

## ARTICLE OPEN



# Designing mechanically tough graphene oxide materials using deep reinforcement learning

Bowen Zheng<sup>1</sup>, Zeyu Zheng<sup>2</sup> and Grace X. Gu<sup>1</sup>✉

Graphene oxide (GO) is playing an increasing role in many technologies. However, it remains unanswered how to strategically distribute the functional groups to further enhance performance. We utilize deep reinforcement learning (RL) to design mechanically tough GOs. The design task is formulated as a sequential decision process, and policy-gradient RL models are employed to maximize the toughness of GO. Results show that our approach can stably generate functional group distributions with a toughness value over two standard deviations above the mean of random GOs. In addition, our RL approach reaches optimized functional group distributions within only 5000 rollouts, while the simplest design task has  $2 \times 10^{11}$  possibilities. Finally, we show that our approach is scalable in terms of the functional group density and the GO size. The present research showcases the impact of functional group distribution on GO properties, and illustrates the effectiveness and data efficiency of the deep RL approach.

*npj Computational Materials* (2022)8:225; <https://doi.org/10.1038/s41524-022-00919-z>

## INTRODUCTION

Graphene, a monolayer carbon allotrope, has been regarded as a cornerstone in materials science research ever since its discovery<sup>1</sup>. As such, there are several research directions related to graphene in both computational and experimental works in science and engineering applications<sup>2–5</sup>. Graphene oxide (GO), one of the best-known graphene derivatives, inherits many unique and exquisite properties of graphene and is playing an increasingly important role in various research areas such as electronics, energy storage, and biomedical applications<sup>6–12</sup>. Structurally, GO comprises a graphene basal plane (GBP) and a variety of oxygen-containing functional groups including hydroxyl (C–OH), epoxide (C–O–C), carbonyl (C=O), and carboxyl (O=C–OH) groups. Among these functional groups, hydroxyl and epoxide groups are dominant in number and distribute on the face of the GBP, while carbonyl and carboxyl groups are outnumbered and are only attached to the edges of the GBP<sup>13</sup>. As a result, the total amount and the relative ratio of hydroxyl and epoxide groups dictate the chemical composition, which plays a central role in influencing the mechanical properties of GO<sup>14</sup>. Contrarily, carbonyl and carboxyl groups are shown to be insignificant in affecting the chemistry and mechanical properties of GO.

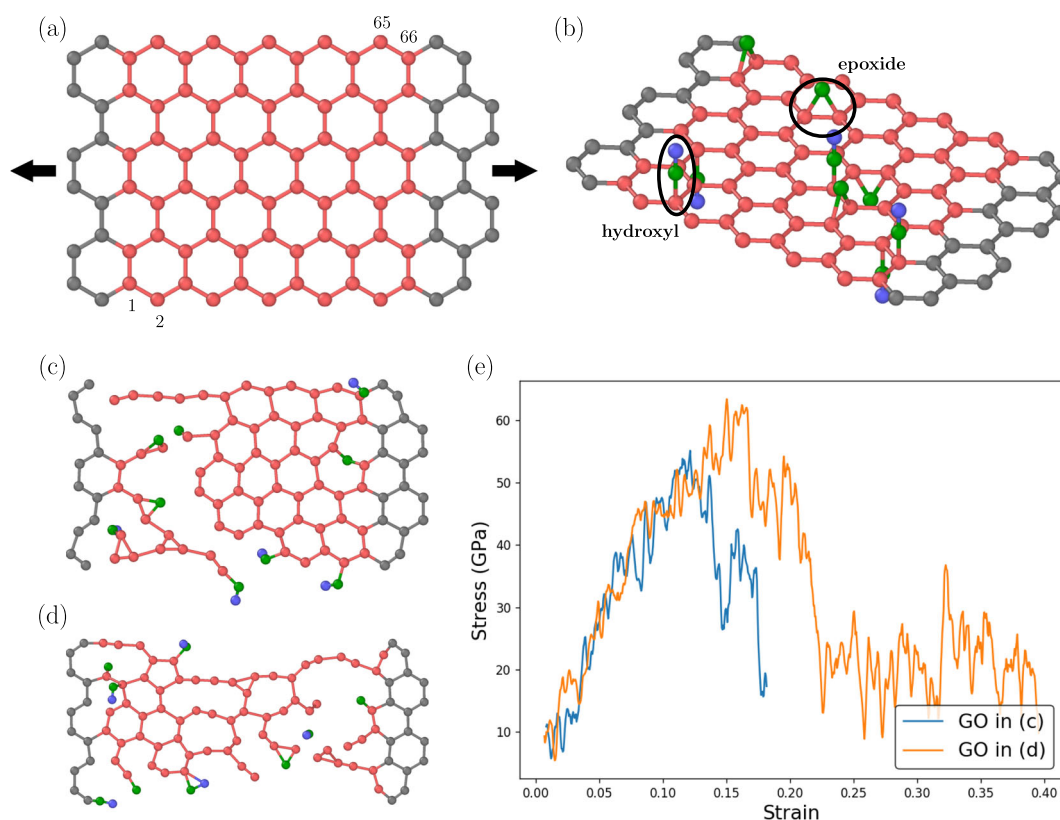
However, the mechanical property of a GO cannot be accurately inferred from only its chemical composition. Given one specific chemical composition, there can exhibit a range of GO mechanical properties, due to the variability in the functional group spatial distribution on the GBP. Research has shown that the functional group distribution can impact GO properties such as plasticity and ductility due to the mechanochemical interactions between functional groups<sup>15</sup>. One mechanical property of interest is toughness, defined as the amount of energy per unit volume that a material can absorb before rupturing. It quantifies the ability of a material to absorb energy and plastically deform without fracturing, thus requiring a balance of strength and ductility. GOs with high toughness are much desired, which can potentially

enhance the performances of many GO-based applications such as nanocomposites, flexible electronics, among others.

Given a specific chemical composition such as the oxygen-to-carbon ratio and the relative concentrations of functional groups, our goal is to maximize the toughness of GO by altering only the functional group spatial distribution. The existing literature has not sufficiently addressed this problem, and presumes that the effect of functional group distribution is secondary. From the perspective of optimization, it is a challenging task and has the following difficulties. First, optimizing over functional group distribution is in essence a combinatorial optimization problem, which can be NP-hard and analytically intractable, especially when the problem dimension is large. Second, the problem involves complex functional group interactions that evolve over time. There is little intuition about where to place functional groups at the beginning such that the GO will benefit in the long run. Third, both GO simulations and experiments can be expensive. Hence, an effective, data-efficient optimization strategy is highly valued.

