

Article

Reinforcement Learning-Based Inverse Design of Multilayer Particles

Zhaohui Li ^{1,*} , Fang Gao ¹ and Delian Liu ² ¹ College of Communication and Information Technology, Xi'an University of Science and Technology, Xi'an 710054, China; gaofang@stu.xust.edu.cn² School of Optoelectronic Engineering, Xidian University, Xi'an 710071, China; dliu@xidian.edu.cn

* Correspondence: lizhaohui@xust.edu.cn

Abstract

Multilayered particles possess exceptional optical properties and hold significant potential for applications in chemical analysis, life sciences, optical sensing, and photonic integration. In practical applications, however, it is often necessary to perform inverse design of multilayered particles with given optical characteristics to meet specific requirements, a process that remains time-consuming. To overcome this challenge, we propose a reinforcement learning-based method for the automated design of multilayered particles. Leveraging the self-learning capacity of reinforcement learning models in combination with an optical characteristics calculation model, the method iteratively determines particle parameters that fulfill the desired optical responses. This method effectively addresses the many-to-one parameter mapping problem in inverse design, eliminates the need for extensive pre-computations, and provides an innovative approach to the automated design of complex nanostructures.

Keywords: reinforcement learning; multilayer particle; inverse design; Deep Q-Network

1. Introduction

Multilayered particles, known for their exceptional optical properties, find broad applications in photonic devices [1–3], environmental science [4,5], life sciences [6–8], photocatalysis [9–11], and optical sensing [12–14], are of considerable theoretical interest and practical value, attracting significant attention from researchers worldwide. The optical response of multilayered particles depends strongly on parameters such as material composition and layer thickness. Designing particles with given optical properties constitutes an inverse design problem [15]. A common approach is the enumeration method, which explores the parameter space, computes the corresponding optical responses, and selects the parameters that yield the desired outcomes. However, this process is computationally expensive and time-consuming.

In recent years, artificial intelligence (AI) techniques have been widely applied to the inverse design of nanostructures [16–21]. He et al. employed the discrete dipole approximation algorithm to calculate the optical responses of multilayered particles [22]. Building on this, they proposed an automated design method based on a multilayer perceptron network, which relied on precomputing a large set of optical response curves for various parameter configurations. This approach enabled the establishment of a mapping between optical responses and their corresponding parameter sets [23]. However, the method requires a one-to-one mapping between responses and parameter configurations. When



Academic Editor: Giovanni Angiulli

Received: 2 March 2026

Revised: 4 April 2026

Accepted: 6 April 2026

Published: 10 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

multiple parameter sets correspond to the same optical response, the neural network fails to converge. To overcome this many-to-one challenge, Liu et al. introduced a dual neural network architecture, with one network dedicated to forward design and the other to validation, thereby avoiding the convergence issue [24]. Similarly, Peurifo et al. combined neural networks with the gradient descent method to address the problem [25]. Despite these advances, both approaches still require extensive precomputation to train the forward neural network, which forms the basis of the inverse design. As a result, these methods remain heavily dependent on precomputed data and often struggle to identify optimal parameter configurations when the search space is large.

Reinforcement learning, an emerging paradigm in artificial intelligence, has been widely applied in domains such as autonomous driving, industrial control, and dynamic programming, achieving notable success. This approach, which involves constructing an agent that interacts with its environment, provides an effective solution to optimization problems. A key advantage of reinforcement learning is its ability to autonomously converge to an optimal strategy without relying on precomputed, large-scale training datasets, thus enabling the agent to learn and optimize through self-directed exploration. In recent years, significant progress has been made in applying reinforcement learning to the inverse design of nanostructures [26–30]. Building upon these advancements, we propose an automated inverse design method for multilayered particles based on a reinforcement learning model. This method enables the agent to automatically determine the parameter configurations of multilayered particles given specified optical response characteristics. The proposed reinforcement learning-based approach effectively addresses the many-to-one parameter mapping problem that often arises in the inverse design process. Furthermore, it obviates the need for extensive precomputed data, overcoming the limitation inherent in neural network algorithms that typically require large volumes of training data.

The contributions of this study are twofold. First, we propose an inverse design framework for multilayered particles based on reinforcement learning, where the design challenge is formulated as a sequential decision-making task. The core reinforcement learning components—states, actions, and reward functions—are systematically defined in accordance with the physical constraints of multilayered structures, facilitating the autonomous discovery of optimal configurations. Second, we develop a physics-informed reward mechanism that integrates the scattering characteristics of multilayered particles. This mechanism enables the agent to efficiently navigate highly nonlinear design spaces where conventional optimization methods often struggle to converge.

2. The Calculation of Optical Characteristics of Multilayer Particles

For multilayered particles, their optical characteristics are characterized by the absorption and scattering of incident light of different wavelengths. When a plane wave is incident on a multilayered particle, the electromagnetic field in each layer l can be expressed using an appropriate set of spherical wave functions [31–33]. The size parameter of each layer of the multilayered particle is

$$\chi_l = (2\pi N_m r_l) / \lambda = k r_l \quad (1)$$

The relative refractive index of each layer is

$$m_l = N_l / N_m \quad (2)$$

where $l = 1, 2, \dots, L$ denotes the index of each layer, λ is the wavelength of the incident wave in vacuum, r_l is the outer radius of the l th layer, N_l is the refractive index of the l th layer, N_m is the complex refractive index of the surrounding medium of the multilayered

particle, and k is the wave number. In the case, where the multilayered particle is in vacuum, the relative refractive index of the external region is $m_{L+1} = 1$.

Assuming the incident electric field is an x-polarized wave, then

$$\vec{E}_i = E_0 \exp[ikr \cos(\theta)] \vec{e}_\chi \quad (3)$$

Its time-dependent term is given by $\exp(-i\omega t)$, and ω is the angular frequency. The space is thus divided into two regions: the internal region of the multilayered particle and the external region. Both the electric and magnetic fields inside and outside the multilayered particle can be regarded as a superposition of spherical wave function sets.

Based on MIE theory, the incident wave $\vec{E}_{in}$ and the scattered wave $\vec{E}_{out}$ can be expressed using complex spherical eigenvector, namely,

$$\vec{E}_{in} = E_n \sum_{n=1}^{\infty} [c_n^{(l)} \vec{M}_{o1n}^{(1)} - id_n^{(l)} \vec{N}_{e1n}^{(1)}] \quad (4)$$

$$\vec{E}_{out} = \sum_{n=1}^{\infty} E_n [ia_n^{(l)} \vec{N}_{e1n}^{(3)} - b_n^{(l)} \vec{M}_{o1n}^{(3)}] \quad (5)$$

where $E_n = i^n E_0 (2n+1)/n(n+1)$, $\vec{M}_{o1n}^{(1)}$ and $\vec{N}_{e1n}^{(1)}$ are vector harmonic functions. When $j = 1$, they have radial dependence described by the spherical Bessel function of the first kind; when $j = 3$, they have radial dependence described by the spherical Hankel function of the first kind. The expressions for $\vec{M}_{o1n}^{(1)}$ and $\vec{N}_{e1n}^{(1)}$ can be found in reference [32].

In the external region of the particle, the total external field is the superposition of the incident field and the scattered field; that is,

$$\vec{E} = \vec{E}_i + \vec{E}_s \quad (6)$$

which can be expanded by,

$$\vec{E}_i = \sum_{n=1}^{\infty} E_n [\vec{M}_{o1n}^{(1)} - i\vec{N}_{e1n}^{(1)}] \quad (7)$$

$$\vec{E}_s = \sum_{n=1}^{\infty} E_n [ia_n \vec{N}_{e1n}^{(3)} - b_n \vec{M}_{o1n}^{(3)}] \quad (8)$$

Among them, a_n and b_n are the scattering coefficients. A more detailed calculation process regarding the scattering characteristics of multilayered particles can be found in [31,32]. Figure 1 illustrates the interaction between the incident light and the multilayered particle investigated in this study.

