Deep Reinforcement Learning for Digital Materials Design

Fanping Sui, Ruiqi Guo, Zhizhou Zhang, Grace X. Gu,* and Liwei Lin*

Cite This: ACS Materials Lett. 2021, 3, 1433−1439

ABSTRACT: Designing composites has been a research topic of interest in the field of materials science. As an elegant mathematical representation for composites, the concept of digital materials (DMs) was developed to express structures with complex geometries and various material distributions. DMs have a vast design space to achieve targeted physical properties, which makes it challenging for solving inverse problems. Here, a deep reinforcement learning (DRL) scheme is utilized to automate the DM design process without the designer’s prior knowledge. Based on the reward signal of structural mechanical property changes, DRL algorithms can initiate new design patterns in a self-updating process. As a demonstration example, a DM system composed of two different materials are selected as testing environments with three different levels of design space sizes. The collaborative deep Q network (DQN) architecture is developed to comprise two cooperative agents for two types of element-level modification operations to satisfy the design constraints, such as material fraction. The quality of each composite pattern is calculated through the finite element analysis (FEA) simulation. Results show the proposed approach can effectively handle the complex state-action space problems for the digital material design process with significant computation advantages, compared with those of the genetic algorithm with a 15.9% final design quality enhancement. As such, this new class of DRL scheme could be a powerful tool to enable the autonomous discovery process for next-generation free-form DM designs.

Additive manufacturing (AM), also known as three-dimensional (3D) printing, can fabricate structures with complex geometries and various material distributions. Commercial 3D printers build a structure in a layer-by-layer manner consisting of numerous material compositions in each layer. The high degree of freedom in the material placements inspires next-generation composite material designs with varying physical properties and free-form geometries. These composites are sometimes characterized as digital materials (DMs), utilizing the 3D printing process to assemble finite material voxels. Different from the regular geometries that can be described with only a few discrete parameters, DM design considers the structures as pixelated images. As the design space is enlarged by orders of magnitude in DM structures, the design screening and optimization process has become rather complex. A robust method that can automatically and efficiently generate optimal designs has become a key research topic in this field.

Recent advances in artificial intelligence have shown various machine learning (ML) applications in a wide range of science and engineering, such as the image recognition, controlling system, speech recognition, neuroscience, energy and environmental technologies. In the field of DM design, the ML applications have mostly been in the form of supervised learning. The supervised learning method can solve the forward problem: identifying the physical property of a given design configuration, such as the stiffness and toughness, thermal conductivity, vibration frequency, and material defects. Supervised learning models, especially the trained neural networks, can operate at orders of magnitude faster than the conventional commercial numerical simulators. While these attempts have undoubtedly led to pioneering results, there are still important applications that have yet to be fully exploited, especially when solving inverse problems: exploring the optimal design pattern to achieve the desired physical property. The solution space of complex inverse problems is nonconvex and there are many local optima; thus, it is
challenging to discover the global optimal solution to this type of one-to-many problem.

To solve for inverse problems in DM design without investigating the entire design space, researchers have proposed automated strategies such as the gradient-based topology optimization, and the generative networks (GN)-based schemes. However, these methods face limitations. For example, the genetic approach and topology-based method have a fast computation speed, but they are likely to yield an local optimum, which is largely dependent on the initial states. The GN-based method ameliorates the problem by large-scale random initialization but has lower optimization efficiency, because they require pretrainings with these data. Furthermore, most of the proposed samples are generated within its initial distribution regime, because of the high sample correlation issue during the optimization process.

Benefiting from big data and deep learning technologies, there is a renaissance of reinforcement learning in the recent years. Modern DRL methods have shown significant success in solving complex real-world problems, such as Google’s AlphaGo program, autonomous vehicles, precision robot arm controls, and molecular discoveries. The RL method can select the actions that maximize the future profit of a system by iteratively interacting with their operation environment. RL algorithms do not rely on prior knowledge nor large amount of initialization samples. In this work, we propose a novel control system for the DM design based on the deep Q network (DQN) structure and use finite element analysis (FEA) software as the simulation environment. The major challenge in this work is to convert the free-form DM design problem to a Markov decision process (MDP) and to find the correct action-state update strategies. In order to demonstrate our proposed DQN-based control system (with details outlined in the Methods section), a two-dimensional (2D) composite planar structure is selected for study by setting the initial design target as the optimal design, which can have the strongest average structure tensile strength along the primary axes. Several key features are accomplished from the proposed intelligent design system:

(1) Automation of the free-form DM design process via the DRL method with element-level modification operations.