Recently, machine learning algorithms have been applied successfully to materials prediction, design, and optimization problems<sup>16–24</sup>. Reinforcement learning (RL), a mathematical formalism for learning-based decision making, describes an approach where an agent performs sequential actions based on interactions with an environment so as to yield the most cumulative rewards<sup>25</sup>. When integrated with deep neural networks and advanced computing, the capability of RL is greatly amplified: Deep neural networks can process high-dimensional input, while RL can choose complex actions. Deep RL applications are numerous. One of the most famous examples is the achievement of superhuman performance in the game Go<sup>26,27</sup>, which was once considered an insurmountable task given the complexity of more than  $10^{140}$  possible solutions. In the context of materials science, deep RL has been gaining ground in molecule discovery and microstructure design<sup>28–33</sup>. More to our interest, deep RL also has an advantage in solving difficult combinatorial optimization problems. For these problems, many traditional

<sup>1</sup>Department of Mechanical Engineering, University of California, Berkeley, CA 94720, USA. <sup>2</sup>Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, USA. ✉email: [ggu@berkeley.edu](mailto:ggu@berkeley.edu)



**Fig. 1** GO schematics and mechanical responses. **a** Schematic of GBP, where red atoms (66 in total) are hosts for functional groups while gray atoms are functional group-free atoms on which the tensile loading is exerted. Arrows show the loading direction. **b** Illustrations of hydroxyl and epoxide groups, where green and blue atoms are oxygen and hydrogen atoms, respectively. **c** Fracture of a low-toughness GO under tension. **d** Fracture of a high-toughness GO under tension. **e** Stress-strain curves of GOs in **c** and **d**.

algorithms involve using hand-crafted heuristics that sequentially construct a solution. Nevertheless, the design of such heuristics can be a daunting task that requires domain expertise, and can often be suboptimal due to the difficult combinatorial nature of the problems. Therefore, the idea to infer heuristics without human intervention is enticing. Deep RL has shown promise to learn efficient heuristics to tackle these problems, and has been used to solve combinatorial optimization problems such as the Traveling Salesman Problem<sup>34–36</sup>, the Maximum Cut Problem<sup>37–39</sup>, and the Bin Packing Problem<sup>40–42</sup>.

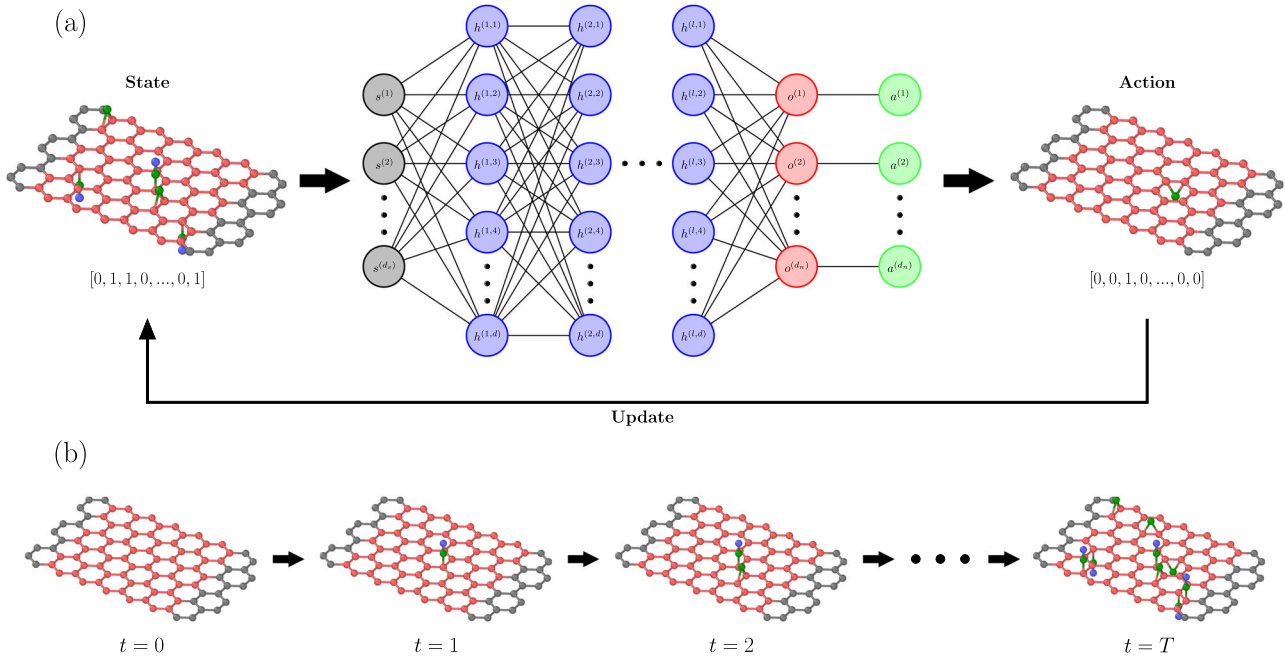
In this study, a deep RL framework is developed to design mechanically tough GOs by optimizing over the functional group distribution. In our deep RL framework, the task of functional group assignment is formulated as a sequential (Markov) decision process, where the state is the current functional group distribution on the GBP and the action is to assign a new functional group. A policy-gradient RL model is employed to maximize GO toughness, which is calculated by reactive molecular dynamics (MD) simulations. We design experiments of four difficulties to gradually challenge our deep RL model, and each difficulty consists of two experiments featuring two oxidation levels. We aim to develop a deep RL model with the following characteristics: (1) stable generation of mechanically tough GO configurations; (2) good scalability in terms of functional group density and GBP size; (3) tractable computation given the large design space.

## RESULTS

### Graphene oxide simulations

In the present study, a majority of GOs are based on GBPs that comprise a total of 94 carbon atoms, where 28 functional group-

free atoms near two opposite edges are clamped to enforce displacement, and 66 free-to-move atoms in the middle are active hosts for functional groups (referred to as the host atoms hereafter, and the number of these atoms are denoted by  $n_c$ ), as shown in Fig. 1a. Later in more complex experiments a larger GBP that is roughly twice the size will be used. In our GO model only hydroxyl and epoxide groups are considered, and less important carbonyl and carboxyl groups on GBP edges are omitted. Figure 1b shows an example GO model and illustrates the molecular structures of hydroxyl and epoxide groups. Each hydroxyl group resides on only one carbon atom, while each epoxide group takes on two neighboring carbon atoms. This difference adds to the optimization difficulty when both functional groups are present on the GBP. In addition, these functional groups can be attached to either side of the GBP. For the loading condition, the GO sheet is subjected to uniaxial tensile loading with a constant loading speed in the zigzag direction of the GBP. The mechanical responses of GOs are computed by reactive MD simulations, and the implementation details are provided in Methods. Reactive MD simulations are favorable in modeling the failure of nanomaterials because they account for bond breaking and formation, which are of vital importance in the fracture behavior at the nanoscale. We have observed in simulations that given the same amount of hydroxyl and epoxide groups, different functional group distributions can result in substantially different stress-strain relations and failure behaviors. Examples are given in Fig. 1c, d. GOs in Fig. 1c, d have the exact same amount of hydroxyl and epoxide groups, but Fig. 1c shows a brittle rupture while Fig. 1d shows a more ductile failure that involves considerable new bond formation and configuration change. Figure 1e compares the stress-strain curves of the two GOs above. It is shown that GO in Fig. 1d is higher in both ultimate



**Fig. 2** Deep RL state transition and trajectory. **a** Illustration of deep RL policy and state transition. **b** An example full trajectory.

