
Efficient Nanopore Optimization by CNN-accelerated Deep Reinforcement Learning

Yuyang Wang^{*†}, Zhonglin Cao^{*†}, Amir Barati Farimani^{‡§†}

[†]Department of Mechanical Engineering

[‡]Machine Learning Department

Carnegie Mellon University, Pittsburgh, PA 15213, USA

Abstract

Two-dimensional nanomaterials, such as graphene, have been extensively studied because of their outstanding physical properties. Structure and geometry optimization of nanopores on such materials is beneficial for their performance in real-world engineering applications such as water desalination. However, the optimization process often involves very large numbers of experiments or simulations which are expensive and time-consuming. In this work, we propose a graphene nanopore optimization framework via the combination of deep reinforcement learning (DRL) and convolutional neural network (CNN) for efficient water desalination. The DRL agent controls the geometry of nanopore, while the CNN is employed to predict the water flux and ion rejection of the nanoporous graphene membrane at a certain external pressure. With the CNN-accelerated property prediction, our DRL agent can optimize the nanoporous graphene efficiently in an online manner. Experiments show that our framework can design nanopore structures that are promising in energy-efficient water desalination.

1 Introduction

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 material for energy storage or building supercapacitor [4, 5], and high-resolution DNA sequencing [6, 7, 8]. Given the potential imminent global water scarcity crisis, another important application with graphene and other 2D materials is energy-efficient water desalination [9, 10, 11, 12]. Equipping nanoporous 2D material membranes, reverse osmosis water desalination process can expect 2-3 orders improvement in water flux compared with traditional polymeric membranes [10, 11, 12, 13]. The pore geometry plays a determinant role in the performance of nanoporous 2D materials for water desalination [9, 11]. A large pore that allows high water flux may perform poorly in rejecting ions; a small pore that rejects 100% unwanted ions, on the other hand, may have limited water flux. Thus, an optimal nanopore for water desalination should be able to allow as much 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. To maximize the performances of the nanoporous graphene membrane in the water desalination process, a fast and inexpensive nanopore optimization method with accelerated nanopore water desalination performance predictor (performance predictor in short) is in need. Inspired by the recent rapid development of

^{*}Equal contribution.

[†]Corresponding author: barati@cmu.edu.

deep learning [14] algorithms, we combined the state-of-art deep reinforcement learning (DRL) algorithm with convolutional neural network (CNN) to solve this challenge.

The core concept of reinforcement learning (RL) [15] is to train an agent that actively interacts with the environment to achieve a goal. Recently, DRL [16, 17], 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 [18, 19, 20] and molecule optimization [21]. In this work, we designed and implemented an artificial intelligence framework consisting of DRL, which is capable of designing the nanopore on a single-layer graphene sheet to reach optimal water desalination performance. By a series of decisions on whether or not to remove carbon atoms and which atom to remove, the DRL agent can eventually create a pore that allows the highest water flux while maintaining an acceptable ion rejection rate threshold. During the training process, the DRL agent’s decision-making evolves based on the feedback given by the water desalination performance predictor about the agent’s action. In our case, the feedback is the resulting water desalination performance (e.g., water flux and ion rejection rate) of the pore generated by the DRL agent. Currently, the conventional way to obtain the water flux and ion rejection of a porous graphene membrane is through molecular dynamics (MD) simulations. However, 10 ns of MD simulation can take up to several days to run, thus rendering the feedback process too time-consuming for DRL to be practical. To overcome the limitation of the conventional MD simulation method, we trained a CNN [22, 23, 24, 25] model to directly predict water desalination performance from a porous graphene membrane structure. With the CNN-accelerated performance estimation process, the DRL model can rapidly optimize graphene nanopore for water desalination.

