Fall 2022

Acoustic Cloak Design Using Generative Modeling and Reinforcement Learning

Linwei Zhuo
San Jose State University

ACOUSTIC CLOAK DESIGN USING GENERATIVE MODELING AND REINFORCEMENT LEARNING

A Thesis

Presented to

The Faculty of the Department of Mechanical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Linwei Zhuo

December 2022

The Designated Thesis Committee Approves the Thesis Titled

ACOUSTIC CLOAK DESIGN USING GENERATIVE MODELING AND REINFORCEMENT LEARNING

by

Linwei Zhuo

APPROVED FOR THE DEPARTMENT OF MECHANICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

December 2022

Feruza A. Amirkulova, Ph.D.	Department of Mechanical Engineering
Teng Moh, Ph.D.	Department of Computer Science
Birsen Sirkeci, Ph.D.	Department of Electrical Engineering

ABSTRACT

ACOUSTIC CLOAK DESIGN USING GENERATIVE MODELING AND REINFORCEMENT LEARNING

by Linwei Zhuo

Metamaterials are engineered composites that can exhibit acoustic, electromagnetic, elasto-dynamic and mechanical properties that are not found in natural materials Due to the complexity of the target objective functions, it is difficult to find the globally optimized solutions in the inverse design of metameterials. This thesis proposes and outlines two model, a gradient-based optimization method combined with generative networks (2D-GLOnets) and a reinforcement learning (RL) model, that can find the optimized metamaterial structures across a wide range of parameters. By perturbing the positions of each cylindrical scatterer in a planar configuration, 2D-GLOnets and the RL model with Deep Deterministic Policy Gradients (DDPG) are developed to design 2D (two-dimensional) broadband acoustic cloaking devices at the desired range of wavenumbers. Both models were implemented using PyTorch, Python libraries, and MATLAB engine. The numerical results are presented and compared with the optimal values produced by *fmincon* algorithms to verify the validation of our approaches. Our results indicated that both methods reduced the total scattering cross section by 17% to 90% compared to the initial conditions.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr, Feruza Amirkulova, for her generous and endless support. I will not able to complete my thesis without her guidance. I would also like to thank Dr. Teng Moh and Dr. Birsen Sirkeci for giving me valuable feedback and ensuring my thesis satisfies the Graduate studies thesis requirements. Last but not least, I would like to express my special thank to my family for their support. I would have never been able to pursue my academic career and work through all the difficulties without their love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Multiple scattering theory studies the wave propagation and scattering in complex media. It is widely used in many engineering applications, such as vibration isolation, acoustic insulation, and sound focusing. For example, a vibration isolation system can attenuated undesired vibrations transmitted from road to the vehicle seats [1]. A sonic crystal barrier reduces sound in the target frequency range [2]. In work [3], acoustic Luneburg lens, i.e. a gradient index lens, focuses the incoming wave to achieve a maximum value on the edge of the lens. An acoustic cloaking device uses the multiple scattering theory to control and guide the incident waves around the objects. Such acoustic cloaking device can make an object invisible to the incoming wave [4]. There are two types of cloak: active and passive. A passive cloaking device utilizes metamaterials to direct wave energy around the object [5]. An active cloaking device uses sources to produce an active field that interferes the incoming wave [6].

The main challenge for cloaking design is to improve the accuracy and efficiency of the optimized cloaking devices at a broad range of the frequencies and letting it to operate at different frequencies. It is difficult and challenging to design and optimize the scattering devices because their target responses are complex and their objective functions are highly nonlinear and multimodal, which have multiple extremum and saddle points. There are many approaches that have been broadly applied in the inverse design of cloak, such as stochastic algorithms [7]–[9], gradients optimized algorithm [10] and topology optimization [11]. However, they are local optimizers, which can potentially stuck in an unfavorable local optimum. The performances of their final solutions heavily rely on the choice of the initial distribution. Devices tend to have better performance if their initial distributions happen to be

located in a desirable design space. Searching for the globally optima is difficult when the problem is complicated. Current methods [7]–[11] work well for designing simple acoustic cloak at a single wavenumber, but they struggle to optimize larger numbers of cloak devices at a broadband of wavenumbers. Therefore, a robust global optimizer that can handle complicated problems is highly demanded in the engineering industry and academy.