**Table 1.** Summary of deep RL components.

DRL component	Notation	Description
State	$\mathbf{s}_t$	Current functional group locations on the GBP at time step $t$
State space	$S$	All possible functional group locations on the GBP
Action	$\mathbf{a}_t$	To assign a functional group to a functional group spot on the GBP
Action space	$\mathcal{A}(\mathbf{s}_t)$	All available functional group spots left given $\mathbf{s}_t$
Reward	$r_t$	Standardized toughness if at terminal step; otherwise, 0

stress and failure strain, suggesting superior mechanical properties. The toughness of a material can be expressed as  $u = \int_0^{\epsilon_f} \sigma d\epsilon$ , where  $u$  is toughness;  $\epsilon$  is strain;  $\epsilon_f$  is the strain upon failure;  $\sigma$  is stress. By the definition above, the toughness equals the area under the stress-strain curve. It is calculated that the toughness of GO in Fig. 1d is 2.1 times that of GO in Fig. 1c. This amount of difference in toughness suggests that the functional group distribution potentially has a profound impact on mechanical properties, and that it is worthwhile to optimize GO mechanical properties over functional group distribution. The two GO configurations also give rise to different out-of-plane deformation, due to different functional group distributions, of which the result is provided in the Supplementary Information.

### Deep reinforcement learning

The optimization problem we aim to solve is given a fixed number of hydroxyl and epoxide groups, how we can distribute these functional groups on the GBP so as to maximize the toughness of GO. Instead of treating the optimization problem as choosing the best functional group distribution in one shot, we model the functional group assigning problem as a sequential decision process and use RL to solve it. More specifically, each individual functional group is assigned to a location on the GBP at each of a sequence of discrete time steps  $t = 0, 1, 2, \dots, T$ , where  $T$  equals the total number of functional groups. At each time step  $t$ , the RL agent receives the representation of the environment's state  $\mathbf{s}_t$ , which is the current functional group locations. In our setup, the state space

is a discrete set that incorporates all functional group possibilities associated with individual carbon atoms or carbon atom pairs on the GBP, and is not defined in the continuous Euclidean space. After receiving a state  $\mathbf{s}_t$ , the RL agent selects an action  $\mathbf{a}_t$ , which is to assign a functional group to a functional group spot on the GBP. This is done by a policy  $\pi_\theta$ , where  $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$  is the probability of selecting the action  $\mathbf{a}_t$  if the state is  $\mathbf{s}_t$  under the policy parameter  $\theta$ , i.e.,  $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t) = \mathbb{P}(\mathbf{a}_t|\mathbf{s}_t; \theta)$ . After taking an action  $\mathbf{a}_t$  at state  $\mathbf{s}_t$ , the agent enters a new state  $\mathbf{s}_{t+1}$ , and this process is called a state transition. The state transition process involving policy network and action is illustrated in Fig. 2a. A trajectory is formulated as  $T = \{\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots, \mathbf{s}_{T-1}, \mathbf{a}_{T-1}, \mathbf{s}_T\}$ , and GO configurations throughout a whole example trajectory are shown in Fig. 2b. Upon entering a new state  $\mathbf{s}_{t+1}$ , the RL agent also receives a numerical reward  $r_{t+1} = r(\mathbf{s}_{t+1}) \in \mathbb{R}$ . We craft the reward as

$$r(\mathbf{s}_t) = \begin{cases} 0, & t < T \\ \hat{u}(\mathbf{s}_t), & t = T \end{cases} \quad (1)$$

where  $\hat{u}(\mathbf{s}_t)$  is standardized toughness given by

$$\hat{u}(\mathbf{s}_t) = \frac{u(\mathbf{s}_t) - \mu_u}{\sigma_u} \quad (2)$$

where  $\mu_u$  and  $\sigma_u$  are the mean and the standard deviation of random GOs. For each trajectory, the MD simulation is only run once at the final step when all functional groups have been assigned to obtain the only non-zero reward  $u(\mathbf{s}_T)$ . All RL components in this study are summarized in Table 1, and more details about the RL implementation are provided in Methods.

To conduct GO optimization using RL, we progressively build up experiment complexity and design four levels of difficulty: Easy, Medium, Hard, and Extra Hard. For Easy experiments, only hydroxyl groups are assigned to only one side of the GBP. For Medium experiments, only hydroxyl groups are assigned to the GBP, but they can be assigned to both sides of the GBP. Medium experiments are more complex than Easy experiments in that the state space and the action space are doubled in size. For Hard experiments, both hydroxyl and epoxide groups are assigned to the GBP, and they can be assigned to both sides of the GBP. The settings of Hard experiments resemble GOs in reality and involve competition between hydroxyl and epoxide groups. Extra Hard experiments are similar to Hard experiments but a larger GBP is used, consisting of 120 functional group hosts compared with 66 in all previous experiments. The descriptions of all experiment difficulties are summarized in Table 2. In addition, each difficulty consists of two oxidation levels: low and high, where the former has an oxygen-to-carbon ratio of around 15% while the latter doubles that. The Extra Hard difficulty is used to test the scalability in terms of the GO size, while the different oxidation levels are for the scalability with respect to the functional group density. In summary, we have 8 different experiments in total to challenge our deep RL algorithms, and the result of each experiment is evaluated based on 4 different random seeds. The numbers of hydroxyl and epoxide groups, and host atoms are summarized in Table 3. In all experiments, invalid actions can be simply stated as assigning a functional group to an already occupied carbon atom on the GBP. However, as the difficulty increases, the elimination of invalid actions becomes an increasingly delicate process, which is detailed in the Supplementary Information. To compute the reward formulated in Eq. (1), the mean  $\mu_u$  and  $\sigma_u$  of random GO configurations need to be calculated. The means and the standard deviations of 2000 random GOs in all experiments are summarized in Table 4, and the distribution histograms are provided in the Supplementary Information.

The algorithm also varies with experiments. For Easy and Medium experiments, only one policy network  $\pi_\theta$  is used to map the state to a probability distribution of all legal actions, i.e., assigning a hydroxyl group to an available spot. However, for Hard and Extra Hard experiments, two policy networks are needed to assign two types of functional groups. We denote the network for hydroxyl groups  $\pi_\theta^h$  and the network for epoxide groups  $\pi_\theta^e$ , where  $\theta$  and  $\rho$  are respective network parameters. Next, we need to decide on the sequence of assigning hydroxyl and epoxide groups. Because a non-zero reward is observed only at the

