have averaged $\sim 40 \text{ ns}^{-1}$ water flux and $\sim 96\%$ ion rejection rate (Fig. 3d, e).

Fig. 3: Training results for 10 DRL agents.

(a)**(b)****(c)****(d)****(e)****(f)**

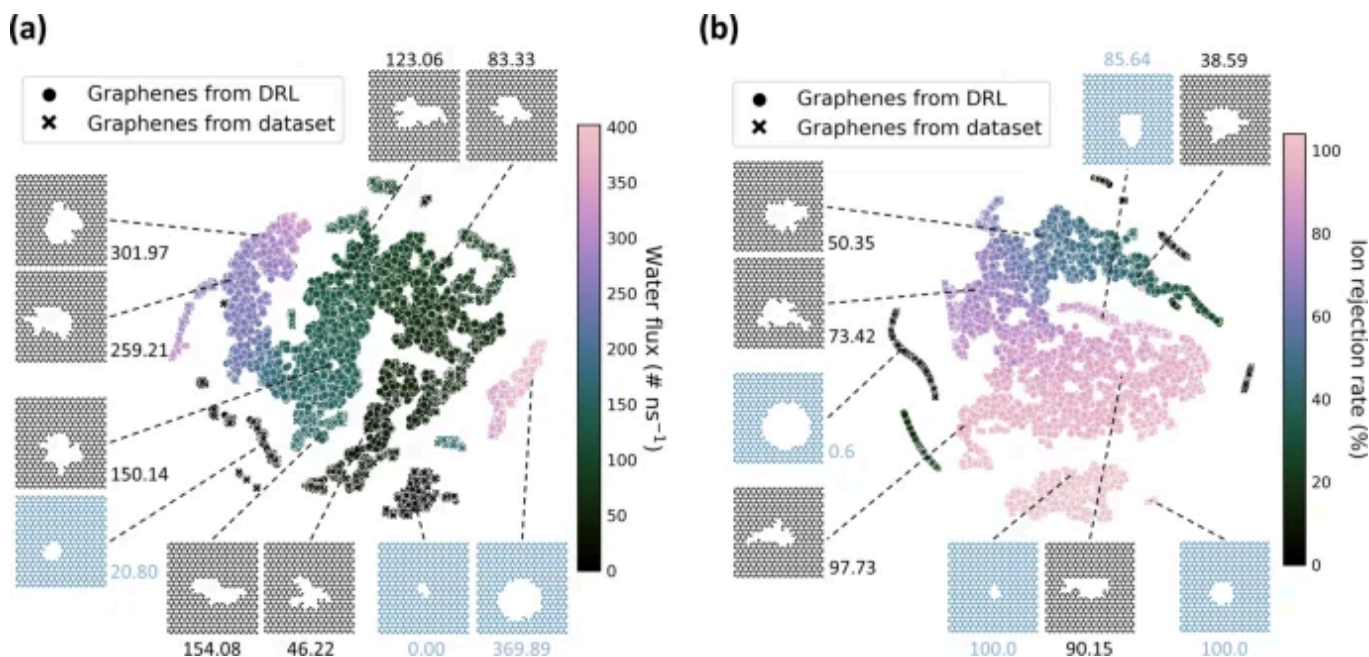
a Summation of reward in each timestep vs. episode, where the red line is the running average of the reward with window size 21 and the blue shadow represents the standard deviation. **b** Summation of reward in each timestep vs.

timestep. **c** Number of removed atoms vs. timestep. **d** Predicted water flux vs. timestep. **e** Predicted ion rejection vs. timestep. **b–e** show the results of DRL agents after trained for 2000 episodes, where the red line indicates the mean and the blue shadow is the standard deviation. **f** Evolution of a graphene nanopore designed by DRL agent.