2 Method

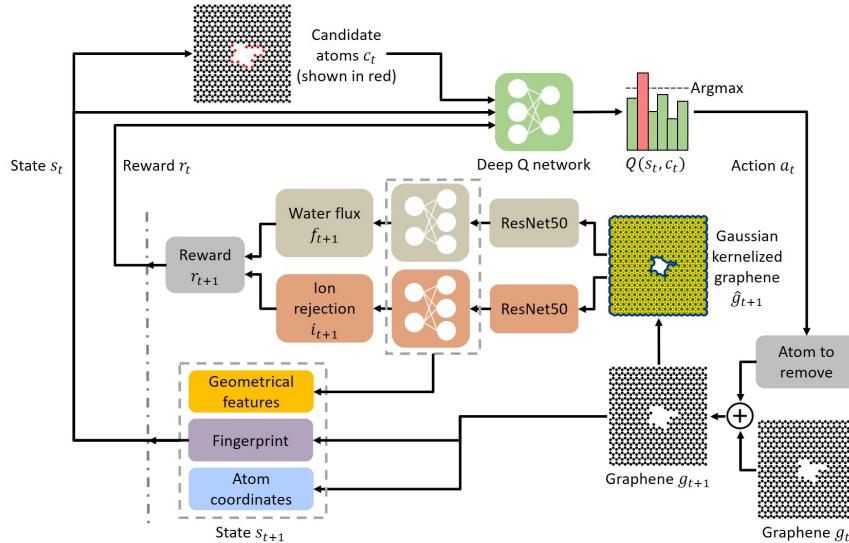


Figure 1: Overview of CNN accelerated DRL nanopore design model. At each timestep, the nanoporous graphene structure is transformed into an geometrical features, which is fed into a performance prediction network to estimate water flux and ion rejection rate. The reward is then calculated based on the predicted water flux and ion rejection. Also, the geometrical features extracted from the performance predictor is concatenated with the fingerprint and atom coordinates for the current state. Given the current graphene structure, candidate atoms are picked which locate at the edge of the nanopore. The RL agent constructed upon Deep Q-network takes as input the reward, candidate atoms, and state to determine the next atom to remove from the graphene.

The graphene nanopore optimization framework for water desalination consists of a DRL agent incorporated with a CNN-based water desalination performance predictor (Fig. 1). Whenever the DRL generates a new nanopore, the performance predictor can rapidly evaluate the water flux/ion rejection rate of the nanopore, such that the DRL agent can get instantaneous feedback on its action. Provided the featurized information of the nanoporous graphene sheet (Morgan fingerprint, Cartesian

coordinates of each atom, and geometry of graphene membrane) and predicted water flux and ion rejection by the performance predictor, DRL agent was trained to create a pore on graphene sheet with the goal to maximize its performance in the water desalination process.

2.1 Water Desalination Performance Predictor

To facilitate the reward estimation of DRL, a CNN model was trained to make an instantaneous prediction of water flux and ion rejection rate of a specific nanoporous graphene membrane. There were 2 steps in the CNN modeling, including extracting features from the geometry of graphene nanoporous membrane and making predictions through a fully-connected multi-layer perceptron (MLP) regression model (Fig.1). First of all, the geometrical features of a graphene nanoporous membrane is extracted to a 380×380 pixels image. 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 was then fed into a CNN for feature extraction. Multiple CNN models, including ResNet18, ResNet50 [25], and VGG16 [24] with batch normalization were implemented to extract feature vectors from the input image. Finally, a MLP was built on top of the CNN to make the prediction. We trained two models for water flux and ion rejection prediction separately.

The dataset used for CNN training is generated by MD simulation. The MD simulation system of water desalination using graphene is identical to previously published works [11, 13]. There were 185 different graphene nanopores simulated. To enlarge the dataset for CNN training, we conducted data augmentation by translating or flipping the pore on the membrane. The dataset after data augmentation consisted of 3937 samples. During training, we used gradient-based Adam [26] optimizer with the learning rate 0.0001 for pretrained convolutional layers and 0.001 for the MLPs. For benchmarking the prediction performance of each model, the whole dataset was split into a training set and a test set in the ratio of 4:1. All models were trained only on the training set and tested based on the unseen test set. Raw values of water flux and ion rejection rate were standardized before fed into prediction models for training. CNN models were trained for 600 epochs. The model with the best performance was selected to be retrained on the whole dataset and utilized in the DRL framework.