terminal step, only the network that assigns the last functional group will get its parameters updated via backpropagation. Therefore, the assignment sequence cannot be a deterministic one since we need to improve both networks. To this end, we use a Bernoulli distribution  $\text{Bernoulli}(m_h/(m_h+m_e))$  to sample the index of network used at each step, where  $m_h$  and  $m_e$  are the numbers of hydroxyl and epoxide groups left to assign at the current time step. This approach randomizes the sequence of functional group assignment in each episode and give both networks an opportunity to update parameters. The pseudo-codes of these two policy gradient algorithms are summarized in the Supplementary Information. Figure 3 shows the deep RL optimization results for all eight experiments. The numerical value of the return represents how many standard deviations the design is above the mean of random GOs with the same amount of functional groups (summarized in Table 4). It is shown in Fig. 3a-c that the final returns in the Easy, Medium and Hard experiments reach an average return of around 3, suggesting that the RL generated GO functional group distributions have a higher toughness than 99.87% of all GO configurations. In the Extra Hard experiments, our model achieves returns above 2, thus beating 97.73% of all GOs (Fig. 3d). In addition, all experiments reach a local maximum within only 5000 episodes (no more than 5000 MD simulations are run for each experiment), which is much smaller than the number of possible GO configurations. For the Easy difficulty, the low-oxidation and high-oxidation experiments have  $C_{10}(66) = 2.1 \times 10^{11}$  and  $C_{20}(66) = 4.1 \times 10^{16}$  possible functional group arrangements, and there are even much more arrangement possibilities for more difficult experiments. Last but not least, good performances in experiments of different oxidation levels and the Extra Hard experiments suggest that our RL design approach is scalable in terms of the functional group density and the GO size. The policy network architectures/parameters for all experiments are presented in Methods. The distribution histograms of our RL design within first 2000 episodes (to compare with the distribution of 2000 random GOs) and full 5000 episodes are provided in the Supplementary Information.

Finally, to gain insights from the perspective of microstructure, we compare the molecular structure and detailed failure behavior between a random GO and an RL-designed GO. The two GO examples are drawn from the Hard, high-oxidation experiment, and the comparison between the two GOs under different strains is shown in Fig. 4. From the initial configurations, we observe that the functional group distribution designed by RL tends to be more spread out than the random GO. Nevertheless, there is little intuition

**Table 2.** Experiment difficulty descriptions.

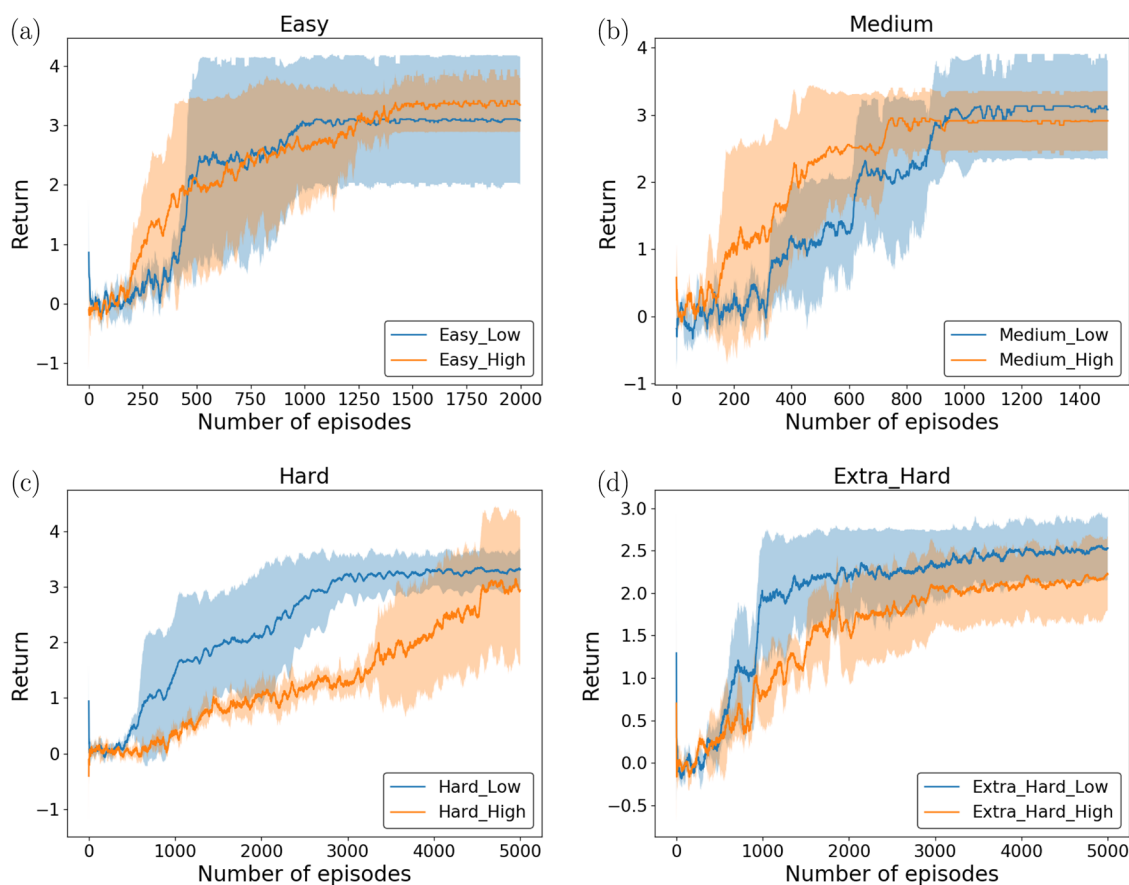
Difficulty	Description
Easy	Assigning only hydroxyl groups on only one side of the GBP
Medium	Assigning only hydroxyl groups on both sides of the GBP
Hard	Assigning hydroxyl and epoxide groups on both sides of the GBP
Extra Hard	Assigning hydroxyl and epoxide groups on both sides of a larger GBP

**Table 3.** Summary of the number of hydroxyl groups  $n_h$ , the number of epoxide groups  $n_e$ , and the number of carbon atoms that are hosts for functional groups  $n_c$  for all experiments.

	Easy			Medium			Hard			Extra Hard		
	$n_h$	$n_e$	$n_c$	$n_h$	$n_e$	$n_c$	$n_h$	$n_e$	$n_c$	$n_h$	$n_e$	$n_c$
low oxidation	10	0	66	10	0	66	5	4	66	9	8	120
high oxidation	20	0	66	20	0	66	10	8	66	18	16	120

**Table 4.** Summary of statistics of random GOs in all experiments (unit: GPa).

	Easy		Medium		Hard		Extra Hard	
	$\mu_u$	$\sigma_u$	$\mu_u$	$\sigma_u$	$\mu_u$	$\sigma_u$	$\mu_u$	$\sigma_u$
low oxidation	7.814	1.299	7.668	1.259	7.120	1.326	8.108	1.340
high oxidation	7.694	1.368	7.245	1.355	6.236	1.249	7.115	1.310



**Fig. 3** Deep RL performances on various experiments. **a** Easy, **(b)** Medium, **c** Hard, and **d** Extra Hard experiments.