(2) Significant final design quality enhancement and sample efficiency, compared with the conventional genetic algorithm method for complex design tasks.

(3) Capability of generating major population of the final optimal designs outside of the initial distribution, which indicates the high sample efficiency of the system.

## METHODS

### System Architecture

The architecture for the deep RL control system is shown in Figure 1. The composite sheet is constrained to have a symmetrical design about both axes to avoid any singular behavior during deformation, such that the size of the design domain is one-fourth of the entire composite sheet. It is assumed that the composite sheet contains two types of 3D printing materials with isotropic linear elastic properties. The red voxel (material 0) is stiff with a modulus $E_0 = 1136$ MPa, Poisson’s ratio $\nu_0 = 0.33$, and the blue voxel (material 1) corresponds to a soft material with a modulus $E_1 = 5.5$ MPa and Poisson’s ratio $\nu_1 = 0.37$ (properties adopted from refs 36 and 37). At the beginning, the design domain distributions are encoded into 0 and 1 and randomly initialized by fixing the number ratios of 0 to 1 during each experiment. The digitalized design domain and optimal design can be captured via the iterative sequential self-updating process that follows.

During each iteration, the previous observation matrix is flattened into a one-dimensional (1D) array representation of states: $[s_0, s_1, \ldots, s_m]$ which corresponds to the sequential readings of the original observation matrix at the $i$th iteration with $m$ elements. The previous states are passed to the DQN-based decision maker, which is a deep neural network that gives Q-values as the outputs. The Q-values of $[Q_0, Q_1, \ldots, Q_n]$ represent the expected total rewards (total design improvement) of taking the corresponding action with the action space of $n$. By searching the position of the maximum Q-value, the next actions for the new observation update can be identified and the new observation is prepared for the next round of update.

Another important part of the pipeline is to calculate the structure’s equivalent modulus for the reconstructed symmetric composite sheet from the new observation. The equivalent modulus along two symmetric axes and its average are defined as

$$E_{\text{horizontal}} = \frac{1}{ar_{\text{sheet}}} \int_{\Delta z_{\text{slice}}} \sigma_z$$

(1)
Figure 2. “Flipping” process of the collaborative DQN method, where the one soft voxel first flips to stiff voxel and then another stiff voxel flips to soft voxel.

\[
E_{\text{vertical}} = \frac{1}{a_{\text{sheet}}} \int \sigma_{yy} \, dA_{yy}
\]

\[
E_{\text{mean}} = E_{\text{horizontal}} + \frac{E_{\text{vertical}}}{2}
\]

where \( a \) represents the edge length of the composite sheet, \( \sigma_{xx} \) and \( \sigma_{yy} \) represent the horizontal and vertical tensile component of the Cauchy stress at the corresponding domain. Under the vertical loading scenario, the composite sheet is clamped at its bottom edge and the top edge has a vertical tensile displacement control with 2\% of the sheet edge length (\( a_{\text{sheet}} \)). The horizontal loading scenario has the same criterion where the left edge is clamped and the right edge is loaded. The material behavior simulation is realized by solving the governing equation for small elastic deformation:

\[
\nabla \cdot \sigma + f = 0
\]

\[
\sigma = \lambda \text{tr}(\varepsilon) I + 2\mu \varepsilon
\]

\[
\varepsilon = \frac{1}{2}(\nabla \mu + (\nabla \mu)^T)
\]

where \( \sigma \) is the stress tensor, \( f \) is the body force, \( I \) is the identity matrix, and \( \text{tr} \) is the trace function. The governing equations are then solved using the FEA method coded with the numerical python package FEniCS. Simulation results, observations, and actions are saved to the replay buffer, which serves as a database for trajectories. The updating process of the DQN algorithm is the core unit of our work and can be treated as a training process for the deep neural networks (DNNs), which can be used to estimate the complex, nonlinear Q-value function. To fix possible instabilities, DQN uses two heuristics, the target Q-network and the replay buffer to reduce the data correlation between different trajectories, which makes the deep RL less likely to be trapped in the local optima.