Recently, a variety of new methods have been developed for the inverse design of metamaterials. With the study of various new approaches, we identified the validation and potential of deep learning (DL) in the optimization problems. In our research, we develop a novel approach on the inverse design of metamaterials by using DL, RL, and generative modeling. A new proposed optimizer, namely GLOnet, is based on DCGAN can produce highly efficient optimized 1D (one-dimensional) metagrating over a wide range of parameters [12]. We generalized 1D GLOnets [12] to develop 2D-GLOnets with reparameterization for 2D acoustic problems at various range of wavenumbers and parameters. Compared to GLOnets, we make our model more general as it can be applied to constrained optimization problems and can be extended and deployed to different inverse designs of metamaterials, such as implementation of braodband acoustic, elastodynamic, optical, and electromagnetic meta-devices. Furthermore, our proposed generative model can generate highly efficient cloaking devices that can outperform the conventionally at certain scenarios. Our second model is RL model implemented with Deep Deterministic Policy Gradients (DDPG). Our RL model consists of an actor and a critic. The actor network outputs actions that continuously move the scatterers in our acoustic environment. The critic evaluates the value for each action selected by the actor. We trained our RL model to

discover the optimal policy that can search for the acoustic devices with improved cloaking efficiency with a set of given initial conditions at the desired range of wavenumbers. Our findings clearly show that 2D-GLOnets and the RL model can sufficiently optimize a metamaterial device and serve as a potential global optimizer.

2 LITERATURE REVIEW

In this section, we will review the various optimization methods deployed in the inverse design of metamaterial and the new trend of using DL, RL, and generative modeling in metamaterial optimization.

2.1 Optimization Methods in Metamaterial Design

Review of literature shows that stochastic algorithms [7]–[9], gradient-based optimization [4], [13], and topology optimization [14], [15] are the popular approaches used in the inverse design of metamaterials. The former methods, such as genetic algorithms [16] and simulated annealing [17], are gradient-free optimizers, but their converge time increases as the complexity of the model grows and the number of design parameters increases. The topology optimization [14], [15] utilizes numerical gradients to optimize materials with a given set of conditions. The gradient-based optimization method [4], [18] use many local optimizers to search for the possible solutions. These optimizers are either local optimizer or require great amount of computational resources to search for the global solutions.

2.2 Deep Learning in Metamaterial Design

2.2.1 Generative Modeling for Metamaterials

In recent years, we are experiencing the innovation driven by DL in computer science, physics, engineering, and art. Amoung various DL techniques, Generative Adversarial Network (GAN) becomes a promising method to generate new data with specific requirements. GAN is an architecture of DL model designed by Ian GoodFellow et al [19] in 2014. The main purpose of this network is to generate a new data set with the same statistics as the given training set. This network is widely used to generate fake faces [20] and repair

images [21]. Over the years of development, researchers have adopted and modified the architecture of GAN to apply on various problems [12], [22]–[30].

Gurbuz et al. [22] combined GAN and finite element simulation to design acoustic metamaterials for broadband sound insulation. Challapalli et al. [27] proposed a new inverse design framework using GAN to improve 40-120% of lattice unit cells' load carry capacity. Blanchard-Dionne and Martin [24] used GAN to search for the optimal geometry of a 2D optical cloak for specific property. A modified version of GAN called WGAN (Wasserstein GAN), which minimizes an approximation of the Earth-Mover's distance (EM) instead of the Jensen-Shannon divergence, is also applied to optimize metamaterial devices. Han et al. [31] proposed conditional WGAN to improve prediction performance of Deep Neural Network for protein solubility. An et al. [26] used conditional WGAN to design metasurfaces that produce double focusing effect.

Another modification of GAN is DCGAN (Deep Convolutional GAN) which replaces the pooling layers with the strided convolution and uses batch normalization layers in both generator and discriminator. A novel optimizer based on DCGAN, GLOnets, [12] can produce highly efficient optimized devices over a wide range of parameter. Hodge et al. [23] used DCGAN to degisn tensorial metasurfaces for radio-frequency (RF) applications. There are also different types GAN that are modified to apply to specific types of inverse design problems. For example, Hou et al. [28] used partially Conditional GAN (pCGAN) to predict the performance of sets of metamaterials by inputting the desired spectrum. Nobari et al. [29] used Performance Conditioned Diverse (PcDGAN) to design airfoils and enhance the conditioning likelihood by 69% in an airfoil generation task than the original GAN. Khodaee

et al. [32] used AIGAN/GAN to design terahertz bandpass filters. Variation Autoenconder (VAE) is also a type of generative modeling. It aims to learn the complicated data distribution and maximize the probability of generating new data in the same distribution. Similar to GAN, VAE is also a novel and promising optimization approach in inverse design. Wang et al. [30] used a variational autoencoder (VAE) and a regressor to design heterogeneous metamaterial systems to obtain the desired distortion behaviour. Tran et al. [33] used VAE to discover the optimal configurations with given TSCS at a discrete set of wavenumbers.