2.2 DRL Nanopore Design Agent

Our goal was to design the 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 design the nanopore, an agent was expected to remove atoms sequentially till a desired pore geometry is developed. To this end, the agent was set to interact with nanoporous graphene 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 $R_t = \sum_{t=1}^T \gamma^{t-1} r_t$ 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 nanoporous graphene G_t , the agent observed the state s_t , which was composed of Morgan fingerprint [27], coordinates of all the atoms, along with graphene geometrical features. The colored graphene g'_t was fed into the flux and ion rejection predictor respectively. The fully connected layer before the output was extracted and concatenated as the geometrical feature. Once an atom is removed, its coordinate is set to the origin since a homogeneous input dimension is required for MLP. 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 Eq. 1 and Eq. 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 $\sigma(\cdot)$ was the generalized logistic function [28] and α was 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 generate nanopores, which allowed high 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, which allowed the growth of the nanopores. However, when i_t was low, $\sigma(i_t)$ fiercely penalized the agent by outputting a large negative value. 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 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.

To train the agent, deep Q-learning [17] with experience replay was implemented. To model the Q function, the Q-network parameterized by θ and target network parameterized by θ' , two fully connected networks with the identical architecture were built. During training, only the parameters θ in the Q-network was updated through backpropagation from loss function. Whereas 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. Besides, the agent's experience (s, c, r, s') in the episodes were stored to a replay buffer \mathcal{D} [17], such that the experience can be leveraged to update the network parameters multiple times. The loss function (Eq. 3) measured the difference between the target Q value $Q^*(s', c'; \theta'_i)$ and the prediction of current Q network $Q(s, c; \theta_i)$:

$$L_i(\theta_i) = \mathbb{E}_{(s, c, r, s') \sim U(\mathcal{D})} \left[\left(r + \gamma \max_{a'} Q(s', c'; \theta'_i) - Q(s, c; \theta_i) \right)^2 \right]. \quad (3)$$

In our setting, we use Adam optimizer [26] with learning rate 0.001. The relay buffer is of capacity 10000 and batch size is set to 128.

3 Results

The mean squared error (MSE) and coefficient of determination (R^2) are used as metrics to evaluate the performances 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. ResNet [25] significantly outperformed other models on both metrics (Table 1), and the fined-tuned ResNet50 model reaches the highest accuracy in predicting both water flux and ion rejection rate. Since the accuracy of performance predictor directly influence how accurately the DRL agent is rewarded/penalized during training, ResNet50 is used to predict the water desalination performance of various nanoporous graphenes to accelerate the DRL training.

Table 1: Performance of different models for graphene property prediction.

Model	Flux MSE	Flux R^2	Ion rejection MSE	Ion rejection R^2
VGG16 [24]	0.0448	0.957	0.0156	0.985
ResNet18 [25]	0.0024	0.998	0.0039	0.996
ResNet50 [25]	0.0022	0.998	0.0038	0.996

We trained the DRL agent with 10 random seeds to generate various graphene nanopores. In the DRL agent training processes with different random seeds (Fig.2), the red curves indicate mean values and the blue shadows represent one standard deviation. The accumulated reward for each episode increases during training the DRL agent (Fig. 2a). 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 enormous penalty for low ion rejection rate. During the training, the DRL agent gradually learns a stable policy through maximizing the rewards (balancing the tradeoff between water flux and ion rejection rate). DRL agent performances after 2000 episodes of training are demonstrated in Fig. 2(b)-(e). The DRL agent generates nanopore which gives positive reward at each timestep, and the agent also automatically learns to stop enlarging the nanopore to avoid low ion rejection rate (Fig. 2b and 2c). For example, the evolution of a DRL generated pore (Fig. 2f) 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