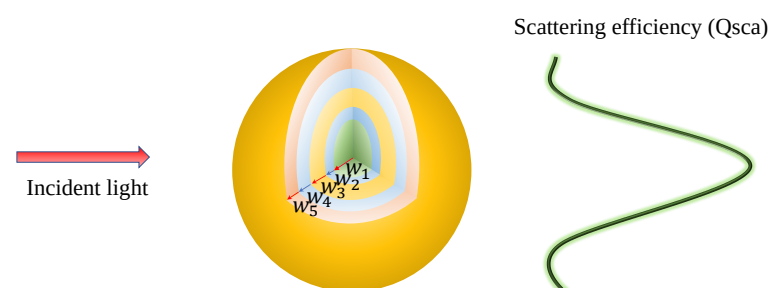


Figure 1. Schematic of incident light interacting with multiple layered particle.

The scattering characteristics of multilayered particles for incident light reflect their response to the incident optical field. Since the spectral scattering peaks of multilayered particles are highly sensitive to changes in particle shape, size, distribution, and the surrounding environment, designing multilayered particles with desired scattering char-

acteristics holds significant theoretical importance and practical value. The scattering characteristics of multilayered particles are represented by $Q_{sca}(\lambda)$. Once the material composition and the thickness of each layer are specified, $Q_{sca}(\lambda)$ can be calculated using the algorithms described in references [31,32].

In multilayer particles, a phenomenon occurs where multiple structural configurations correspond to the same scattering characteristics due to inherent physical properties. To illustrate this, numerical calculations were performed on a five-layer particle. The materials used are consistent with those reported in Ref. [34]. These materials and their corresponding refractive indices are listed in Table 1.

Table 1. Refractive indices of used materials.

Class (<i>I</i>)	0	1	2	3	4
Material	<i>SiO₂</i>	<i>MgO</i>	<i>ZnO</i>	<i>ZrO₂</i>	<i>TiO₂</i>
Refractive Index (<i>N</i>)	1.465	1.720	1.945	2.074	2.431

The thickness configurations for each layer are summarized in Table 2.

In Table 2, w_i denotes the width of each layer in unit of nanometer, and I_i denotes the refractive index of each layer. The calculated scattering characteristics are presented in Figure 2.

Table 2. Configuration of multilayer particles.

	w_1	w_2	w_3	w_4	w_5	I_1	I_2	I_3	I_4	I_5
No. 1	40	36	46	32	48	4	0	1	2	3
No. 2	42	42	42	42	36	0	1	1	2	3
No. 3	37	36	50	47	30	2	3	0	3	3
No. 4	32	44	45	46	38	4	0	1	2	3
No. 5	62	31	31	41	40	0	2	0	2	3

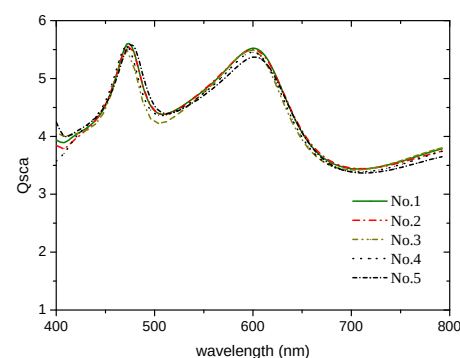


Figure 2. Scattering characteristics of 5-layer particles.

It can be observed from the figure that although the structural parameters and material compositions differ across these five particles, their scattering characteristics exhibit negligible differences (Mean-Square Error, MSE < 0.01).

A fundamental challenge in the inverse design of multilayer particle lies in the intrinsic many-to-one parameter mapping between structural parameters and optical spectra. From the perspective of MIE scattering theory, this degeneracy originates primarily from two physical mechanisms. First, phase accumulation degeneracies arise due to the periodic nature of light oscillation. The phase shift accumulated within a layer is given by $\phi = kNd = 2\pi/\lambda Nd$, which is periodic with respect to 2π . Consequently, different combinations of refractive index (N) and thickness (d) can yield the same phase condition,

resulting in identical constructive or destructive interference conditions. Second, resonance overlaps occur when distinct multipole modes, such as electric a_n and magnetic b_n Mie coefficients, spectrally overlap. A modification in the geometry may shift one mode while compensating with another, thereby preserving the overall scattering profile. This physical redundancy renders the inverse problem non-unique, posing a fundamental challenge for conventional data-driven approaches. Traditional neural networks, which typically assume a deterministic one-to-one mapping for regression tasks, often fail in this context: they tend to predict the average of multiple viable solutions, exhibit high prediction variance, or become trapped in local minima, thus failing to recover diverse yet physically valid structures corresponding to a target spectrum. To address this, our work employs reinforcement learning to effectively navigate the multimodal solution space.

3. Reinforcement Learning Model

As one of the three primary paradigms of machine learning, alongside supervised and unsupervised learning, reinforcement learning has found widespread applications in fields such as autonomous driving, path planning, and industrial control [35–37]. Reinforcement learning is one of the most active and frontier methods in artificial intelligence. It learns optimal decision-making strategies by establishing interactions between an agent and its environment, adjusting the agent's behavior based on the rewards received. Unlike traditional machine learning methods that rely on static datasets, reinforcement learning simulates the trial-and-error learning process observed in biological systems, enabling agents to autonomously discover optimal behavioral strategies in complex and uncertain environments.

The theoretical foundation of reinforcement learning is the Markov decision process, which formalizes automatic decision-making problems and enables algorithms to converge toward optimal policies. A typical reinforcement learning model consists of the following core components.

Agent: The decision-maker or learner, which interacts with the environment by observing states and taking actions to achieve goals.

Environment: The external system with which the agent interacts, providing responses to the agent's actions.

State: A representation of the current situation of the environment, which can be numerical values, vectors, or more abstract representations.

Action: The operations the agent can perform in a given state.

Reward: The immediate feedback signal from the environment, guiding the learning process.

The main idea of reinforcement learning is the principle of reward maximization. The agent explores different actions, observes the corresponding rewards, and gradually adjusts its policy to maximize the cumulative long-term reward. This process embodies the essence of trial-and-error learning, closely mirroring the way humans and animals learn from experience.

4. Multilayered Particle Automatic Inverse Design Based on Reinforcement Learning

Reinforcement learning model is an optimization model based on the Markov process. It solves optimization problems by constructing interactions between an agent and the environment as shown in Figure 3.

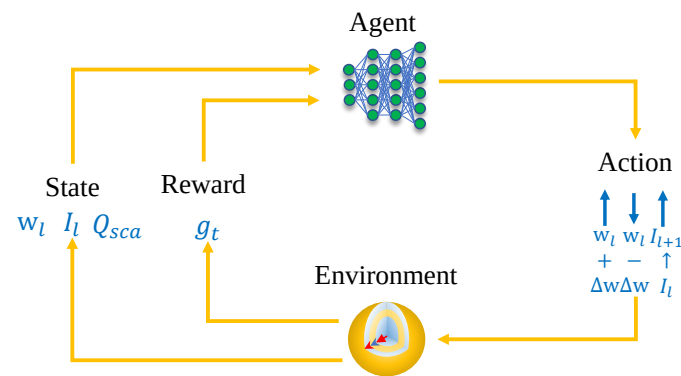


Figure 3. Schematic of multilayered particle inverse design based on reinforcement learning.

However, reinforcement learning model is merely a mathematical framework. To apply it to practical problems, it is necessary to construct the individual components of the reinforcement learning model according to its requirements. For the inverse design of multilayer particles, this means encoding each element of the reinforcement learning model to enable the search for optimal parameters. Below is the encoding of the core elements in the reinforcement learning-based automatic inverse design of multilayer particles to satisfy the requirements of the reinforcement learning model.

Agent: In this study, the agent serves as the core component for the design of multilayered particles, responsible for dynamically adjusting the parameters of each layer. Upon modifying the structural parameters, the Mie scattering theory is integrated to calculate the resulting scattering characteristics. These characteristics are then evaluated by a reward function, which provides the agent with a feedback signal (reward) corresponding to the executed action. Driven by the objective of maximizing cumulative rewards, the agent continuously optimizes its internal parameters (policy) to adapt to various environmental states and select appropriate actions. Ultimately, the convergence towards maximized rewards signifies the successful realization of the automatic inverse design for the multilayered particles. Here, the agent is modeled using a multilayer perceptron neural network with 5 layers, each containing 512 neurons, and the activation function used is the *tanh* activation function.