DQN Decision Maker. One requirement for the reinforcement learning algorithms is to formulate a valid MDP process. For our specific stepwise structure optimization process, two possible action update policies have been developed to be expressed with two pairs of DQN, namely collaborative DQN.

As demonstrated in Figure 2, this method expresses the “flipping process” with two consecutively updated Q networks, representing two cooperating RL agents: Q-net1 and Q-net2. These two networks have the same input and output size and \( n = m \). The Q-net1 serves the purpose of picking up one of the soft voxels and switching to the stiff voxel, while Q-net2 does the opposite. The updating logic can be expressed as follows:

\[
\phi_1 \leftarrow \phi_1 - \alpha \sum \frac{dQ_{\phi_1}(s_k, a_k)}{d\phi_1}(Q_{\phi_1}(s_k, a_k) - \gamma \max_{a'_{\text{st}}} Q_{\phi_2}(s_k', a_k'))
\]

\[
\phi_2 \leftarrow \phi_2 - \alpha \sum \frac{dQ_{\phi_2}(s_k', a_k')}{d\phi_2}(Q_{\phi_2}(s_k', a_k') - \gamma \max_{a_{\text{sf}}} Q_{\phi_1}(s_k'', a_k''))
\]

where \( \phi_1 \) or \( \phi_2 \) are the weight parameters for Q-net1 or Q-net2. \( \alpha \) is the learning rate, and \( \gamma \) is the discount factor. \((s_k, a_k)\) is the state action pair from sample batch and \((s_k', a_k')\) is the next state and potential action to give maximum Q value. The action-state pair \((s_k', a_k')\) is corresponding to the intermediate observation with one more stiff voxel and one fewer soft voxel. The \( a_k'' \) is also a possible action to take and \( s_k'' \) corresponds to the new state, which has the same stiff to soft voxel ratio as the observation of \( s_k \). The single-step reward \( r(s_k, a_k) \) is calculated as the change of the averaged equivalent modulus (\( E_{\text{mean}} \)) after taking the action. Here, Q-net1 and Q-net2 are updated from two independent replay buffers. However, this scheme can be further optimized by a shared replay buffer for more sampling efficiency. The corresponding pseudocode is shown in the Supporting Information. The deep reinforcement learning model is coded and trained on Pytorch.

Genetic Algorithms. The genetic algorithms have been widely applied in design discovery schemes and acknowledged for their easy implementation and high performance under low dimensional spaces. Briefly, genetic optimization is mimicking the natural selection and evolution process where the best one survives and breeds. In this work, the performance of the genetic algorithm is regarded as a baseline comparison for the deep reinforcement learning methods as discussed above. For the specific composite design, the algorithm starts from generating random design array with a size of \( N^{\text{ob}} \) (\( N = 30 \) for \( 3^3 \) and \( 5^4 \), \( N = 80 \) for \( 7^7 \)), which is then sorted according to their performance (\( E_{\text{mean}} \)) as computed by FEA. The top \( p \) candidates (\( p = 10 \) for \( 3^3 \) and \( 5^4 \), \( p = 20 \) for \( 7^7 \)) are selected as parent designs. A pair of offsprings are then generated from each pair of parent designs by random permutations on the unmatched indexes. The offsprings also have a small probability to mutate where certain entries are randomly permuted. The rest slots of the design array are then filled by new random designs. This entire process is looped for \( c \) cycles (\( c = 10 \) for \( 3^3 \), \( c = 300 \) for \( 5^4 \), \( c = 250 \) for \( 7^7 \)). We want to make a note here that the chosen genetic algorithm parameters (\( N, p, c \)) are by no means optimal. However, they
are tuned within a certain range of parameter configuration to show decent performance, for instance, $N = 80$ stands out from $[20, 40, 80, 120, 160]$ for the 7*7 design space.