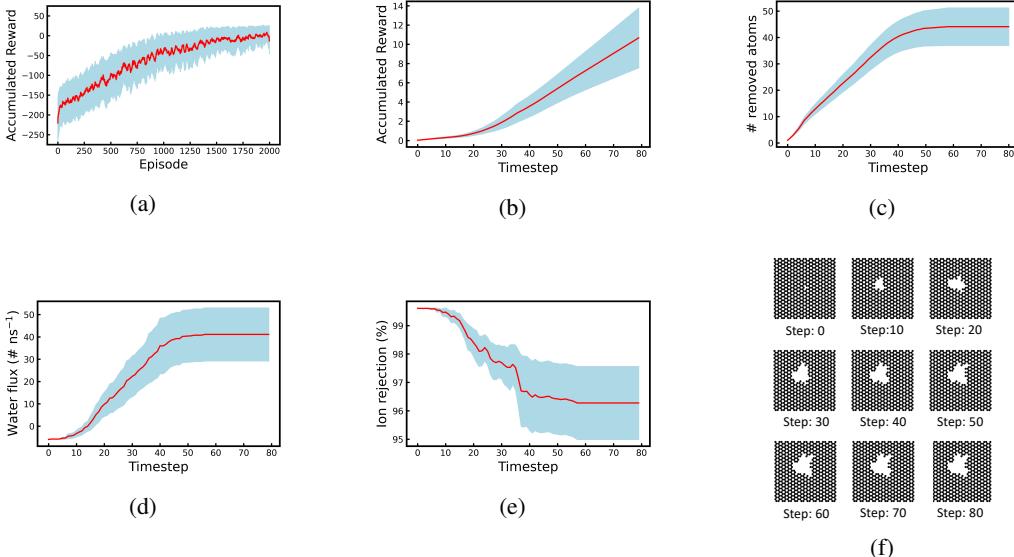


Figure 2: Training results for 10 DRL agents. (a) Summation of reward in each timestep vs. episode, where the red line is the running average of the reward with window size 21 and the blue shadow represents the standard deviation. (b) Summation of reward in each timestep vs. timestep. (c) Number of removed atoms vs. timestep. (d) Predicted water flux vs. timestep. (e) Predicted ion rejection vs. timestep. Fig. 2(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.

ion rejection rate. Based on the prediction of our fine-tuned ResNet50, the averaged DRL generated nanoporous graphenes possess water flux of $\sim 40 \text{ #/ns}$ and ion rejection $\sim 96\%$ (Fig. 2d and 2e).

The collection of both DRL generated nanoporous graphene membranes (7999 samples) and membranes in the training dataset (3937 samples) is visualized using T-SNE[29] algorithm (Fig. 3a and 3b). T-SNE is a dimensionality reduction tool that is capable of mapping high-dimensional data to lower-dimension form while preserving the similarities between data points. In other words, samples 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 clustered samples with similar water flux or ion rejection. This result indicates that features extracted from CNN models have a strong correlation with the water flux and ion rejection rate.

The water desalination performances of all nanopores (DRL generated and training dataset) are compared in Fig. 4a. 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 hours on each sample, using one 56-core CPU node). Compared to training dataset nanopores with 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.

Further MD simulations are conducted with DRL generated membranes to evaluate how the DRL helps in optimizing graphene nanopore for water desalination. The area of each selected graphene nanopore is calculated, and its water desalination performance is compared with that of the circular pore with the same area (Fig. 4b, 90% threshold of ion rejection rate is marked by a red dashed line). Although DRL generated pores generally have lower water flux compared with circular pores with the same area, they have much higher ion rejection rate. For example, when the pore area is 113\AA^2 , DRL generated nanopore maintained over 90% ion rejection rate while the circular pore rejects only approximately 65% of ions even though allowing higher water flux. 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 generated nanopore with 88\AA^2 circular pore shows that DRL