Investigation on DRL-created graphene nanopores

The collection of both DRL-created graphene nanopores (7999 samples) and nanopores in the training dataset (3937 samples) is visualized using t-SNE⁴² algorithm (Fig. 4). t-SNE maps the high-dimensional features (1000 dimension) extracted from trained CNN models to the low-dimensional domain while preserving the similarity between data points as the relative distance in 2D. In other words, CNN features that are more similar to each other will have a higher tendency of being clustered. In this work, using CNN-extracted features from each graphene membrane, t-SNE successfully clusters samples with similar water flux or ion rejection. Also, as illustrated in Fig. 4, graphenes with different nanoporous structures are far from each other in the plot while those with similar structures are shown close. The results indicate that our CNN model successfully learns to extract features that strongly correlate the water desalination performance (i.e., water flux and ion rejection) with the geometry of the nanopores.

Fig. 4: Visualization of 2D t-SNE embedding of CNN features.

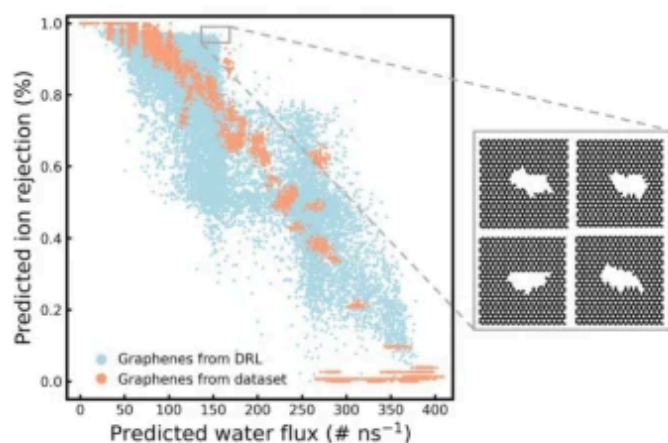
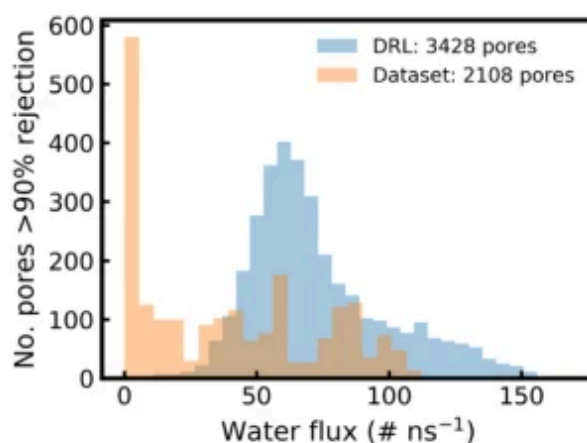
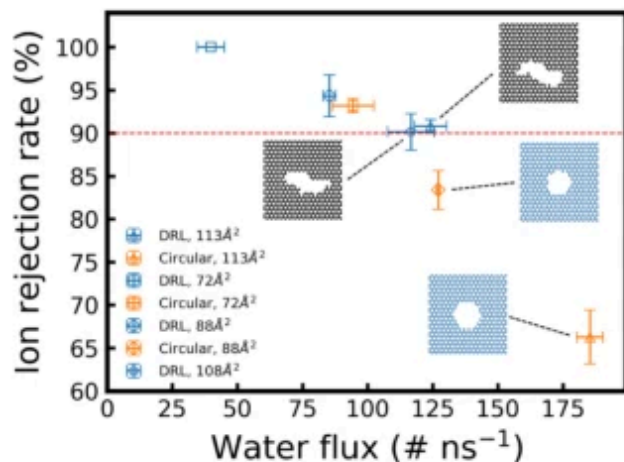
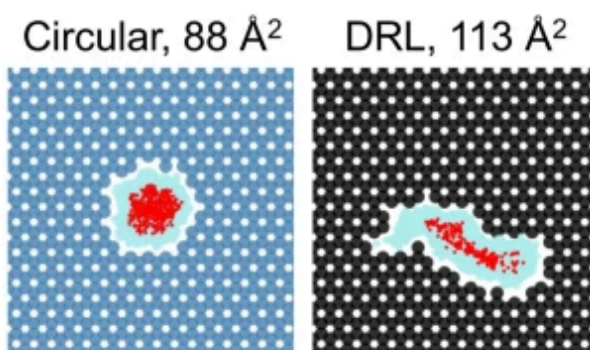
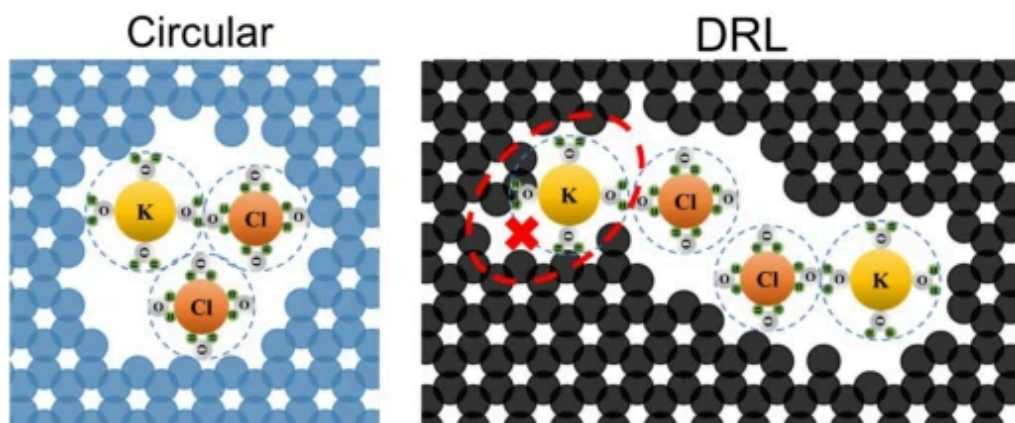