2.2.2 Reinforcement Learning in Inverse Design of Metamaterials

RL is a branch of machine learning concerned with how agent learn to take actions in an environment in order to find the optimal policy. Many reseachers used deep RL to increase the efficiency of electromagnetic properties in metamaterial design. Sajedian et al. [34] used a double deep Q-learning network (DDQN) to find geometric design and material properties to improve the transmission efficiency of metasurface holograms, a type of photonic structure. Their agent was able to enhance the the transmission efficiency of from 17% to 32%. Sajedian et al. [35] also applied DDQN to enhance the color generation of dielectric reflective color filters. Shah et al. [36] developed a RL model with DDPG and DDQN to design cloaking devices.

RL can also discover a new strategy to optimize materials' structures that have desired performance in aerodynamic and fluid mechanics. For example, Kim et al. [37] used deep RL to optimize airfoil shape with multiple predefined conditions for various flow conditions. Viquerat et al. [38] used deep RL to discover a generic shape optimization strategy in fluid mechanics and aerodynamics without any prior knowledge. Other researcher have modified

and developed RL to further improve its capability in solving various types of inverse design problems. One of them is to combine RL with GAN. For example, Guimaraes et al. [39] introduced a GAN and RL based method (ORGAN) that moves generation towards desirable regions. Putin et al. [40] introduced a method that combines RL and GAN to design of novel small-molecule organic structures.

We observed the functionalities that GAN and RL offered in designing inverse material and optimizing engineering structures. Currently, stochastic algorithms, gradient-based optimization, and topology optimization are insufficient to find the optimal cloaking devices that can suppress the scattering response at desired operating range of frequencies or certain conditions. Therefore, we want to leverage the power of DL into our 2D acoustic inverse design problems.

3 PRELIMINARIES

3.1 Acoustic Multiple Scattering

We consider the multiple scattering in the context of the two dimensional acoustic time harmonic wave equation. The governing equation for the acoustic pressure $p(\mathbf{x})$, $\mathbf{x} \in R^2$, is the Helmholtz equation [4]:

$$\nabla^2 p + k^2 p = 0, \tag{3.1}$$

where $k = \omega/c$ is the wavenumber, c is the acoustic speed, and ω is the frequency. The total field $p(\mathbf{x})$ is defined as the sum of incident p_{inc} and scattered p_{sc} pressure fields:

$$p = p_{inc} + p_{sc}. \tag{3.2}$$

The incident field is a plane wave which propagates from left to right and interacts with a given configuration of M separate scatterers. We consider circularly cylindrical scatterers, which may be either rigid scatterers.

3.1.1 Total Scattering Cross Section (TSCS)

Let σ denote the total scattering cross section (TSCS) defined as [4]:

$$\sigma(\mathbf{r}_{jm}, ka) = -\frac{4}{}\Re \mathbf{a}^{\dagger}\mathbf{b}, \tag{3.3}$$

where $\Re$ stands for real part of complex variable, $\mathbf{a}^{\dagger}$ is the complex conjugate transpose (Hermitian), the vectors **a** and **b** are defined by Equation 3.4 in Ref. [4],

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}^{(1)} \\ \mathbf{a}^{(2)} \\ \vdots \\ \mathbf{a}^{(M)} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}^{(1)} \\ \mathbf{b}^{(2)} \\ \vdots \\ \mathbf{b}^{(M)} \end{pmatrix}, \quad \mathbf{a}^{(j)} = \begin{pmatrix} A_{-N}^{(j)} \\ A_{-N+1}^{(j)} \\ \vdots \\ A_{N}^{(j)} \end{pmatrix}, \quad \mathbf{b}^{(j)} = \begin{pmatrix} B_{-N}^{(j)} \\ B_{-N+1}^{(j)} \\ \vdots \\ B_{N}^{(j)} \end{pmatrix}. \tag{3.4}$$

and $\mathbf{r}_{jm} = \mathbf{r}_j - \mathbf{r}_m$ is a position vector of multipole O_m with respect to multipole O_j depicted in Figure 3.1.