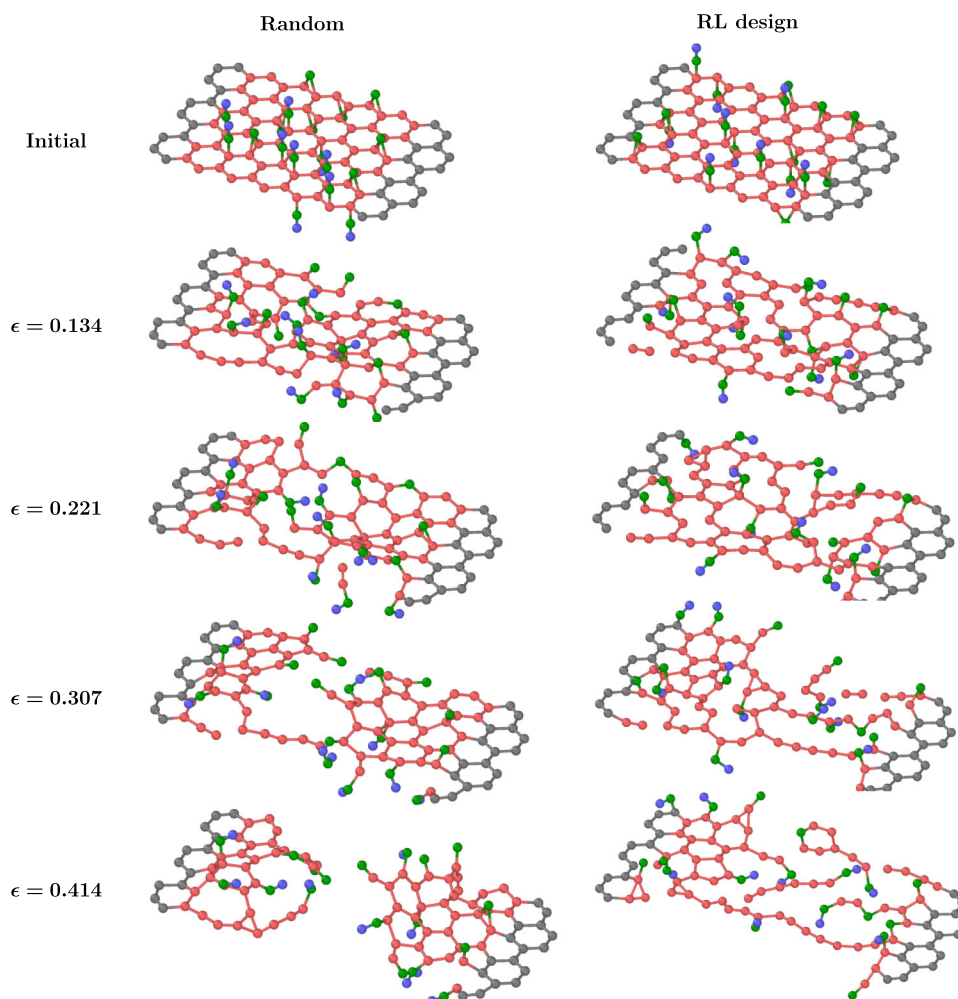
regarding how to design the specific functional group arrangement to achieve a high toughness. As the strain increases, the random GO fractures along a clearly defined path, while the fracture of RL designed GO initiates from multiple spots and forms a network-like structure that involves substantial new bond formation. This phenomenon suggests that the RL designed GO has more atoms contributing to energy absorption, which ultimately leads to a higher toughness (11.88 GPa versus 4.96 GPa for the random GO). To gain more physical insights as for what makes a GO tough, we conducted analysis to locate functional group sites that are more frequently occupied for high-toughness GOs. Concretely, we calculate the count of functional group appearances on every possible functional group site for high-toughness GOs for all levels of difficulty, and the results are provided in the Supplementary Information. It is shown that functional groups on high-toughness GOs seem to more likely distribute near the edges and not at the center. We interpret the observation as the following: functional groups generally have a negative effect on the GO toughness, and distributing them away from the center can help alleviate this effect. This is supported by the toughness results of random GO configurations, where high oxidation always has a lower mean toughness than low oxidation. However, it is shown that highly occupied functional group sites are not located only near the edges, and some sites inside the GBP also have a high occupancy. This may be explained by the involvement of other more complex mechanisms such as the interaction between functional groups, which emphasizes the necessity of using our RL-based design approach to solve this challenging problem.

## DISCUSSION

Our RL framework is on-policy, where the RL agent needs to sample a new trajectory for each episode. In our problem setting,

an MD simulation will be called to run at the terminal step of each trajectory to generate the reward according to Eq. (1), and this is where the most computation is spent. Future work includes developing a surrogate model that takes the state as the input, and outputs the reward to alleviate the computation of MD simulations during RL rollouts. Another issue arises from the present double policy network design. During each episode, only one network can get improved while the other network remains unchanged, which is not particularly a data-efficient algorithm design. In addition, when the numbers of two types of functional groups are imbalanced, the policy network of the minority functional group type may update very slowly. Future work includes designing a better policy network architecture to resolve or mitigate the two issues above.

For heterogeneous or disordered nanoscale systems, the arrangement of defects or functional groups has a major impact on the material properties when the system is small. However, as the system size increases, the effect of individual defect of functional group becomes smaller. It is expected that the potential of optimizing over functional group locations will become less significant. In the future work, we will further investigate the effect of functional group location on the mechanical properties of GOs as a function of the system size, as well as its RL-based optimization capability. Another limitation of the present study is that we have not taken into account the thermodynamics of designed GOs, meaning the output GO configurations may not be thermally stable. We would like to make a note that our deep RL design approach can still be of value in the following ways. First, the approach can be used as an effective layer of materials screening. For example, we can output 100 Deep RL designs and then apply a thermodynamic criterion to select both mechanically superior and thermally stable candidates. Second, we can modify



**Fig. 4 Rupturing comparison between a random GO and an RL-designed GO.** Examples are from the Hard, high-oxidation experiment.

the reward in the RL algorithm that favors thermally stable graphene oxides. Concretely, we can include the binding energy per oxygen in the reward, which can be expressed as

$$E_B = \frac{E_{GO} - E_g - n_h E_h}{n_e + n_h} \quad (3)$$

where  $E_{GO}$ ,  $E_g$ , and  $E_h$  are the total energies of the GO structure, pure graphene, and hydroxyl groups, respectively. We will investigate this approach in our future work. Third, our RL approach can be used to efficiently establish an upper bound for GO mechanical properties, given the chemical composition and the size.

In summary, a deep RL framework is developed to design GOs with high toughness by optimizing over the functional group distribution. The design task is formulated as a sequential decision process, where the state is the current functional group distribution on the GBP and the action is to assign a new functional group. A policy-gradient RL model is employed to maximize the toughness of GO, which is calculated by reactive molecular dynamics simulations. Eight experiments with increasing difficulty are devised to gradually challenge our deep RL model. We show that in the first six experiments our model can stably generate functional group distributions that achieve a toughness three standard deviations above the mean of random GOs, suggesting that the RL generated GOs have a higher toughness than 98.87% of all GOs. In the final two most difficult experiments, our model achieves two standard deviations above the mean of random GOs, thus beating 97.73% of all GOs. In

addition, our RL approach reaches an optimized functional group distribution within only 5,000 rollouts, while the easiest experiment has  $C_{10}(66) = 2.1 \times 10^{11}$  possibilities. Finally, we show that our RL design approach is scalable in terms of the functional group density and the GO size. The present research showcases the impact of functional group distribution on GO properties, and illustrates the effectiveness and data efficiency of deep RL in optimizing it.