a 2D t-SNE embedding of features extracted from water flux prediction CNN model, where each point is colored by its predicted water flux. **b** 2D t-SNE embedding of features extracted from ion rejection rate prediction CNN model, where each point is colored by its predicted ion rejection. Each axis represents a dimension of the t-SNE embedding.

Dot and X-marks represent graphene nanopores from DRL and the training dataset, respectively. Several graphene nanopores from the training dataset are shown in black and DRL-created membranes are shown in blue.

The water desalination performances of all nanopores, including DRL-created and those in the training dataset, are compared in Fig. 5a. Comparison between permeation rate of nanopores (Supplementary Fig. 4) shows the water flux different normalized of the external pressure. It is worth noting that the process of generating 7999 nanopores using DRL and predicting their water flux/ion rejection rate takes less than a single week; however, evaluating the performance of the same amount of nanopores using MD simulation will take ~33 years (average 36 hrs on each sample, using one 56-core CPU node). Among the nanopores with the same level of ion rejection rate, some nanopores discovered by DRL allow much higher water flux. One common feature shared by those high-performance nanopores is the semi-oval geometry with rough edges. We set 90% ion rejection rate as the threshold to determine if a nanopore can effectively reject ions. The water flux histogram (Fig. 5b) shows that given the baseline ion rejection rate as 90%, DRL can extrapolate from the training dataset and discover graphene nanopores that generally allow higher water flux.

Fig. 5: Analysis of DRL-created graphene nanopores.

(a)**(b)****(c)****(d)****(e)**

a Predicted water flux (ns^{-1}) and ion rejection rate (%) of all graphene nanopores (7999 DRL-created + 3937 in training dataset). Zoom-in window shows the geometries of high-performance nanopores. **b** Histogram of water flux of nanopores with $>90\%$ ion rejection rate. **c** Comparison of water desalination performance (under 100 MPa pressure) of

circular and DRL-created graphene nanopores. Each data point is obtained by averaging the ion rejection and water flux of 4 MD simulations. The error bars represent one standard deviation. **d** Distribution of water molecules (aqua blue) and ions (red) when they are inside of the nanopores (left: circular, area = 88 \AA^2 ; right: DRL-created, area = 113 \AA^2). **e** Schematic showing how ions are blocked by DRL-created pore due to the steric effect. Ions with their hydration shell (blue dashed circle) are too large to pass through the corner area (red dashed circle) in the DRL-created pore, thus rendering that area an ion-free zone.