Environment: In this study, the environment is a simulation framework designed to address the optimization of parameter search. Specifically, under the condition of prescribed scattering characteristics of the multilayer particle, the task is to determine the thickness parameters of each layer. Given the variation in the scattering coefficient within a specified spectral range, the inverse problem is solved to reconstruct the thickness of the individual layers of the multilayer particle. The environment acts as an interface between the reinforcement learning agent and the electromagnetic solver discussed in Section 2. It is responsible for receiving the structural actions from the agent, updating the particle configuration, computing the optical response, and returning the corresponding reward.

State: In the reinforcement learning framework, the agent must continuously identify its current state. Through interaction with the environment, it obtains rewards and subsequently selects its next action based on these rewards. The state represents the input from the environment to the agent. In this study, the agent achieves the automated design of multilayer particles by iteratively adjusting the thickness of individual layers. The thickness configuration of each layer in the multilayer particle corresponds to the state. The encoding of the state must be uniquely defined. Once the parameters of the multilayer particle are determined, the state is also fixed; conversely, decoding the state yields the structural parameters of the particle. In this study, the state is encoded as a single design attempt for a multilayer particle. Accordingly, the state is represented by the

sequence of layer thicknesses from the innermost to the outermost layer of the multilayer nanoparticle w_l . At time step t , the state s_t is defined as a vector containing the thickness values of each layer, the refractive index of each layer and the scattering characteristics of current configuration: $s_t = [I_l, w_l, Q_{sca}(\lambda)]$. To ensure physical feasibility, the thickness of each layer is constrained within a bounded range of $w_l \in [30, 100]$ nm. The state space encompasses all possible thickness combinations within these constraints.

Action: The action is what the agent takes upon observing the environment's state. In this work, the action design must allow the agent to traverse the entire state space of the multilayer particle. To ensure high state coverage by the agent's actions, the actions are designed as shown in Table 3.

Table 3. The encoding of the agent's actions.

Actions	Meaning	Comments
Increase	$w_l + \Delta w$	The l layer thickness increases by Δw .
Decrease	$w_l - \Delta w$	The l layer thickness decreases by Δw .
Change	$I_l \rightarrow I_{l+1}$	The refractive index of layer l changed to another.

Each layer corresponds to 3 actions; for a five-layer particle, there will be 35 actions. Considering that the scattering efficiency of the multilayer particle change little with small variations in each layer, a minimum step size of $\Delta w = 2$ nm is set to traverse the entire state space. The search space is from 30 nm to 100 nm.

Reward: After the agent completes one exploration, the state reached by the environment corresponds to a design of the multilayer particle. Whether the multilayer particle designed by the agent is good or bad requires an appropriate evaluation. Based on the quality of the designed multilayer particle, a corresponding reward is given, which encourages the agent to modify its design and try again until the final criteria are met. Clearly, the design of the reward is key to enabling the agent to behave intelligently. The reward reflects the quality of the design, guiding the agent toward improved configurations through successive interactions. In this work, the design objective is to achieve high scattering within a specified wavelength range $[\lambda_l, \lambda_h]$ as illustrated in Figure 4.

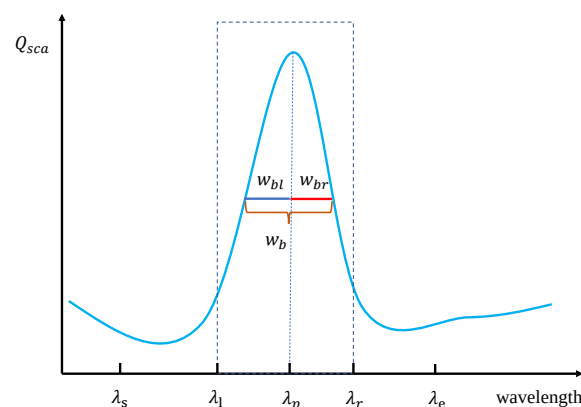


Figure 4. Wavelength range.

This study aims to design five-layer nanoparticles that exhibit specific scattering characteristics within a target wavelength range. The agent's search domain is defined as $[\lambda_s, \lambda_e]$, where λ_s and λ_e represent the starting and ending wavelengths, respectively. As illustrated in Figure 4, a scattering peak is assumed to exist within the investigated spectral range, characterized by its peak wavelength λ_p and full width at half maximum (FWHM) w_b . To evaluate spectral symmetry, the left and right half-widths of the peak are denoted as w_l and w_r , respectively.

The primary objective is to maximize in-band scattering. Specifically, the design requirements dictate that the scattering efficiency at the target wavelength should be maximized, the bandwidth should be minimized, and the scattering profile should exhibit high symmetry. Accordingly, the reward function g_t is formulated as:

$$g_t = \alpha_1 Q_{sca}(\lambda_t) + \alpha_2 B_b + \alpha_3 C_s \quad (9)$$

where α_1, α_2 , and α_3 are scaling factors; $Q_{sca}(\lambda_p)$ represents the scattering efficiency at the target wavelength λ_p ; and $B_b = 1/w_b$ is the reciprocal of the scattering bandwidth w_b . The symmetry component C_s is defined to reward configurations where the symmetry factor A_s approaches unity:

$$C_s = \frac{1}{\sqrt{2\pi}\sigma_s} \exp\left[-\frac{(A_s - 1)^2}{2\sigma_s^2}\right] \quad (10)$$

where the symmetry factor A_s is calculated as the ratio of the half-widths:

$$A_s = \frac{w_{bl}}{w_{br}} \quad (11)$$

This reward formulation enables the agent to effectively optimize multilayer nanoparticles to achieve high scattering efficiency and narrow-band performance within the desired spectral range.

5. Reinforcement Learning Algorithm

Reinforcement learning algorithms adjust their parameters based on rewards obtained from interactions between the agent and the environment, enabling the agent to exhibit intelligent behavior. Among these algorithms, the Deep Q-Network (DQN) is a classic method. DQN is a reinforcement learning model learning algorithm built on the foundation of Q-learning. It uses a deep neural network to replace the traditional Q-table used in Q-learning [38–40].

Q-learning stores the value of state-action pairs (s, a) in a Q-table to provide the best action path for the agent in subsequent steps. Suppose the value of the state-action pair (s, a) is $Q(s, a)$, which is stored in the Q-table. The Q-learning algorithm continuously updates this Q-table during the learning process, thereby enabling the agent to become intelligent. The Q-learning update rule is

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[g_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (12)$$

where, $Q(s, a)$ in the Q-table is updated to the current reward obtained after taking action a in state s plus the maximum expected future reward under the optimal policy. $\gamma \in (0, 1)$ is the discount factor, and g_t is the reward received after executing action a in state s . By continuously updating the Q-table with this formula, the agent's actions eventually converge to the optimal policy.

The core of this algorithm lies in using the Q-table to represent the agent's memory. However, when the state-action space is large—especially in problems like the inverse design of multilayer particles studied here—a huge Q-table must be constructed, making computation very time-consuming. Furthermore, a large Q-table leads to sparse Q-values, making it difficult for the agent to form an optimal path. To address these challenges, this study employs a DQN to facilitate autonomous agent learning. The standout feature of DQN compared to Q-learning is its use of a deep neural network to replace the Q-table. By inputting the state, the network directly outputs the Q values of actions, enabling map-

ping from states to actions and effectively solving decision problems in high-dimensional state spaces.

The detailed execution process of the DQN algorithm is as follows:

(1) Initialization

Initialize an experience replay buffer D to store transition samples (s_t, a_t, g_t, s_{t+1}) from agent–environment interactions. At the same time, create two neural networks.

- Online network $Q(s, a; \theta)$, responsible for selecting actions and updating parameters.
- Target network $Q(s, a; \theta^-)$, used to calculate target Q-values, with parameters (θ^-) periodically synchronized from the online network.

(2) Interaction and environment sampling

At each time step t :

- State observation: The agent receives the current state s_t and outputs the Q value of possible actions a_t .
- Action selection: Use an ϵ -greedy policy to select action a_t : with probability ϵ , select a random action (exploration); otherwise, select $a_t = \arg \max_a Q(s_t, a; \theta)$ (exploitation).
- Execute action: Perform a_t , receive reward g_t , and observe next state s_{t+1} .
- Store sample: Store (s_t, a_t, g_t, s_{t+1}) in the replay buffer D .