## METHODS

### Deep RL setup

At each time step  $t$ , the RL agent receives the representation of the environment's state  $\mathbf{s}_t \in \mathcal{S}$ , where  $\mathcal{S}$  is the state space that comprises all possible states. In our case,  $\mathbf{s}_t$  is the current functional group locations at time step  $t$ , and  $\mathcal{S}$  denotes the set of all possible functional group locations on the GBP. We construct  $\mathbf{s}_t$  as a one-hot encoded vector, of which the dimension equals the number of all possible spots for functional groups on the GBP. Using the Easy experiment as an example, the dimension of  $\mathbf{s}_t$  is 66 since there are 66 spots in total for hydroxyl groups. If we use both hydroxyl and epoxide groups, the dimension of  $\mathbf{s}_t$  will increase to account for all possible spots for epoxide groups. If the  $i$ th spot has been assigned a functional group, the value of the  $i$ th entry of  $\mathbf{s}_t$ ,  $\mathbf{s}_t[i]$ , is 1; otherwise,  $\mathbf{s}_t[i]$  is 0. The number of 1's in  $\mathbf{s}_t$  equals the number of functional groups that have already been assigned at time step  $t$ . After receiving a state  $\mathbf{s}_t$ , the RL agent selects an action  $\mathbf{a}_t \in \mathcal{A}(\mathbf{s}_t)$ , where  $\mathcal{A}(\mathbf{s}_t)$  is

the set of legal actions given state  $\mathbf{s}_t$ . In our case,  $\mathbf{a}_t$  is to assign a functional group to a functional group spot on the GBP, and  $\mathcal{A}(\mathbf{s}_t)$  is the set of all available functional group spots left given  $\mathbf{s}_t$ .  $\mathbf{a}_t$  is also a one-hot encoded vector, of which the dimension equals the number of possible spots for assigning a specific type of functional group. In our RL framework,  $\mathbf{a}_t$  is different from  $\mathbf{s}_t$  in that  $\mathbf{a}_t$  only accounts for one specific type of functional group (either hydroxyl or epoxide group) while  $\mathbf{s}_t$  accounts for both types. If the action is to assign a functional group to  $i$ th spot among all possible spots,  $\mathbf{a}_t[i]=1$ . Because each action is restricted to the assignment of one functional group, there will only be one 1 entry in  $\mathbf{a}_t$ . In the present work we use neural networks to model the policy  $\pi_\theta$ , where  $\theta$  is the neural network parameters including weights and biases. In addition,  $\mathbf{a}_t$  is strictly enforced by the hybridization condition of host atoms on the GBP, which requires that one host atom can be only associated with one functional group. Therefore, after each functional group assignment, one or more actions will become infeasible for the next time step, and the possibilities of selecting these actions will be set to zero. The elimination process of invalid actions depends on the nature of the experiment in which the RL is implemented, and is detailed in the Supplementary Information. After taking an action  $\mathbf{a}_t$  at state  $\mathbf{s}_t$ , the agent enters a new state  $\mathbf{s}_{t+1}$ . In our context, after assigning a functional group to the current GO, we obtain a new GO. Details of how the state and the action are obtained at each time step are summarized in Algorithms 1 and 2 in the Supplementary Information. State transition function  $f(\mathbf{s}, \mathbf{a}, \xi)$  defines the successor state after selecting action  $\mathbf{a}$  in a state  $\mathbf{s}$  and random input  $\xi$ . In the present research the state transitions are deterministic,  $f(\mathbf{s}, \mathbf{a}, \xi) = f(\mathbf{s}, \mathbf{a})$ . Notably, our states have the Markov property, where the future states depend only upon the current state, not on the past states, i.e.,  $\mathbb{P}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{s}_{t-1}, \dots, \mathbf{s}_1, \mathbf{s}_0) = \mathbb{P}(\mathbf{s}_{t+1}|\mathbf{s}_t)$ . The functional group positions on the GBP serves as a Markov state which summarizes the functional group assignment history that has led up to it.

The goal of RL is to maximize the expected return, where the return is a function of the reward sequence. However, based on the reward setting in Eq. (1), the agent will only receive a non-zero reward at the terminal step. This is inspired by the AlphaGo research where the agent only receives a non-zero reward when the game ends:  $r=1$  if the agent wins the game;  $r=-1$  if the agent loses the game. In this study, policy gradient algorithms are used to maximize the expected return, which directly optimizes a parametrized policy via gradient descent. Concretely, for a policy  $\pi_\theta(\mathbf{a}|\mathbf{s})$  parametrized by  $\theta$ , the change of parameter after each episode (sampling a full trajectory  $\mathcal{T}$ ) is given by<sup>43</sup>

$$\Delta\theta = \frac{\partial}{\partial\theta} \left( \sum_{t=1}^T \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \right) \mathbb{E}_{\tau \sim \mathbb{P}_\theta(\tau)} \left[ \sum_{t=1}^T r(\mathbf{s}_t) \right] \quad (4)$$

Using the Monte Carlo sampling we have

$$\mathbb{E}_{\tau \sim \mathbb{P}_\theta(\tau)} \left[ \sum_{t=1}^T r(\mathbf{s}_t) \right] \approx \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T r(\mathbf{s}_t) \quad (5)$$

In this study, we update the policy network parameter once every trajectory, therefore  $N=1$ . We arrive at

$$\Delta\theta = \frac{\partial}{\partial\theta} \left( \sum_{t=1}^T \log \pi_\theta(\mathbf{a}_t|\mathbf{s}_t) \right) \left( \sum_{t=1}^T r(\mathbf{s}_t) \right) \quad (6)$$

The parameter update follows  $\theta \leftarrow \theta + a\Delta\theta$ , where  $a$  is the current learning rate.

### Neural networks

For all experiments, we use fully connected neural networks of various sizes and ReLU activations. At the last layer we use a fully connected layer followed by a softmax activation which outputs

the probability distributions of actions, as a way to address the exploration-versus-exploitation dilemma. The input and output dimensions in different experiments are summarized in Table 5.

We next set zero the probabilities of selecting invalid actions and re-normalize the distribution such that the sum of the probabilities of all legal actions at each time step equals 1. An Adam optimizer is used. Learning rate shrinks by a factor of two every 500 iterations, but is set no smaller than  $5e-5$ . The sizes and initial learning rates used in all experiments are summarized in Table 6. Weights and biases are initialized from  $\mathcal{U}(-1/\sqrt{d_{in}}, 1/\sqrt{d_{in}})$ , where  $\mathcal{U}$  denotes a uniform distribution, and  $d_{in}$  denotes the dimensionality of the input for each layer.