Further MD simulations are conducted with DRL-created graphene nanopores that show high predicted performances to evaluate how the DRL helps in discovering the optimal graphene nanopore for water desalination (simulation process recorded in Supplementary Movie). Although DRL-created pores generally have lower water flux compared with circular pores with the same area, they have a much higher ion rejection rate (Fig. 5c, 90% threshold of ion rejection rate is marked by a red dashed line). For example, when the pore area is 113 \AA^2 , DRL-created nanopore maintained over 90% ion rejection rate while the circular pore rejects only approximately 65% of ions. A pore with high water flux but a very low ion rejection rate is not desirable in water desalination application. Moreover, the comparison between 113 \AA^2 DRL-created nanopore with 88 \AA^2 circular pore shows that DRL-created pore can reject more ions when achieving the same water flux: they both have approximately 125 \# ns^{-1} water flux while 113 \AA^2 DRL-created pore can reject approximately 7% more ions. The comparison between simulation results proves that DRL tends to prioritize the ion rejection rate over water flux, which makes it capable of maximizing the water flux of nanopores while maintaining a valid ion rejection rate. Nanopores with a larger area result in higher pore density on the graphene membrane. The pore density of the graphene membranes with the above-mentioned nanopores are tabulated in Supplementary Table 4. In real-world experiments or applications, the graphene nanopores can be stabilized by adding passivation such as hydrogen to the edge of the pore⁴³.

To gain a deeper understanding of the reason behind the high ion rejection rate of DRL-created pores, distribution of water molecules and ions inside of 113 \AA^2 DRL-created pore and 88 \AA^2 circular pore have been visualized (Fig. 5d). From the ion distribution (marked by red dots), we can observe that ions can traverse the circular pore evenly through the entire central area of the pore. The distributions of water molecules (marked by aqua blue color) and ions in the circular pore are in a homogeneous pattern. However, the corners inside of DRL-created nanopore are small enough to block the passage of ions while being large enough to accommodate the transport of water molecules. With the knowledge that ions are covered by hydration shell during the transport through the nanopore, it can be seen that ion-free zones (corners) inside of DRL-created nanopore obstruct the traversing of ions with hydration shell by steric effect (Fig. 5e). The perimeter/area ratio can be used as a shape parameter to quantitatively evaluate the influence of geometry on the water desalination performance of nanopores. Due to the rough edges, the comparison of the perimeter/area ratio of DRL-created and circular pores (Supplementary Fig. 6) shows that DRL-created pore generally have higher perimeter/area ratio (Supplementary Table 4). Higher perimeter/area ratio enables DRL-created pores to achieve higher ion rejection rate compared with circular pores with similar water flux or permeation rate. This

is the reason why high-performance nanopores (zoom-in Fig. 5a, more high-performance DRL-created pores shown in Supplementary Fig. 7) all have rough edges. Discovers and utilizes this special geometry, DRL identifies nanopores that can reject most ions while allowing high water transport.

DISCUSSION

In this work, we propose an AI framework that combines the DRL and CNN performance predictor to discover the optimal graphene nanopore for water desalination. The DRL agent takes the current graphene geometrical features and the candidate atoms as inputs to determine which atom to remove at each timestep. Trained with the DQN algorithm, the agent learns to generate nanopores that allow high water flux while maintaining high ion rejection. ResNet50, a widely used CNN model, is trained on a graphene nanopore dataset to instantly predict the water flux and ion rejection rate under certain pressure. Such prediction by the ResNet50 enables the real-time interaction between the DRL agent with the graphene nanopores, as well as the online optimization of the DRL agent. CNN-accelerated DRL training significantly expedites the exploration of graphene nanopores: 7999 different nanopores are created and evaluated for water desalination performance during 1-week training of DRL. Evaluating the same amount of graphene nanopores using MD simulation can take approximately 33 years with a 56-cores CPU cluster. When we set the baseline ion rejection rate to be 90%, DRL shows the capability of extrapolating from the existing training dataset to discover nanopore with higher water flux. Further MD simulations confirm that DRL-created nanopores outperform circular nanopores in terms of ion rejection rate when they have approximately the same water flux. The better water desalination performance of DRL-created pores can be attributed to DRL's utilization of rough edges and small corners to increase the perimeter/area ratio of pores and to block ions with the hydration shell. In conclusion, DRL shows the capability of discovering optimal graphene nanopores for water desalination. Moreover, with only minor modifications, this framework can be directly extended to many other fields concerning nanomaterial design. With a well-trained machine learning property predictor, the DRL can automatically learn to discover the optimal material structure effectively and efficiently.