(3) Experience replay and training

Randomly sample a minibatch of samples (s_i, a_i, g_i, s_{i+1}) from the buffer:

- Compute target Q-values: If s_{i+1} is a terminal state, then $y_i = g_i$. Otherwise, use the target network to compute $y_i = g_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta^-)$, where γ is the discount factor.
- Compute the loss function: mean squared error $L(\theta) = 1/M \sum_i (y_i - Q(s_i, a_i; \theta))^2$.
- Perform gradient descent: update online network parameters θ via backpropagation to minimize the loss.

(4) Target network update

Every C steps, copy the online network parameters θ to the target network (hard update), or perform a soft update by $\theta^- \leftarrow \tau \theta + (1 - \tau) \theta^-$.

(5) Loop and termination

Repeat the above steps until reaching the maximum number of training steps or convergence. During training, gradually decay ϵ to balance exploration and exploitation.

Through the above steps, DQN learns a policy that approximates the optimal Q-function.

6. Simulation Results and Discussion

6.1. Inverse Design Multilayer Particle with One Scattering Band Feature

The proposed method is employed for the inverse design of a five-layer nanoparticle composed of five different materials. These materials and their corresponding refractive indices are listed in Table 1. The objective is to refine the scattering spectrum of the multilayered particle within the 520–600 nm range, specifically targeting high scattering at 560 nm.

The execution steps of the proposed method are as follows:

- (1) Set the search range for the thickness of each layer. In this study, the minimum thickness of each layer is assumed to be no less than 30 nm, and the maximum thickness no more than 100 nm.
- (2) Randomly generate the refractive index I_l of each layer of the multilayered particle. The agent begin to design the multilayered particle from a start thickness. These

values are input into the agent, which is the aforementioned multilayer perceptron network. The agent then outputs Q value of the action for the current state.

- (3) The agent adjusts the design parameters of the particle (I_l , and w_l) based on the selected action.
- (4) After the state of the multilayered particle changes, the method described in Section 2 is used to calculate the scattering efficiency $Q_{sca}(\lambda)$ under the new parameter configuration.
- (5) The scattering efficiency $Q_{sca}(\lambda)$ is substituted into Equation (9) to compute the reward. The reward result, state, and action (s_t, a_t, g_t, s_{t+1}) are stored in the experience replay buffer (D).
- (6) The agent is trained utilizing the experience replay and training methodologies detailed in Section 5.
- (7) Steps (3) to (6) are repeated until the maximum number of training steps is reached or convergence is achieved.
- (8) The multilayer perceptron network parameters of the agent are saved.

The parameters for training the reinforcement learning model are given in Table 4.

Table 4. Parameters of the reinforcement learning model for training.

Parameters	Value
Number of layers	5
Number of neurons per layer	512
Activation functions	tanh
Initial learning rate	5×10^{-5}
Step learning rate	$\gamma_s = 0.1$, step size = 1000
Episodes number	2000
Batch size	256
τ	0.1
γ	0.98
α_1	1
α_2	0.2
α_3	0.05
σ_s	0.5

In the calculation of $Q_{sca}(\lambda)$ of the multilayer nanoparticle, the wavelength interval was set to 8 nm. The truncation order was obtained using the calculation method from the literature [41]. The result is shown in Figure 5.

Figure 5a shows the reward variation curve during the training process of the agent. The red curve represents the reward obtained in each epoch, while the blue curve shows the average reward over 100 epochs. Higher values indicate that the agent has found results consistent with expectation. Training is stopped after 2000 epoches, and the agent's parameters are saved. By loading the trained network parameters and randomly initializing the refractive index I_l of the each layers, the agent achieves the desired result through self-iteration (I : [4, 0, 0, 4, 4]; w : [34, 36, 46, 46, 42]). Figure 5b is the scattering curve of the designed multilayer particle, which shows strong scattering value in the given band. Figure 5c shows the variation in the five-layer parameters during one design instance by the agent. It illustrates the refractive index and the thickness variations in the five layers during the agent's design process. The corresponding changes in the Q_{sca} curve are shown in Figure 5d. In the early stages of automated design, the Q_{sca} curve differs significantly from the expectation. After adjustments to each layer, a scattering peak is optimized, but its shape still deviates from the expectation. The agent continues to fine-tune the thicknesses of each layer, gradually modifying the scattering peak to the expectation shape, ultimately achieving the design goal.

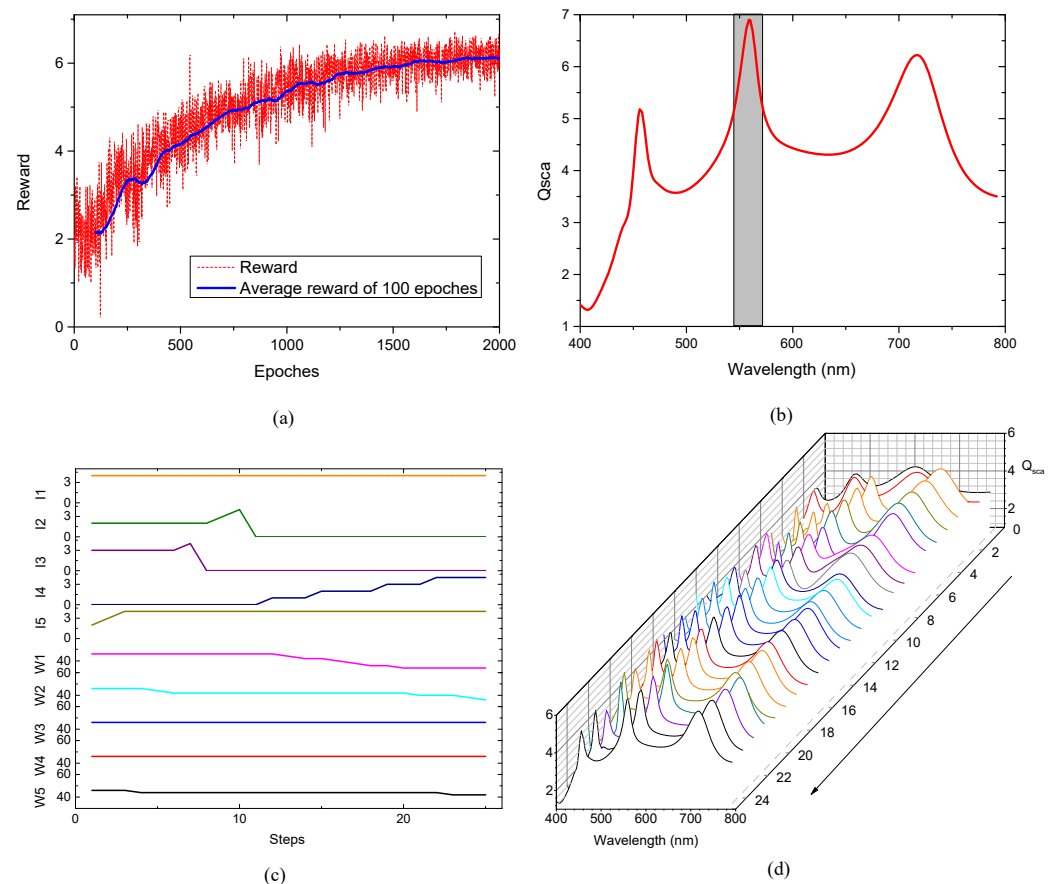


Figure 5. Result of inverse design multilayer particle with one scattering band feature. (a) Reward variation curve during the training process of the agent. (b) Scattering efficiency curve of the designed multilayer particle. (c) Variation of the five-layer parameters during one design instance by the agent. (d) Corresponding changes of the scattering efficiency curve.

6.2. Inverse Design Multilayer Particle with Two Scattering Band Features

The previously introduced inverse design of multilayer nanoparticles featuring a single scattering band can be expanded to meet specific requirements by simply modifying the corresponding reward function. For the inverse design of multilayer particles featuring two scattering bands, the reward function described in Section 5 is modified. Considering that both scattering bands should ideally satisfy the same optimization criteria, the reward function is defined as:

$$g_t = g_{t1} \cdot g_{t2} \quad (13)$$

where g_{t1} and g_{t2} represent the reward values within the two specified wavelength ranges, respectively. In this instance, the multilayer nanoparticle is designed to exhibit a scattering peak at 496 nm within Band 1 (456 nm to 546 nm) and another scattering peak at 600 nm within Band 2 (560 nm to 640 nm). Utilizing the parameters listed in Table 4, the results of this instance are illustrated in Figure 6.