### Molecular dynamics simulations

Molecular dynamics simulations are performed using the open-source code LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator)<sup>44</sup>. ReaxFF potential, a reactive force field, is adopted to model the interactions among carbon, hydrogen and oxygen atoms in GOs<sup>45</sup>. ReaxFF potential models both non-bonded interactions such as van der Waals and Coulomb interactions, and bond breaking and formation. Specifically, potential parameters developed in Ref. <sup>45</sup> are used, which have been proved reliable by various studies on the physical and chemical behavior of graphene systems<sup>46–49</sup>. A three-dimensional, full atomistic model is used. Periodic boundary conditions are applied in all three spatial dimensions. The size of the simulation box is initialized at  $58.5 \text{ \AA}$  by  $21.4 \text{ \AA}$  by  $15.2 \text{ \AA}$  for Easy, Medium, and Hard experiments, and is initialized at  $63.3 \text{ \AA}$  by  $25.6 \text{ \AA}$  by  $15.2 \text{ \AA}$  for Extra Hard experiments. The equations of motion are integrated with a timestep of  $0.1 \text{ fs}$  ( $0.1 \times 10^{-15} \text{ s}$ ) using the Verlet algorithm, which ensures the computational stability. The trajectories, velocities, forces, and energies of all atoms are recorded every 10 timesteps. To simulate tensile loading at room temperature, an ensemble of random velocity corresponding to the temperature  $300 \text{ K}$  is firstly generated throughout all atoms. Then an equilibrium is realized by running a simulation in the isothermal-isobaric (NPT) ensemble with a Nose–Hoover thermostat<sup>50</sup> at the same temperature for 5000 timesteps. The loading scenario is simulated in the canonical (NVT) ensemble at  $300 \text{ K}$ . The unidirectional in-plane tensile load is applied along the zigzag direction based on a deformation-control manner until failure. The loading speed is  $1000 \text{ m}\cdot\text{s}^{-1}$ . During the NPT simulation the box size changes but very minimally, and during the NVT simulation the box size does not change. The box size as a

**Table 5.** Neural network input and output dimensions for all experiment difficulties.

	Easy	Medium	Hard	Extra Hard
input ( $\mathbf{s}_t$ )	66	132	308	570
output ( $\pi(\cdot \mathbf{s}_t)$ )	66	132	132; 176	240; 330

For Hard and Extra Hard outputs, the first number refers to the hydroxyl network and the second number refers to the epoxide network.

**Table 6.** Neural network parameters for all experiments.

	Easy		Medium		Hard		Extra Hard	
	low	high	low	high	low	high	low	high
hidden layer number	2	2	2	2	2	2	2	2
hidden layer size	200	400	300	600	600	800	800	800
initial learning rate	1e-3	1e-3	1e-3	1e-3	1e-3	5e-4	1e-3	1e-3

function of time in the Fig. 1c, d examples are provided in the Supplementary Information.

The components of stress tensor  $[\sigma]$  are calculated by the following:

$$\sigma_{ab} = \frac{1}{\Omega} \left( \sum_i \sum_{j>i} (x_a^{(i)} - x_a^{(j)}) \frac{\partial V}{\partial (x_b^{(i)} - x_b^{(j)})} - \sum_i m \dot{x}_a^{(i)} \dot{x}_b^{(i)} \right) \quad (7)$$

where  $a$  and  $b$  take on spatial dimensions 1 (zigzag), 2 (armchair), or 3 (out-of-plane) to generate the 6 independent components of the symmetric tensor;  $\Omega = At_e$  is the system volume;  $A$  is the area of the GBP;  $t_e$  is the effective thickness of graphene oxide;  $i$  and  $j$  are atom indices;  $x$  denotes the displacement;  $\dot{x}$  denotes the derivative of  $x$  with respect to time;  $V$  is the interatomic potential model. In this study,  $t_e = 7.76 \text{ \AA}$  is used, the interlayer spacing of GOs measured in experiments<sup>51,52</sup>.

## DATA AVAILABILITY

All data used in this work can be generated using the codes at [https://github.com/BOWENmeZHENG/go\\_rl](https://github.com/BOWENmeZHENG/go_rl).

## CODE AVAILABILITY

All codes in this work are available at [https://github.com/BOWENmeZHENG/go\\_rl](https://github.com/BOWENmeZHENG/go_rl).

Received: 23 April 2022; Accepted: 20 October 2022;