Methods

MD simulations

MD simulations were conducted using LAMMPS package⁴⁴, where porous graphene membranes simulated were either created using Visual Molecular Dynamics⁴⁵ or automatically generated by DRL agent (samples from the early stage of training). All water molecules in this work were simulated using SPC/E model⁴⁶, with SHAKE⁴⁷ algorithm to constrain the bond length and angles. Lennard–Jones (LJ) potentials (Supplementary Table 1) along with long-range Coulombic electrostatic potentials were adopted as interatomic potentials in the MD simulation. The cutoff for the interatomic potentials was set to be 12 Å. Lorentz–Berthelot rules were employed for the calculation of LJ potentials between different kinds of atoms. Particle-particle particle-mesh (PPPM) Ewald solver⁴⁸ with 0.005 root-mean-squared error was used for long-range Coulombic potential

correction. The porous graphene membrane and piston were each regarded as an entity during the simulation (internal interatomic potentials were not calculated) in order to reduce the computational cost.

In the first stage of each individual simulation, the internal energy of the system was minimized for 1000 iterations. The system then ran for 5 ps under the *NPT* (isothermal–isobaric) ensemble at 300 K after the velocities of molecules were initialized based on Gaussian distribution. After the equilibrating, the system under *NPT* ensemble, the system was switched to *NVT* (canonical) ensemble to run for another 10 ns. The temperature was maintained at 300 K by Nosé–Hoover thermostat^{49,50} with a time constant of 0.5 ps. At this stage, a z-direction constant external pressure of 100 MPa was applied on saline water by the piston to mimic the RO process in water desalination. Since the relationship between water flux and external pressure in the RO process was generally linear^{9,11,12,13}, the performance of pores under 100 MPa could be extrapolated to lower pressures. Therefore, we chose to run simulations under 100 MPa external pressure to rapidly collect meaningful data. Molecular trajectories of each simulation were collected every 5 ps for data processing. Data augmentation was conducted using the Atomic Simulation Environment (ASE) package⁵¹. Area and perimeter of the graphene nanopores are calculated using computer vision methods (details in Supplementary Fig. 5).

CNN water desalination performance predictor

There were two steps in the CNN modeling, including extracting features from the geometry of graphene nanopore and making predictions through an MLP regression model. First of all, the geometrical features of a graphene nanopore were extracted to a 380×380 pixels representation. Color was applied on top of each atom, and all geometrical features were resized to the dimension of 224×224 pixels. The processed geometrical features were then fed into a CNN. Multiple CNN models, including ResNet18, ResNet50 (ref. ³²), and VGG16 (ref. ³¹) with batch normalization, were benchmarked based on the MSE and R^2 of their resulting water flux/ion rejection rate predictions. An extracted feature vector with the dimension of 1000 was output from the CNN model. Finally, given the feature vector, the MLP was able to make predictions of flux and ion rejection rates. The MLP used in this work consisted of two layers with 256 and 64 neurons in the first and second layers, respectively. A residual block³² and ReLU⁵² activation function were added after each layer of MLP. Two CNN models were trained: one for the prediction of water flux and the other for ion rejection rate.