Figure 6a illustrates the reward curve, showing that the rewards obtained by the agent gradually increase as training progresses before eventually plateauing. Training is terminated after 2000 episodes, at which point the agent's weights and biases are stored. Subsequently, by loading the trained agent and providing it with random refractive indices for the multilayer structure, the agent can successfully design a five-layer nanoparticle that fulfills the specified requirements. Figure 6b presents the scattering characteristics of the nanoparticle designed by the agent ($\mathbf{I}$: [0, 0, 0, 4, 4]; $\mathbf{w}$: [44, 46, 46, 44, 42]), demonstrating that the performance is consistent with expectations. Figure 6c depicts the evolution of the nanoparticle parameters

during a specific design process, while Figure 6d shows the corresponding changes in its scattering characteristics. These results indicate that the trained agent is capable of achieving the inverse design of multilayer nanoparticles featuring dual scattering bands.

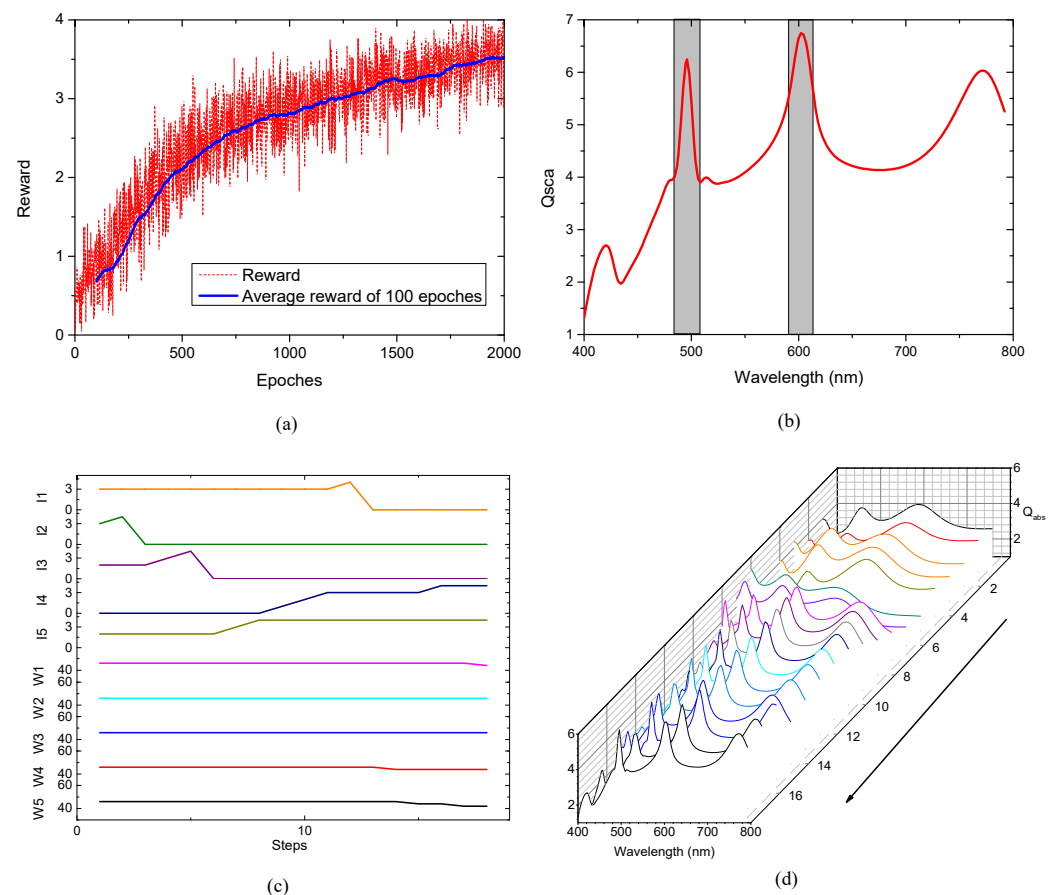


Figure 6. Result of inverse design multilayer particle with two scattering band feature. (a) Reward variation curve during the training process of the agent. (b) Scattering efficiency curve of the designed multilayer particle. (c) Variation of the five-layer parameters during one design instance by the agent. (d) Corresponding changes of the scattering efficiency curve.

6.3. Inverse Design Multilayer Particle with Given Scattering Spectrum

In certain scenarios, a specific scattering spectrum is provided as a target, and the objective is to inversely design a multilayer nanoparticle that reproduces this spectrum. To meet this requirement, the reward function is further modified as:

$$g_t = \frac{1}{\text{MSE} + \xi} \quad (14)$$

where ξ is a small constant—set to 0.05 in this study—to prevent division by zero. MSE represents the mean squared error between the scattering spectrum of the current multilayer nanoparticle and the reference spectrum. When the MSE falls below a small threshold (e.g., 0.05), indicating that the design requirements have been satisfied, a reward of 100 is returned and the search process is terminated. The agent is trained using the same parameters as previously listed, with the exception of the initial learning rate (l_r), which is set to 1×10^{-4} . The results are illustrated in Figure 7.

Figure 7a shows the evolution of the reward values. As training progresses, the reward gradually increases. Unlike the previous cases, the curve exhibits significant fluctuations (jitter); this occurs because the agent successfully identifies structures that satisfy the

threshold requirements, triggering high reward spikes. Figure 7b compares the scattering spectrum of the nanoparticle designed by the agent (I : [0, 1, 1, 2, 3]; w : [44, 38, 46, 38, 38]) with the target (desired) spectrum. The minimal discrepancy between the two confirms that the agent is capable of designing multilayer nanoparticles that meet specific spectral requirements. Figure 7c depicts the evolution of the refractive indices and thicknesses of each layer during a specific design iteration, along with the corresponding changes in the scattering spectrum (Figure 7d). These results demonstrate that reinforcement learning is an effective approach for the inverse design of complex multilayer nanoparticles.