## RESULTS AND DISCUSSION

The performances of the genetic algorithm and collaborative DQN are compared over three different sizes of design domain via the historical best returns, which records the sample amount necessary to give the historical highest performance. The specific sizes and material proportions for case study is listed in Table 1. For the collaborative DQN method, the evaluation average return and training loss are recorded to demonstrate the learning process. Over iterations, the evaluation average return represents the $E_{\text{mean}}$ output and the training loss gives the $Q$-value difference of neural network predictions and actual returns. The total learning curves for the three levels of tasks are separately drawn in Figures S1−S3 in the Supporting Information.

Figure 3a shows that, for the 3*3 domain settings, where the entire design space has 126 possibilities at most, both collaborative DQN and the genetic algorithm can easily find the global optimum patterns. For example, under the constraints of 7 stiff units and 2 soft units, the maximum $E_{\text{mean}}$ of the structure is 853.9 MPa. The genetic algorithm is the faster one to find the global optimum, which only needs 70 samples to get the maximum design. The collaborative DQN discovers the best pattern with 400 samples, which is less efficient than the genetic algorithm. Therefore, under small design spaces with limited combinations, the DRL method does not have advantage over genetic algorithms. This phenomenon can be also concluded from other learning curves in Figure S1 in the Supporting Information.

However, as the design space increases, the total amount of possible design patterns increase by orders of magnitude and the difficulty of design tasks increases dramatically. The results indicate that our proposed collaborative DQN method can have significant performance enhancement than that of the genetic algorithm. As shown in Figure 3b, for the 5*5 domain under the constraints of 10 stiff and 15 soft units, the genetic algorithm reaches its averaged historical best at 306.0 MPa after 9540 samples. The collaborative DQN method reaches its historical best at 323.5 MPa after 6400 samples. Although the genetic algorithm can discover better designs faster in the beginning, the collaborative DQN surpasses it after 1400 samples. From other learning curves in Figure S2 in the Supporting Information, it is found that the design quality of the collaborative DQN would become better or at least equal to the genetic algorithms gradually. When the design domain size increases to 7*7, the design task becomes more difficult. As shown in Figure 3c, under the constraints of 20 stiff and 29 soft units, the genetic algorithm reaches its averaged historical best at 306.5 MPa after 23 300 samples. The collaborative DQN method reaches its historical best at 355.1 MPa after 7000 samples. The result indicates that the proposed method has a 15.9% of design quality enhancement than that of the

### Table 1. Design Domain Sizes and Material Proportions

<table>
<thead>
<tr>
<th>domain size</th>
<th>ratios between material 0 and 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3*3 (easy)</td>
<td>[3:6], [5:4], [7:2]</td>
</tr>
<tr>
<td>5*5 (medium)</td>
<td>[5:20], [10:15], [15:10], [20:5]</td>
</tr>
<tr>
<td>7*7 (hard)</td>
<td>[10:39], [20:29], [30:19], [40:9]</td>
</tr>
</tbody>
</table>

Figure 3. Evaluation average return, training loss, and historical best return according to stiff-soft material ratio, with shaded area indicating the standard deviation: (a) 3*3 domain, 7 stiff and 2 soft units; (b) 5*5 domain, 10 stiff and 15 soft units; and (c) 7*7 domain, 20 stiff and 29 soft units.
genetic algorithm. Furthermore, the collaborative DQN can find optimal design faster than the genetic method after 1800 samples, and other learning curves in Figure S3 in the Supporting Information also show that the collaborative DQN has a higher sample efficiency than that of the genetic algorithm.

Unlike the genetic algorithm, the DRL method does not monotonically enhance the design quality with increased sample numbers. Instead, we can observe some perturbations in the evaluation average return curves from Figure 3 and Figure S1. This is due to the aggressive ε-greedy exploration strategy adopted in the experiments. ε refers to the possibility of the RL agent to take random action at each iteration, the higher the value, the policy would be less likely to be trapped into the local optima but at the risk of exploding loss and nonconvergence training. With current setting of ε = 0.5, the loss curves show that the two Q-networks Qnet1 and Qnet2 are always converging with a decreasing trend, and the evaluation average return curves of the proposed method shows better stability as the learning task becomes more complex.