Published online: 05 November 2022

## REFERENCES

- Novoselov, K. S. et al. Electric field effect in atomically thin carbon films. *Science* **306**, 666–669 (2004).
- Huang, M., Pascal, T. A., Kim, H., Goddard, W. A. & Greer, J. R. Electronic–mechanical coupling in graphene from in situ nanoindentation experiments and multiscale atomistic simulations. *Nano Lett.* **11**, 1241–1246 (2011).
- Zheng, B. & Gu, G. X. Tuning the graphene mechanical anisotropy via defect engineering. *Carbon* **155**, 697–705 (2019).
- Craciun, M. F., Russo, S., Yamamoto, M. & Tarucha, S. Tuneable electronic properties in graphene. *Nano Today* **6**, 42–60 (2011).
- Zheng, B., Zheng, Z. & Gu, G. X. Scalable graphene defect prediction using transferable learning. *Nanomaterials* **11**, 2341 (2021).
- Wei, Z. et al. Nanoscale tunable reduction of graphene oxide for graphene electronics. *Science* **328**, 1373–1376 (2010).
- Wu, X. et al. Epitaxial-graphene/graphene-oxide junction: an essential step towards epitaxial graphene electronics. *Phys. Rev. Lett.* **101**, 026801 (2008).
- Chung, C. et al. Biomedical applications of graphene and graphene oxide. *Acc. Chem. Res.* **46**, 2211–2224 (2013).
- Lee, J., Kim, J., Kim, S. & Min, D.-H. Biosensors based on graphene oxide and its biomedical application. *Adv. Drug Deliv. Rev.* **105**, 275–287 (2016).
- Bo, Z. et al. Green preparation of reduced graphene oxide for sensing and energy storage applications. *Sci. Rep.* **4**, 4684 (2014).
- Xu, J., Wang, K., Zu, S.-Z., Han, B.-H. & Wei, Z. Hierarchical nanocomposites of polyaniline nanowire arrays on graphene oxide sheets with synergistic effect for energy storage. *ACS Nano* **4**, 5019–5026 (2010).
- Chen, C.-T., Martin-Martinez, F. J., Ling, S., Qin, Z. & Buehler, M. J. Nacre-inspired design of graphene oxide–polydopamine nanocomposites for enhanced mechanical properties and multi-functionalities. *Nano Futures* **1**, 011003 (2017).
- Johari, P. & Shenoy, V. B. Modulating optical properties of graphene oxide: role of prominent functional groups. *ACS Nano* **5**, 7640–7647 (2011).
- Zheng, B. & Gu, G. X. Prediction of graphene oxide functionalization using gradient boosting: implications for material chemical composition identification. *ACS Appl. Nano Mater.* **4**, 3167–3174 (2021).
- Wei, X. et al. Plasticity and ductility in graphene oxide through a mechanochemically induced damage tolerance mechanism. *Nat. Commun.* **6**, 8029 (2015).
- Jin, Z., Zhang, Z., Demir, K. & Gu, G. X. Machine learning for advanced additive manufacturing. *Matter* **3**, 1541–1556 (2020).
- Theodoridis, S. *Machine learning: a Bayesian and optimization perspective* (Academic press, 2015).
- Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
- Karniadakis, G. E. et al. Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
- Lee, S., Zhang, Z. & Gu, G. X. Generative machine learning algorithm for lattice structures with superior mechanical properties. *Mater. Horiz.* **9**, 952–960 (2022).
- Chen, C.-T. & Gu, G. X. Learning hidden elasticity with deep neural networks. *Proc. Natl Acad. Sci. USA* **118**, e2102721118 (2021).
- Li, S., Bai, H., Shepherd, R. F. & Zhao, H. Bio-inspired design and additive manufacturing of soft materials, machines, robots, and haptic interfaces. *Angew. Chem. Int. Ed.* **58**, 11182–11204 (2019).
- Gu, G. X., Chen, C.-T., Richmond, D. J. & Buehler, M. J. Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment. *Mater. Horiz.* **5**, 939–945 (2018).
- Zheng, B. & Gu, G. X. Machine learning-based detection of graphene defects with atomic precision. *Nano-Micro Lett.* **12**, 1–13 (2020).
- Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction*. 2nd edn. (MIT press, 2018).
- Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
- Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
- Popova, M., Isayev, O. & Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **4**, eaap7885 (2018).
- Putin, E. et al. Reinforced adversarial neural computer for de novo molecular design. *J. Chem. Inf. Model.* **58**, 1194–1204 (2018).
- Zhou, Z., Kearnes, S., Li, L., Zare, R. N. & Riley, P. Optimization of molecules via deep reinforcement learning. *Sci. Rep.* **9**, 10752 (2019).
- Sun, H. & Ma, L. Generative design by using exploration approaches of reinforcement learning in density-based structural topology optimization. *Designs* **4**, 10 (2020).
- Sui, F., Guo, R., Zhang, Z., Gu, G. X. & Lin, L. Deep reinforcement learning for digital materials design. *ACS Mater. Lett.* **3**, 1433–1439 (2021).
- Rajak, P. et al. Autonomous reinforcement learning agent for stretchable kirigami design of 2D materials. *npj Comput. Mater.* **7**, 102 (2021).
- Bello, I., Pham, H., Le, Q. V., Norouzi, M. & Bengio, S. Neural combinatorial optimization with reinforcement learning. *Preprint at <https://doi.org/10.48550/arXiv.1611.09940>* (2016).
- Khalil, E., Dai, H., Zhang, Y., Dilkina, B. & Song, L. Learning combinatorial optimization algorithms over graphs. *Adv. Neural Inf. Process. Syst.*, <https://papers.nips.cc/paper/2017/file/d9896106ca9896198d9896103d9896105b9896108cbdf9896104fd9896108b9896113a9896101-Paper.pdf> (2017).
- Nazari, M., Oroojlooy, A., Snyder, L. & Takáč, M. Reinforcement learning for solving the vehicle routing problem. *Adv. Neural Inf. Process. Syst.*, <https://proceedings.neurips.cc/paper/2018/file/2019fb4651c2005b2012ed2070fba2015afe2010b2039a2550-Paper.pdf> (2018).
- Barrett, T., Clements, W., Foerster, J. & Lvovsky, A. Exploratory combinatorial optimization with reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* **34**, 3243–3250 (2020).
- Tang, Y., Agrawal, S. & Faenza, Y. Reinforcement learning for integer programming: learning to cut. *Int. Conf. Mach. Learn.*, <https://doi.org/10.48550/arXiv.1906.04859> (2020).
- Cappart, Q., Goutier, E., Bergman, D. & Rousseau, L.-M. Improving optimization bounds using machine learning: decision diagrams meet deep reinforcement learning. *Proc. AAAI Conf. Artif. Intell.* **33**, 1443–1451 (2019).
- Hu, H., Zhang, X., Yan, X., Wang, L. & Xu, Y. Solving a New 3D bin Packing Problem with deep Reinforcement Learning Method. *arXiv <https://doi.org/10.48550/arXiv.1708.05930>* (2017).
- Duan, L. et al. A multi-task Selected Learning approach for solving 3D Flexible bin Packing Problem. *arXiv <https://doi.org/10.48550/arXiv.1804.06896>* (2018).
- Cai, Q. et al. Reinforcement Learning driven Heuristic Optimization. *arXiv <https://doi.org/10.48550/arXiv.1906.06639>* (2019).
- Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 229–256 (1992).
- Plimpton, S. Fast parallel algorithms for short-range molecular dynamics. *J. Comput. Phys.* **117**, 1–19 (1995).
- Chenoweth, K., Van Duin, A. C. & Goddard, W. A. ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation. *J. Phys. Chem. A* **112**, 1040–1053 (2008).
- Chen, Z., Khajeh, A., Martini, A. & Kim, S. H. Identifying physical and chemical contributions to friction: a comparative study of chemically inert and active graphene step edges. *ACS Appl. Mater. Interfaces* **12**, 30007–30015 (2020).



47. Vashisth, A. et al. ReaxFF simulations of laser-induced graphene (LIG) formation for multifunctional polymer nanocomposites. *ACS Appl. Nano Mater.* **3**, 1881–1890 (2020).
48. Berman, D., Deshmukh, S. A., Sankaranarayanan, S. K. R. S., Erdemir, A. & Sumant, A. V. Macroscale superlubricity enabled by graphene nanoscroll formation. *Science* **348**, 1118–1122 (2015).
49. Yoon, K., Ostadhosseini, A. & van Duin, A. C. T. Atomistic-scale simulations of the chemomechanical behavior of graphene under nanoparticle impact. *Carbon* **99**, 58–64 (2016).
50. Hoover, W. G. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A* **31**, 1695–1697 (1985).
51. Bai, H., Li, C., Wang, X. & Shi, G. A pH-sensitive graphene oxide composite hydrogel. *Chem. Commun.* **46**, 2376–2378 (2010).
52. Zhang, D., Tong, J., Xia, B. & Xue, Q. Ultrahigh performance humidity sensor based on layer-by-layer self-assembly of graphene oxide/polyelectrolyte nanocomposite film. *Sens. Actuators B Chem.* **203**, 263–270 (2014).

## ACKNOWLEDGEMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges system, which is supported by National Science Foundation (Fund number: ACI-1548562). The authors acknowledge support from the Alfred P. Sloan Foundation and the National Science Foundation (Fund Number: DMR-2119276).

## AUTHOR CONTRIBUTIONS

All authors contributed to the study conception and design, data analysis, and writing of the manuscript.

## COMPETING INTERESTS

The authors declare no competing interests.

## ADDITIONAL INFORMATION

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41524-022-00919-z>.

**Correspondence** and requests for materials should be addressed to Grace X. Gu.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

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



**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/>.

© The Author(s) 2022