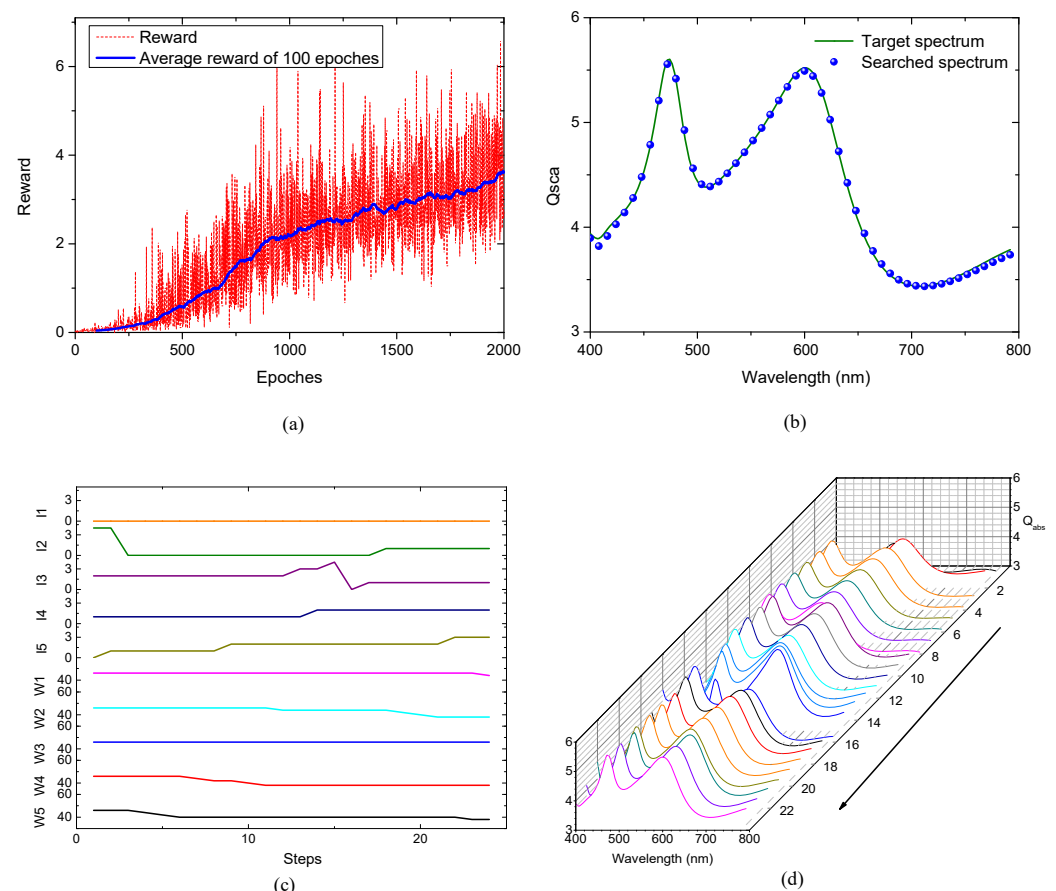


Figure 7. Result of inverse design multilayer particle with given scattering spectrum. (a) Reward variation curve during the training process of the agent. (b) Scattering efficiency curve of the nanoparticle designed by the agent with the target (desired) curve. (c) Variation of the five-layer parameters during one design instance by the agent. (d) Corresponding changes of the scattering efficiency curve.

The evolutionary dynamics of the design parameters are shown in Figures 5c, 6c and 7c; the corresponding final converged material and structural values are detailed in Table 5.

Table 5. Final geometry and material parameters for the three experiments.

Case 1			Case 2		Case 3	
Layer	Material	Width	Material	Width	Material	Width
0	TiO_2	34	SiO_2	44	SiO_2	44
1	SiO_2	36	SiO_2	46	MgO	38
2	SiO_2	46	SiO_2	46	MgO	46
3	TiO_2	46	TiO_2	44	ZnO	38
4	TiO_2	42	TiO_2	42	ZrO_2	38

6.4. Comparison with Other Traditional Algorithms

The objective of this study is the inverse design of multilayer particles to match a prescribed reference scattering spectrum. As a typical inverse problem, this can be addressed using classical optimization techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). To evaluate their performance, we conducted a comparative study of these three representative algorithms. All these algorithms aim to design multilayer particles whose scattering spectra match the reference spectrum described in Section 6.3. The variation curves of the reward values of these algorithms during the optimization iteration process are shown in Figure 8.

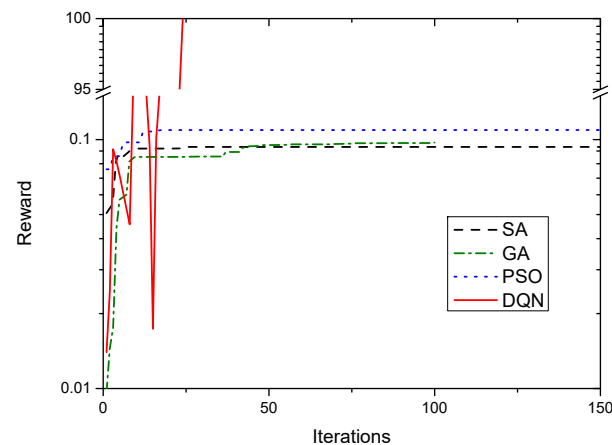


Figure 8. Comparison optimization results of the multilayer particle of GA, SA, PSO, and DQN.

As shown in Figure 8, the performance follows a clear hierarchy: PSO outperforms GA, which in turn outperforms SA. However, these traditional methods are susceptible to converging on local optima, which can compromise the accuracy of the solutions. In contrast, the proposed reinforcement learning-based algorithm exhibits a superior capability to identify the better solution than that of the evaluated methods, effectively overcoming the limitations of conventional optimization approaches.

To evaluate the time efficiency of different algorithms, all experiments were conducted on a computer with a 3.4 GHz CPU (Intel® Core™ i7-13700KF) and 65,536 MB of memory for the automatic design of multilayer particles. Table 6 lists the final structures obtained by each algorithm, along with their respective computational times.

Table 6. Comparison of final design outcomes and computational time across the algorithms.

Layer	SA		GA		PSO		DQN	
	Material	Width	Material	Width	Material	Width	Material	Width
0	SiO ₂	30.5	TiO ₂	35.0	SiO ₂	71.9	SiO ₂	44
1	SiO ₂	81.6	MgO	39.0	SiO ₂	81.8	MgO	38
2	SiO ₂	83.7	SiO ₂	53.0	SiO ₂	30.0	MgO	46
3	MgO	31.4	MgO	39.0	MgO	81.5	ZnO	38
4	MgO	68	ZnO	53.0	MgO	30.0	ZrO ₂	38
time	642 s		110 s		47 s		train: 3496 s, execution < 1 s	

As demonstrated in Table 6, the PSO algorithm achieves the shortest computational duration, followed by GA and SA, whereas DQN incurs the most significant training overhead. This highlights a characteristic trade-off in learning-based algorithms. However, once the training phase is complete, the DQN model executes in under one second, substantially outperforming PSO, GA, and SA in terms of inference speed.

Beyond traditional heuristic algorithms, we also compare our DQN-based algorithm with recent deep learning approaches for inverse design, specifically Tandem Neural Networks (TNNs) and Generative Adversarial Networks (GANs).

The results of GANs and TNNs that aim to design a multilayer particle with scattering spectrum, such as the one in Section 6.3, are summarized in Table 7, which lists the material compositions and structural parameters output by each method. The corresponding absorption spectra of these particles are shown in Figure 9.

Table 7. Comparison of final design outcomes across the algorithms.

Layer	GANs		TNNs		DQN	
	Material	Width	Material	Width	Material	Width
0	ZnO	35.89	SiO ₂	53.12	SiO ₂	44
1	ZnO	35.21	SiO ₂	62.54	MgO	38
2	ZnO	38.80	ZnO	100.83	MgO	46
3	ZnO	35.26	SiO ₂	30.18	ZnO	38
4	ZnO	60.01	MgO	46.61	ZrO ₂	38

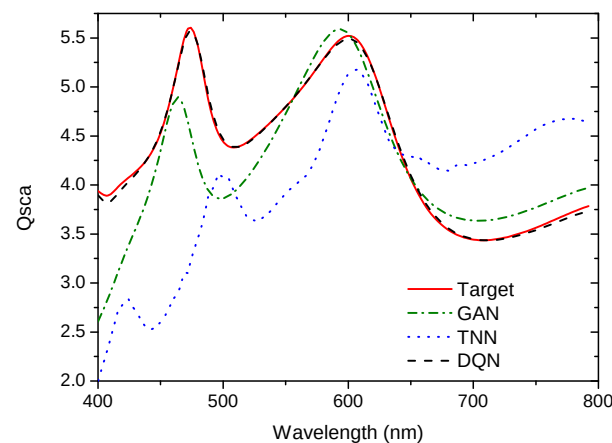


Figure 9. Scattering spectra of the multilayer particle of GANs, TNNs, and DQN.

As shown in Figure 9, although both GANs and TNNs can produce spectra resembling the target, notable discrepancies persist. Among them, GANs achieve better spectral matching than TNNs, yet both are outperformed by DQN. It is worth noting that the TNN and GAN models were each pre-trained on 100,000 randomly generated structure–spectrum pairs—a large, pre-computed dataset. In contrast, DQN learns online without any pre-training data, reusing experiences via a replay buffer to achieve superior data efficiency. Moreover, in addressing the many-to-one mapping challenge, DQN’s discrete action space and Q-value distribution naturally accommodate multiple valid solutions, whereas regression-based TNNs tend to average over the solution space. Collectively, these results suggest that the proposed DQN-based method offers distinct advantages for the inverse design of multilayer particles, particularly in terms of data efficiency, design accuracy, and resilience to solution non-uniqueness.

Several factors may contribute to this performance gap. One notable difference lies in how each method handles discrete refractive indices. In this study, the refractive index of each layer is restricted to a set of discrete values. GANs and TNNs directly output continuous values, which must be truncated to the nearest discrete allowed value, potentially introducing rounding errors that could affect spectral fidelity. DQN, in contrast, operates on a discrete action space: each action directly corresponds to a specific refractive index (or thickness step), thereby avoiding post-output truncation. Additionally, TNNs employ a

cascaded architecture where the inverse network's output is fed into a pre-trained forward network to compute the spectral loss. This forward pass, trained on continuous data, may be sensitive to discontinuities introduced by truncation, and the error is backpropagated through two networks, which could amplify inaccuracies. GANs avoid the cascaded forward pass but still face the inherent rounding step.