The CNN models, including VGG³¹ and ResNet³², were implemented based on PyTorch library⁵³ and pre-trained on the ImageNet dataset⁵⁴ to learn the robust CNN feature extractor. A random-initialized MLP was built on top of the convolutional layers to project the CNN-extracted features to the predicted water desalination performance (i.e., flux and ion rejection rate). In training the deep learning models on our graphene dataset, we used gradient-based Adam optimizer⁵⁵ with the learning rate 0.0001 and 0.001 for pre-trained convolutional layers and the MLPs, respectively. The whole graphene dataset was split into a training set and a test set with the ratio of 4:1, and the models were trained only on the training set for 600 epochs and evaluate on the test set. The model which reached the best performance (i.e., lowest MSE in predicting

the flux/ion rejection rate) on the test set was selected as the water desalination performance predictor in the DRL framework. These strategies in CNN training maintains the robust and informative CNN feature extractors in the pre-trained CNN models and avoided the model from overfitting the graphene dataset.

DRL agent

To train the agent, deep Q-learning¹⁶ with experience replay was implemented. Our task only considered deterministic environment, namely given the pair (s, c) and the action a , $((s^{\prime}, c^{\prime}))$ at the next timestep was determined. Based on Bellman equation¹⁷, the optimal action-value function $Q^*(s, c)$ in the deterministic environment was defined as

$$Q^*(s, c) = r + \gamma \max_{c'} Q^*(s', c')$$

(3)

To model the Q function, the Q -network parameterized by θ and target network parameterized by (θ') , two fully connected networks with identical architecture were built. During training, only the parameters θ in the Q -network were updated through backpropagation from the loss function. The parameters (θ') in the target network were updated with θ every 10 steps and are kept fixed otherwise. The input to the network was the pair of graphene state and action candidates, (s, c) , and the output was the Q values of all the actions in the candidate. The agent then picked the action with the highest Q value. In addition, the agent's experience $((s, c, r, s'))$ in the episodes were stored to a replay buffer (\mathcal{D}) ¹⁶, such that the experience can be leveraged to update the network parameters multiple times. During training, a mini-batch of samples was drawn uniformly at random from the replay buffer $((s, c, a, r, s')) \sim U(\mathcal{D})$. The loss function (Eq. (4)) measured the difference between the target Q value $(Q^*(s', c'); \theta')$ and the prediction of current Q -network $Q(s, c; \theta)$:

$$L_i(\theta) = \mathbb{E}_{(s, c, r, s') \sim U(\mathcal{D})} \left[\left(r + \gamma \max_{a'} Q(s', c'; \theta') - Q(s, c; \theta) \right)^2 \right]$$

(4)

In our setting, we use an Adam optimizer⁵⁵ with learning rate 0.001. The replay buffer is of capacity 10,000 and batch size is set to 128.

Data availability

The graphene nanopore desalination performance dataset generated during the current study are available in the Github repository (<https://github.com/BaratiLab/Graphene-RL/tree/main/data>).

Code availability

The code accompanying this work can be found at <https://github.com/BaratiLab/Graphene-RL>.

References

1. Jiang, D.-E., Cooper, V. R. & Dai, S. Porous graphene as the ultimate membrane for gas separation. *Nano Lett.* **9**, 4019–4024 (2009).
 2. Li, H. et al. Ultrathin, molecular-sieving graphene oxide membranes for selective hydrogen separation. *Science* **342**, 95–98 (2013).
 3. Kim, H. W. et al. Selective gas transport through few-layered graphene and graphene oxide membranes. *Science* **342**, 91–95 (2013).
 4. Wang, Y. et al. Supercapacitor devices based on graphene materials. *J. Phys. Chem. C* **113**, 13103–13107 (2009).
 5. Liu, C., Yu, Z., Neff, D., Zhamu, A. & Jang, B. Z. Graphene-based supercapacitor with an ultrahigh energy density. *Nano Lett.* **10**, 4863–4868 (2010).
 6. Farimani, A. B., Min, K. & Aluru, N. R. DNA base detection using a single-layer mos2. *ACS Nano* **8**, 7914–7922 (2014).
-

7. Barati Farimani, A., Dibaeinia, P. & Aluru, N. R. DNA origami–graphene hybrid nanopore for DNA detection. *ACS Appl. Mater. Interfaces* **9**, 92–100 (2017).