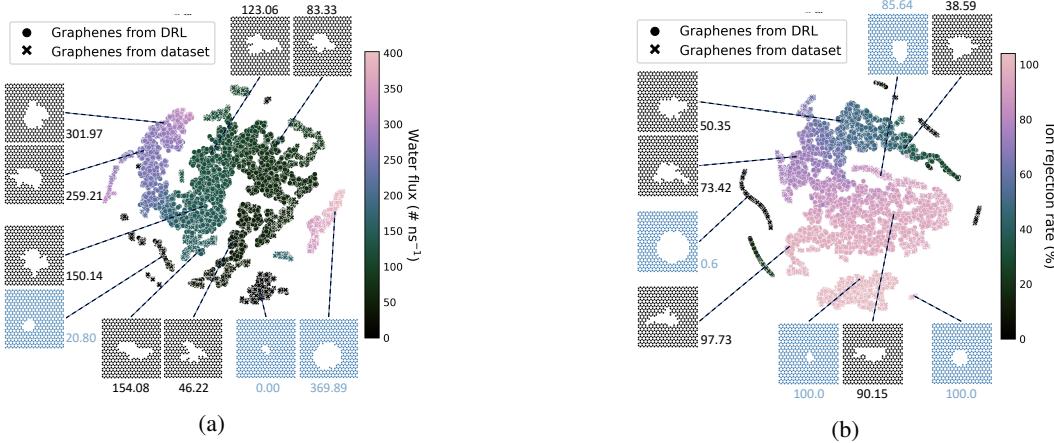


Figure 3: (a) T-SNE[29] visualization of features extracted from water flux prediction CNN model. (b) T-SNE[29] of features extracted from ion rejection rate prediction CNN model. Black colored are DRL generated membranes and blue ones are from training dataset

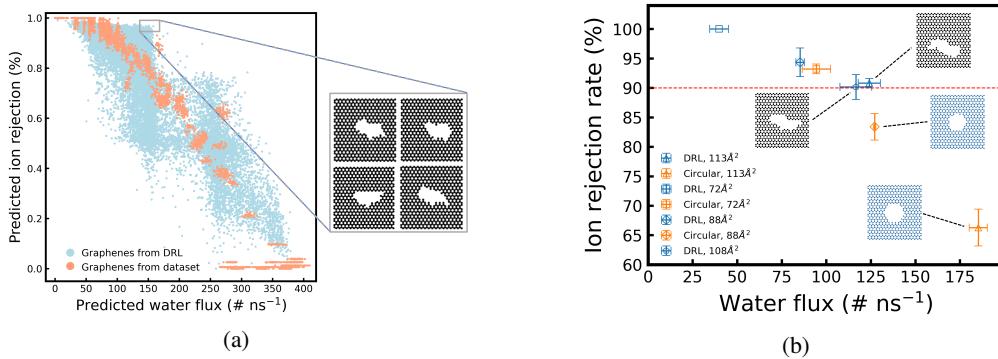


Figure 4: (a) Predicted water flux (ns^{-1}), and ion rejection rate (%) of all graphene nanopores (7999 DRL generated + 3937 in training dataset). Zoom-in window shows the geometries of high-performance nanopores. (b) Comparison of water desalination performance (under 100 MPa pressure) of circular and DRL generated 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.

generated pore can reject more ions when achieving same water flux: they both have approximately 125 #/ns water flux while 113 \AA^2 DRL generated 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.

4 Conclusion

In this work, we propose a graphene nanopore optimization framework based on the DRL accelerated by the learning-based material property predictor. In particular, we focus on the optimal design of nanoporous graphene for water desalination. The DRL agent takes the current graphene structure information and the candidate atoms as inputs to determine which atom to remove at each timestep. If we use traditional MD simulation to give DRL agent feedbacks about how its actions affect the water desalination performance of the graphene nanopore, the training process of the DRL agent can be impractically time-consuming. Therefore, ResNet50, a widely used CNN model, is trained on a nanoporous graphene dataset to instantly predict the water flux and ion rejection rate under certain pressure. Such prediction by the CNN model 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 and is capable of designing graphene nanopore to maintain a high ion rejection rate while maximizing water flux. Moreover, with only minor modification, this framework can be directly extended to many other fields concerning nanopore design. With a well-trained machine learning property predictor, the DRL can automatically learn to design the optimal material structure effectively and efficiently.