That said, it is not conclusive that the performance gap is solely or exclusively due to truncation. Other factors—such as network architectures, training dynamics, loss functions, and hyperparameter choices—may also play important roles. With sufficiently extensive tuning (e.g., adjusting network depth, activation functions, regularization, or learning schedules), it is possible that TNNs or GANs could narrow the gap to DQN, or even achieve comparable performance in certain cases. However, given the representational differences between continuous-output regression and discrete-action selection, whether such methods can fully close the gap without fundamental architectural changes remains an open question. Future work may further investigate this comparison under controlled conditions.

The simulation results demonstrate that the reinforcement learning model can be effectively applied to the automated design of complex nanostructures and yields highly satisfactory results.

6.5. Robustness and Tolerance Analysis

To assess the robustness of the reinforcement learning-designed particles against realistic manufacturing imperfections, we conducted a fabrication tolerance analysis. In typical nanoparticle synthesis processes (e.g., Atomic Layer Deposition or colloidal assembly), layer thickness control is subject to stochastic variations. To simulate this, we introduced random noise to the optimal layer thicknesses (w_l^{opt}) predicted by the DQN model. Specifically, we added independent Gaussian noise $r_i \sim N(0, \sigma^2)$. For practical applications, fabrication errors are typically up to ± 5 nm [42,43]. Accordingly, we set the standard deviation to $\sigma = 2.5$ nm such that the maximum error is ± 5 nm under the 2σ criterion.

A total of 10 independent noise realizations were applied to the optimal design, and the corresponding scattering spectra were calculated. The results are shown in Figure 10. It is observed that the perturbed spectra deviate from the ideal reinforcement learning-designed spectrum, spreading to both sides of the target curve. Despite this spread, all spectra remain in close proximity to the target, with the primary resonance peaks shifting by less than ± 10 nm. This indicates that the reinforcement learning-designed structure is robust against common fabrication uncertainties, exhibiting no extreme sensitivity to minor geometric perturbations. Such robustness is a crucial requirement for scalable nanophotonic applications, confirming the practical viability of the reinforcement learning-designed particles. This analysis demonstrates how the designed structures perform under non-ideal conditions, rather than serving as a validation of the learning algorithm itself.

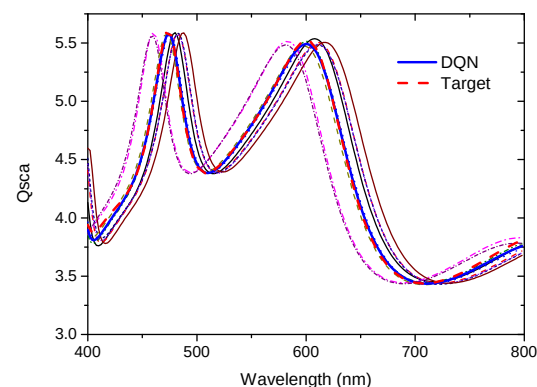


Figure 10. Robustness and tolerance analysis.

7. Conclusions

To address the issues of one-to-many mapping and the need for extensive precomputation in the current automatic design of multilayer particles, this paper proposes a reinforcement learning-based automatic design method. By leveraging the self-learning capability of reinforcement learning combined with the optical characteristic calculation model of multilayer particles, the method automatically iterates to solve for multilayer particle parameters that meet the required optical characteristics. Simulation results demonstrate that the proposed reinforcement learning-based automatic inverse design method effectively avoids the one-to-many parameter mapping problem during the inverse design process and can automatically solve problems within a large parameter space. Moreover, the design process does not require extensive precomputed training data, thereby overcoming the data dependency issues faced by neural network-based algorithms.

In conclusion, this study demonstrates the feasibility of using DQN-based reinforcement learning for the inverse design of multilayer particles, achieving a five-layer structure with target optical spectra. While the simulation results validate the algorithm's ability to navigate complex, high-dimensional design spaces, several practical and theoretical challenges remain. From an experimental perspective, the proposed 5-layer core-shell particle (Table 2) consisting of five distinct materials presents significant synthesis hurdles. Fabricating such a structure requires precise control over interfacial compatibility and lattice matching to prevent defects. Techniques such as Atomic Layer Deposition (ALD) or colloidal layer-by-layer assembly would be necessary to achieve the required interface quality, yet scaling these methods to maintain distinct material layers without interdiffusion remains a non-trivial chemical challenge.

Regarding the algorithmic framework, the current DQN implementation operates on discrete action spaces, necessitating the discretization of continuous parameters like layer thickness. This discretization limits the precision of the solution and introduces step-size dependencies. Future work will focus on transitioning to continuous-action Reinforcement Learning algorithms, such as Deep Deterministic Policy Gradient (DDPG) or Soft Actor-Critic (SAC). These algorithms would allow for the direct optimization of continuous thickness parameters without predefined grids, potentially yielding higher-fidelity designs.

Author Contributions: Z.L.: conceptualization, formal analysis, data curation, supervision, writing—original draft. F.G.: software, validation, visualization. D.L.: investigation, methodology, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (No. 62274124).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
DQN	Deep Q-Network
GA	Genetic Algorithms
GANs	Generative Adversarial Networks
MSE	Mean-Square Error
PSO	Particle Swarm Optimization
SA	Simulated Annealing
TNNs	Tandem Neural Networks