The final proposed composite sheet configurations via the Collaborative DQN method and the corresponding \( E_{\text{mean}} \) is shown in Figure 4. By adding the symmetric information, the composite sheet size increases by two-fold, correspondingly \( 6 \times 6, 10 \times 10, \) and \( 14 \times 14 \) composite sheets. The stiff (red voxel) to soft (blue voxel) element ratio is expressed in the bracket above each configuration pattern and the number below indicates the \( E_{\text{mean}} \) of the design.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure4.png}
\caption{(a–c) Final proposed design configurations via Collaborative DQN method and corresponding \( E_{\text{mean}} \) for recovered \( 6 \times 6, 10 \times 10, \) and \( 14 \times 14 \) composite sheets. The stiff (red voxel) to soft (blue voxel) element ratio is expressed in the bracket above each configuration pattern and the number below indicates the \( E_{\text{mean}} \) of the design.}
\end{figure}

50% of the optimized structure designs obtained by the collaborative DQN method have \( E_{\text{mean}} > 400 \) MPa. This indicates that most of the highest performing samples are not included in the initial distribution, which can be one identical property of the DRL algorithm.

\section*{CONCLUSION}

This work explores the feasibility of applying DRL on a DM distribution optimization within a composite sheet under the constraint of fixed material proportion. The major advantages of the value function-based DRL method implemented in our proposed control system are as follows: (1) reduce the sample correlation, (2) require no pretraining process, and (3) are sample efficient. Our results show that the deep q-learning method has a tendency to outperform conventional optimization processes (e.g., genetic algorithms) as the environment gets more complicated, which benefits from the generalization ability of deep neural networks on large observation and action spaces. Future work includes tuning Q-network hyperparameters and testing on advanced exploration strategies, such as the Bootstrapped DQN.\cite{40} While the implementation of DQN-based methods have led to some pioneering results in this work, there are still important DRL approaches that have great potential, such as the actor–critic\cite{41} and model-based RL.\cite{42} Therefore, evaluations and comparisons of different DRL methods in the DM design should be further studied. Moreover, with only minor modifications, the DRL framework can be directly extended to many other fields concerning pixel-wise material design. We envision that our method can accelerate the design process and be applied to programmable smart composites and other intelligent manufacturing fields.

\section*{ASSOCIATED CONTENT}

\subsection*{Supporting Information}

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsmaterialslett.1c00390.

Collaborative DQN pseudo codes, DQN training and evaluation settings, and final design configurations with genetic algorithm (PDF)
AUTHOR INFORMATION

Corresponding Authors
Grace X. Gu – Department of Mechanical Engineering, University of California, Berkeley, California 94720, United States; orcid.org/0000-0001-7118-3228; Email: ggu@berkeley.edu
Liwei Lin – Department of Mechanical Engineering, University of California, Berkeley, California 94720, United States; Email: lwlin@berkeley.edu

Authors
Fanping Sui – Department of Mechanical Engineering, University of California, Berkeley, California 94720, United States
Ruqi Guo – Department of Mechanical Engineering, University of California, Berkeley, California 94720, United States
Zhizhou Zhang – Department of Mechanical Engineering, University of California, Berkeley, California 94720, United States

Complete contact information is available at: https://pubs.acs.org/10.1021/acsmaterialslett.1c00390

Author Contributions
These authors contributed equally to this work.

Author Contributions
The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

Funding
This work is supported by the Berkeley Sensor & Actuator Center, Powley Grant Funds, 3M, and HP Laboratories.

Notes
The authors declare no competing financial interest.

ACKNOWLEDGMENTS
Special thanks to Prof. Sergey Levine and Dr. Renxia Xu at Berkeley for the helpful discussions.

ABBREVIATIONS
DMs, digital materials; DRL, deep reinforcement learning; DQN, deep Q network; FEA, finite element analysis; AM, additive manufacturing; ML, machine learning; GN, generative networks; MDP, Markov decision process

REFERENCES


(31) Shalev-Shwartz, S.; Shammah, S.; Shashua, A. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. In *Proceedings of the NIPS Workshop Learning Inference and Control of Multi-Agent Systems*, 2016.