8. Schneider, G. F. et al. DNA translocation through graphene nanopores. *Nano Lett.* **10**, 3163–3167 (2010).

9. Cohen-Tanugi, D. & Grossman, J. C. Water desalination across nanoporous graphene. *Nano Lett.* **12**, 3602–3608 (2012).

10. Surwade, S. P. et al. Water desalination using nanoporous single-layer graphene. *Nat. Nanotechnol.* **10**, 459–464 (2015).

11. Heiranian, M., Farimani, A. B. & Aluru, N. R. Water desalination with a single-layer mos 2 nanopore. *Nat. Commun.* **6**, 1–6 (2015).

12. Cao, Z., Liu, V. & Barati Farimani, A. Water desalination with two-dimensional metal–organic framework membranes. *Nano Lett.* **19**, 8638–8643 (2019).

13. Cao, Z., Liu, V. & Barati Farimani, A. Why single-layer mos2 is a more energy efficient membrane for water desalination?. *ACS Energy Lett.* **5**, 2217–2222 (2020).

14. Meidani, K., Cao, Z. & Barati Farimani, A. Titanium carbide mxene for water desalination: a molecular dynamics study. *ACS Appl. Nano Mater.* **4**, 6145–6151 (2021).

15. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

16. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).

17. Sutton, R. S., Barto, A. G. et al. *Introduction to Reinforcement Learning* Vol. 135 (MIT Press, Cambridge, 1998).

18. Mnih, V. et al. Playing atari with deep reinforcement learning. Preprint at <https://arxiv.org/abs/1312.5602> (2013).

19. Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4**, eaap7885 (2018).

20. Karamad, M. et al. Orbital graph convolutional neural network for material property prediction. *Phys. Rev. Mater.* **4**, 093801 (2020).

21. Yao, Z. et al. Inverse design of nanoporous crystalline reticular materials with deep generative models. *Nat. Mach. Intell.* **3**, 76–86 (2021).

22. Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **9**, 1–10 (2019).

23. Russo, C. J. & Golovchenko, J. A. Atom-by-atom nucleation and growth of graphene nanopores. *Proc. Natl. Acad. Sci. USA* **109**, 5953–5957 (2012).

24. Feng, J. et al. Electrochemical reaction in single layer mos2: nanopores opened atom by atom. *Nano Lett.* **15**, 3431–3438 (2015).

25. Fischbein, M. D. & Drndić, M. Electron beam nanosculpting of suspended graphene sheets. *Appl. Phys. Lett.* **93**, 113107 (2008).

26. Wang, L. et al. Fundamental transport mechanisms, fabrication and potential applications of nanoporous atomically thin membranes. *Nat. Nanotechnol.* **12**, 509 (2017).
-
27. Moreno, C. et al. Bottom-up synthesis of multifunctional nanoporous graphene. *Science* **360**, 199–203 (2018).
-
28. Guirguis, A. et al. Perforation routes towards practical nano-porous graphene and analogous materials engineering. *Carbon* **155**, 660–673 (2019).
-
29. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
-
30. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**, 84–90 (2017).
-
31. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at <https://arxiv.org/abs/1409.1556> (2014).
-
32. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778 (IEEE, 2016).
-
33. Guirguis, A. et al. Applications of nano-porous graphene materials—critical review on performance and challenges. *Mater. Horiz.* **7**, 1218–1245 (2020).
-
34. Farimani, A. B., Heiranian, M. & Aluru, N. R. Electromechanical signatures for dna sequencing through a mechanosensitive nanopore. *J. Phys. Chem. Lett.* **6**, 650–657 (2015).
-