References

1. Yang, T.; Fu, T.; An, Y. Radiation direction mutation in a spherical plasma filled multilayered core—Shell particle. *Phys. Plasmas* **2022**, *29*, 012103. [\[CrossRef\]](#)
2. Kailash; Verma, S. Opto-thermal properties of some composite metallic nanoshells for their thermoplasmonic applications. *Plasmonics* **2024**, *19*, 1607–1618. [\[CrossRef\]](#)
3. Huang, J.; Tao, L.; Wei, H.; Huang, H.; Zhang, Q.; Zhou, B. Full-color tuning in multi-layer core-shell nanoparticles from single-wavelength excitation. *Nat. Commun.* **2025**, *16*, 2378. [\[CrossRef\]](#)
4. Gordon, J.A. *Coated Nano Particles for Optical Metamaterials and Nano-Photonic Applications*; The University of Arizona: Tucson, AZ, USA, 2008.
5. Wang, G.; Li, Z.; Hu, C.; Yang, G.; Yang, X.; Liu, B. Deep learning-driven Mie scattering prediction method for radially varying spherical particles. *Opt. Laser Technol.* **2024**, *177*, 111170. [\[CrossRef\]](#)
6. Jena, B.K.; Raj, C.R. Optical sensing of biomedically important polyionic drugs using nano-sized gold particles. *Biosens. Bioelectron.* **2008**, *23*, 1285–1290. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Jayakumar, K.; Rajesh, R.; Dharuman, V.; Venkatesan, R.; Hahn, J.; Pandian, S.K. Gold nano particle decorated graphene core first generation PAMAM dendrimer for label free electrochemical DNA hybridization sensing. *Biosens. Bioelectron.* **2012**, *31*, 406–412. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Wang, W.; Chen, X.; Huang, W.C.; Di, S.; Luo, J. Development of Multilayer Magnetic Janus Sub-Micrometric Particles for Lipase Catalysis in Pickering Emulsion. *Molecules* **2025**, *30*, 2429. [\[CrossRef\]](#)
9. Wang, S.; Yang, Y.; Liu, P.; Zhang, Z.; Zhang, C.; Chen, A.; Ajao, O.O.; Li, B.G.; Braunstein, P.; Wang, W.J. Core-shell and yolk-shell covalent organic framework nanostructures with size-selective permeability. *Cell Rep. Phys. Sci.* **2020**, *1*, 100062. [\[CrossRef\]](#)
10. Sayed, M.; Xu, F.; Kuang, P.; Low, J.; Wang, S.; Zhang, L.; Yu, J. Sustained CO₂-photoreduction activity and high selectivity over Mn, C-codoped ZnO core-triple shell hollow spheres. *Nat. Commun.* **2021**, *12*, 4936. [\[CrossRef\]](#)
11. Wu, J.Y.; Wei, Y.C.; Torimoto, T.; Chien, Y.A.; Chen, C.Y.; Chang, T.F.M.; Sone, M.; Hsieh, P.Y.; Hsu, Y.J. Yolk@ shell nanostructures for water splitting: Current development and future prospects. *ACS Mater. Lett.* **2024**, *6*, 4066–4089. [\[CrossRef\]](#)
12. Monzón-Hernández, D.; Luna-Moreno, D.; Escobar, D.M.; Villatoro, J. Optical microfibers decorated with PdAu nanoparticles for fast hydrogen sensing. *Sens. Actuators B Chem.* **2010**, *151*, 219–222. [\[CrossRef\]](#)
13. Tan, R.; Guo, Y.; Zhao, J.; Li, Y.; Xu, T.; Song, W. Synthesis, characterization and gas-sensing properties of Pd-doped SnO₂ nano particles. *Trans. Nonferrous Met. Soc. China* **2011**, *21*, 1568–1573.
14. Kim, J.H.; Choi, H.W. Numerical Analysis of Laser-Excited SAM-Coated Magnetic Nanoparticles for Electromagnetic Field Enhancement in Optical Gas Sensing. *Sensors* **2026**, *26*, 31. [\[CrossRef\]](#)
15. Soun, D.; Azema, A.; Roach, L.; Drisko, G.I.; Wiecha, P.R. Gradient-based optimization of core-shell particles with discrete materials for directional scattering. *Opt. Express* **2025**, *33*, 25945–25958. [\[CrossRef\]](#)
16. Ma, W.; Liu, Z.; Kudyshev, Z.A.; Boltasseva, A.; Cai, W.; Liu, Y. Deep learning for the design of photonic structures. *Nat. Photonics* **2021**, *15*, 77–90. [\[CrossRef\]](#)
17. Kim, M.J.; Kim, J.T.; Hong, M.J.; Park, S.W.; Lee, G.J. Deep learning-assisted inverse design of nanoparticle-embedded radiative coolers. *Opt. Express* **2024**, *32*, 16235–16247. [\[CrossRef\]](#)
18. Umetaliev, T.; Valagiannopoulos, C. AI-based photonic inverse design: Hugely polarization-selective multilayered scatterers. *J. Opt. Soc. Am. B* **2025**, *42*, 621–630. [\[CrossRef\]](#)
19. Wu, N.; Sun, Y.; Hu, J.; Yang, C.; Bai, Z.; Wang, F.; Cui, X.; He, S.; Li, Y.; Zhang, C.; et al. Intelligent nanophotonics: When machine learning sheds light. *Light* **2025**, *5*, 5. [\[CrossRef\]](#)
20. Chen, Y.; McNeil, A.M.; Park, T.; Wilson, B.A.; Iyer, V.; Bezick, M.; Choi, J.I.; Ojha, R.; Mahendran, P.; Singh, D.K.; et al. Machine-learning-assisted photonic device development: A multiscale approach from theory to characterization. *Nanophotonics* **2025**, *14*, 3761–3793. [\[CrossRef\]](#)
21. Riganti, R.; Zhu, Y.; Cai, W.; Torquato, S.; Dal Negro, L. Multiscale Physics-Informed Neural Networks for the Inverse Design of Hyperuniform Optical Materials. *Adv. Opt. Mater.* **2025**, *13*, 2403304. [\[CrossRef\]](#)
22. He, W.; Huang, X.; Ma, X.; Zhang, J. Simulation of yolk-shell nanostructures optical properties. *J. Nanophotonics* **2023**, *17*, 016003. [\[CrossRef\]](#)
23. He, W.; Ma, X.; Zhang, J.; Xu, K.; Gao, J.; Lei, S.; Zhan, C. A calculation method for optical properties of yolk shell based on deep learning. *PLoS ONE* **2024**, *19*, e0302262. [\[CrossRef\]](#)
24. Liu, D.; Tan, Y.; Khoram, E.; Yu, Z. Training deep neural networks for the inverse design of nanophotonic structures. *Acs Photonics* **2018**, *5*, 1365–1369. [\[CrossRef\]](#)
25. Peurifoy, J.; Shen, Y.; Jing, L.; Yang, Y.; Cano-Renteria, F.; DeLacy, B.G.; Joannopoulos, J.D.; Tegmark, M.; Soljačić, M. Nanophotonic particle simulation and inverse design using artificial neural networks. *Sci. Adv.* **2018**, *4*, eaar4206. [\[CrossRef\]](#)
26. AlKhonaini, A.; Sheltami, T.; Mahmoud, A.; Imam, M. UAV Detection Using Reinforcement Learning. *Sensors* **2024**, *24*, 1870. [\[CrossRef\]](#)

27. Li, Z.; Yang, S.; Liu, D. Automatic inverse design of second-order differential metasurfaces based on reinforcement learning. *Eng. Appl. Artif. Intell.* **2025**, *153*, 110812. [[CrossRef](#)]
28. Yang, G.; Xiao, Q.; Zhang, Z.; Yu, Z.; Wang, X.; Lu, Q. Exploring AI in metasurface structures with forward and inverse design. *iScience* **2025**, *28*, 111995. [[CrossRef](#)]
29. Kim, M.; Park, H.; Shin, J. Nanophotonic device design based on large language models: Multilayer and metasurface examples. *Nanophotonics* **2025**, *14*, 1273–1282. [[CrossRef](#)]
30. Yeung, C.; Pham, B.; Zhang, Z.; Fountaine, K.T.; Raman, A.P. Hybrid supervised and reinforcement learning for the design and optimization of nanophotonic structures. *Opt. Express* **2024**, *32*, 9920–9930. [[CrossRef](#)]
31. Yang, W. Improved recursive algorithm for light scattering by a multilayered sphere. *Appl. Opt.* **2003**, *42*, 1710–1720. [[CrossRef](#)]
32. Peña, O.; Pal, U. Scattering of electromagnetic radiation by a multilayered sphere. *Comput. Phys. Commun.* **2009**, *180*, 2348–2354. [[CrossRef](#)]
33. Ladutenko, K.; Pal, U.; Rivera, A.; Peña-Rodríguez, O. Mie calculation of electromagnetic near-field for a multilayered sphere. *Comput. Phys. Commun.* **2017**, *214*, 225–230. [[CrossRef](#)]
34. Kuhn, L.; Repän, T.; Rockstuhl, C. Inverse design of core-shell particles with discrete material classes using neural networks. *Sci. Rep.* **2022**, *12*, 19019. [[CrossRef](#)]
35. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)]
36. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [[CrossRef](#)]
37. Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **2018**, *362*, 1140–1144. [[CrossRef](#)]
38. Wang, J.; Gou, L.; Shen, H.W.; Yang, H. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE Trans. Vis. Comput. Graph.* **2018**, *25*, 288–298. [[CrossRef](#)]
39. Wang, Y.; Liu, H.; Zheng, W.; Xia, Y.; Li, Y.; Chen, P.; Guo, K.; Xie, H. Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning. *IEEE Access* **2019**, *7*, 39974–39982. [[CrossRef](#)]
40. Li, J.; Chen, Y.; Zhao, X.; Huang, J. An improved DQN path planning algorithm. *J. Supercomput.* **2022**, *78*, 616–639. [[CrossRef](#)]
41. Bohren, C.F.; Huffman, D.R. *Absorption and Scattering of Light by Small Particles*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
42. Li, K.K.; He, J.; Huang, Q.; Kinoshita, S.; Ding, Y.; Xia, Y. Rational Synthesis of Uniform Au Nanospheres under One-Shot Injection: From Mechanistic Understanding to Experimental Control. *Precis. Chem.* **2025**, *3*, 272–278. [[CrossRef](#)]
43. Kolar-Hofer, P.; Zampini, G.; Derntl, C.G.; Soprano, E.; Polo, E.; Del Pino, P.; Kereyeva, N.; Eggeling, M.; Breth, L.; Haslinger, M.J.; et al. Fabrication of nanoparticles with precisely controllable plasmonic properties as tools for biomedical applications. *Nanoscale* **2025**, *17*, 4423–4438. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.