References

- [1] De-en Jiang, Valentino R Cooper, and Sheng Dai. Porous graphene as the ultimate membrane for gas separation. *Nano letters*, 9(12):4019–4024, 2009.
- [2] Hang Li, Zhuonan Song, Xiaojie Zhang, Yi Huang, Shiguang Li, Yating Mao, Harry J Ploehn, Yu Bao, and Miao Yu. Ultrathin, molecular-sieving graphene oxide membranes for selective hydrogen separation. *Science*, 342(6154):95–98, 2013.
- [3] Hyo Won Kim, Hee Wook Yoon, Seon-Mi Yoon, Byung Min Yoo, Byung Kook Ahn, Young Hoon Cho, Hye Jin Shin, Hoichang Yang, Ungyu Paik, Soongeun Kwon, et al. Selective gas transport through few-layered graphene and graphene oxide membranes. *Science*, 342(6154):91–95, 2013.
- [4] Yan Wang, Zhiqiang Shi, Yi Huang, Yanfeng Ma, Chengyang Wang, Mingming Chen, and Yongsheng Chen. Supercapacitor devices based on graphene materials. *The Journal of Physical Chemistry C*, 113(30):13103–13107, 2009.
- [5] Chenguang Liu, Zhenning Yu, David Neff, Aruna Zhamu, and Bor Z Jang. Graphene-based supercapacitor with an ultrahigh energy density. *Nano letters*, 10(12):4863–4868, 2010.
- [6] Amir Barati Farimani, Kyoungmin Min, and Narayana R Aluru. Dna base detection using a single-layer mos2. *ACS nano*, 8(8):7914–7922, 2014.
- [7] Amir Barati Farimani, Payam Dibaeinia, and Narayana R Aluru. Dna origami–graphene hybrid nanopore for dna detection. *ACS Applied Materials & Interfaces*, 9(1):92–100, 2017.
- [8] Grégory F Schneider, Stefan W Kowalczyk, Victor E Calado, Grégory Pandraud, Henny W Zandbergen, Lieven MK Vandersypen, and Cees Dekker. Dna translocation through graphene nanopores. *Nano letters*, 10(8):3163–3167, 2010.
- [9] David Cohen-Tanugi and Jeffrey C Grossman. Water desalination across nanoporous graphene. *Nano letters*, 12(7):3602–3608, 2012.
- [10] Sumedh P Surwade, Sergei N Smirnov, Ivan V Vlassiouk, Raymond R Unocic, Gabriel M Veith, Sheng Dai, and Shannon M Mahurin. Water desalination using nanoporous single-layer graphene. *Nature nanotechnology*, 10(5):459–464, 2015.
- [11] Mohammad Heiranian, Amir Barati Farimani, and Narayana R Aluru. Water desalination with a single-layer mos 2 nanopore. *Nature communications*, 6(1):1–6, 2015.
- [12] Zhonglin Cao, Vincent Liu, and Amir Barati Farimani. Water desalination with two-dimensional metal–organic framework membranes. *Nano letters*, 19(12):8638–8643, 2019.
- [13] Zhonglin Cao, Vincent Liu, and Amir Barati Farimani. Why single-layer mos2 is a more energy efficient membrane for water desalination? *ACS Energy Letters*, 2020.
- [14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [15] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.
- [16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [18] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

- [19] Mohammadreza Karamad, Rishikesh Magar, Yuting Shi, Samira Siahrostami, Ian D Gates, and Amir Barati Farimani. Orbital graph convolutional neural network for material property prediction. *Physical Review Materials*, 4(9):093801, 2020.
- [20] Zhenpeng Yao, Benjamin Sanchez-Lengeling, N Scott Bobbitt, Benjamin J Bucior, Sai Govind Hari Kumar, Sean P Collins, Thomas Burns, Tom K Woo, Omar Farha, Randall Q Snurr, et al. Inverse design of nanoporous crystalline reticular materials with deep generative models. 2020.
- [21] Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.
- [22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [27] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.
- [28] FJ Richards. A flexible growth function for empirical use. *Journal of experimental Botany*, 10(2):290–301, 1959.
- [29] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.