35. Farimani, A. B., Heiranian, M. & Aluru, N. R. Identification of amino acids with sensitive nanoporous membranes: towards machine learning-based prediction. *NPJ 2D Mater. Appl.* **2**, 1–9 (2018).
-
36. Perez, L. & Wang, J. The effectiveness of data augmentation in image classification using deep learning. Preprint at <https://arxiv.org/abs/1712.04621> (2017).
-
37. Van Dyk, D. A. & Meng, X.-L. The art of data augmentation. *J. Comput. Graph. Stat.* **10**, 1–50 (2001).
-
38. Chen, T. & Guestrin, C. Xgboost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794 (ACM, 2016).
-
39. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**, 742–754 (2010).
-
40. Landrum, G. Rdkit: open-source cheminformatics software. GitHub and SourceForge Vol. 10, 3592822 (2016).
-
41. Richards, F. A flexible growth function for empirical use. *J. Exp. Bot.* **10**, 290–301 (1959).
-
42. Maaten, Lvd & Hinton, G. Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
-
43. Lee, J. et al. Stabilization of graphene nanopore. *Proc. Natl. Acad. Sci. USA* **111**, 7522–7526 (2014).
-
44. Plimpton, S. *Fast Parallel Algorithms for Short-Range Molecular Dynamics*. Tech. Rep. (Sandia National Labs., Albuquerque, NM, 1993).
-
45. Humphrey, W., Dalke, A. & Schulten, K. et al. Vmd: visual molecular dynamics. *J. Mol. Graph.* **14**, 33–38 (1996).
-

46. Mark, P. & Nilsson, L. Structure and dynamics of the tip3p, spc, and spc/e water models at 298 k. *J. Phys. Chem. A* **105**, 9954–9960 (2001).
-
47. Ryckaert, J.-P., Ciccotti, G. & Berendsen, H. J. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.* **23**, 327–341 (1977).
-
48. Plimpton, S., Pollock, R. & Stevens, M. Particle-mesh Ewald and rRESPA for parallel molecular dynamics simulations. In *Proc. Eighth SIAM Conference on Parallel Processing for Scientific Computing* (SIAM, 1997).
-
49. Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **81**, 511–519 (1984).
-
50. Hoover, W. G. Canonical dynamics: equilibrium phase-space distributions. *Phys. Rev. A* **31**, 1695 (1985).
-
51. Larsen, A. H. et al. The atomic simulation environment-a python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002 (2017).
-
52. Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning* (PMLR, 2010).
-
53. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. et al. (eds.) *Advances in Neural Information Processing Systems* **32**, 8024–8035 (Curran Associates, Inc., 2019).
-
54. Deng, J. et al. Imagenet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 248–255 (2009).
-
55. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).
-

Acknowledgements

The authors thank the supercomputing resource Arjuna provided by the Pittsburgh Supercomputing Center (PSC). This work is supported by the start-up fund provided by CMU Mechanical Engineering.

Author information

These authors contributed equally: Yuyang Wang, Zhonglin Cao.

Authors and Affiliations

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

Yuyang Wang, Zhonglin Cao & Amir Barati Farimani

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

Amir Barati Farimani

Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA

Amir Barati Farimani

Contributions

A.B.F. conceived the project, Y.W. and Z.C. designed and coded the DRL architecture, Y.W. coded the CNN performance predictor, Z.C. performed the MD simulations, all authors wrote the manuscript. Y.W. and Z.C. contributed equally to this work and are considered co-first authors.

Corresponding author

Correspondence to [Amir Barati Farimani](#).

Ethics declarations

Competing interests

The authors declare no competing interests.

Additional information

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supplementary information

[Supplementary Information](#)

[Supplementary Video 1](#)

[Supplementary Video 2](#)

[Supplementary Video 3](#)

Rights and permissions

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

[Reprints and permissions](#)

About this article

Cite this article

Wang, Y., Cao, Z. & Barati Farimani, A. Efficient water desalination with graphene nanopores obtained using artificial intelligence. *npj 2D Mater Appl* **5**, 66 (2021). <https://doi.org/10.1038/s41699-021-00246-9>

Received

17 March 2021

Accepted

22 June 2021

Published

12 July 2021

DOI

<https://doi.org/10.1038/s41699-021-00246-9>

Share this article

Anyone you share the following link with will be able to read this content:

[Get shareable link](#)

Provided by the Springer Nature SharedIt content-sharing initiative

Subjects [Graphene](#) • [Mathematics and computing](#) • [Two-dimensional materials](#)

npj 2D Materials and Applications (*npj 2D Mater Appl*) | ISSN 2397-7132 (online)