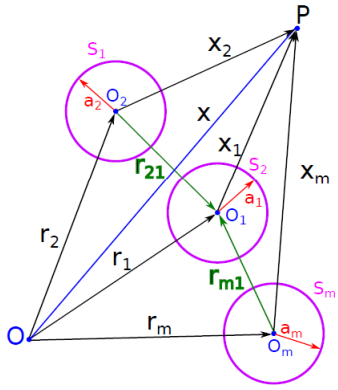


Figure 3.1 An arbitrary planar configuration of M scatterers S_m with a radius a_m, $m = 1, M$.

Our objective is to minimize the root mean square (RMS) of a set of TSCSs over some range of normalized wavenumbers $k_i a\,(i = 1, 2, \ldots, N_k)$, which is denoted as σ_{RMS}:

$$\sigma_{RMS}(\mathbf{r}_{jm}) = \left[\frac{1}{N_k} \sum_{i=1}^{N_k} \left(\sigma(k_i a_j, \mathbf{r}_{jm}) \right)^2 \right]^{1/2} \tag{3.5}$$

where j and $m = 1, 2, \ldots M,$ and M is the total number of scatterers.

For a single scatterer, σ is independent of the position of the scatterer. For a pair of scatterers, it depends on the relative positions $\mathbf{r}_{ij}$, i.e $\sigma = \sigma\ \mathbf{r}_{ij}\ = \sigma\ \mathbf{r}_i - \mathbf{r}_j$, where the vectors $\mathbf{r}_i$, $\mathbf{r}_j$ define the scatterer position vectors as shown in Figure 3.1.

3.1.2 Gradients of TSCS

Amirkulova and Norris [4] provided the gradients with respect to the scatterer positions as:

$$\mathbf{s}_j = \frac{\partial \sigma}{\partial \mathbf{r}_j}, \quad j = 1, 2, \ldots M. \tag{3.6}$$

where a_j is the radius of j-th cylindrical scatterer. By substituting Equation 3.3 into Equation 3.6, the gradient vector can be defined as [4]:

$$\mathbf{s}_j = -\frac{4}{k}\Re\left[\frac{\partial \mathbf{a}^{\dagger}}{\partial \mathbf{r}_j}\mathbf{b} + \mathbf{a}^{\dagger}\mathbb{X}^{-1}\left(\frac{\partial \mathbf{a}}{\partial \mathbf{r}_j} - \frac{\partial \mathbb{X}}{\partial \mathbf{r}_j}\mathbf{b}\right)\right], \tag{3.7}$$

where the interaction matrix X established the coupling between each scatterer of the configuration as;

$$\mathbb{X} = \begin{bmatrix} \mathbf{T}^{(1)^{-1}} & -\mathbf{P}^{1,2} & -\mathbf{P}^{1,3} & \ldots & -\mathbf{P}^{1,M} \\ -\mathbf{P}^{2,1} & \mathbf{T}^{(2)^{-1}} & -\mathbf{P}^{2,3} & \ldots & -\mathbf{P}^{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathbf{P}^{M,1} & -\mathbf{P}^{M,2} & -\mathbf{P}^{M,3} & \ldots & \mathbf{T}^{(M)^{-1}} \end{bmatrix} \tag{3.8}$$

For the RMS of a set of TSCSs over a range of k_{ia}, broadband gradient vectors $\mathbf{q}_j$ with respect to position vectors $\mathbf{r}_j$ in a closed-form is depicted as [4]:

$$\mathbf{q}_j = \frac{\partial \sigma_{RMS}(\mathbf{r}_{jm})}{\partial \mathbf{r}_j}, \tag{3.9}$$

where $j = 1, \ldots M$ and $m = 1, \ldots M$.

Thus, the explicit form of broadband gradients q_j are derived in terms of the individual single frequency gradients $\mathbf{s}_j$ [4] by substituting Equation 3.5 into Equation 3.9:

$$\mathbf{q}_j = \frac{1}{\sigma_{RMS}} \frac{1}{N_k} \left[\sigma(k_1 a)\mathbf{s}_j(k_1 a) + \sigma(k_2 a)\mathbf{s}_j(k_2 a) + \ldots + \sigma(k_{N_k} a)\mathbf{s}_j(k_{N_k} a) \right] \tag{3.10}$$

where $\mathbf{s}_j(k_i a)$ are evaluated at normalized wavenumbers $k_i a\,(i = 1, 2, \ldots, N_k)$ by using Equation 3.7.

3.2 Acoustic Cloak by Inverse Design

3.2.1 Problem Formulation

In our acoustic mutiple scattering problems, we only consider the scattering by planar configuration of rigid cylinders embedded in an external water medium. Our objective is to find the optimal configurations with the lowest σ_{RMS} by adjusting the positions of the scatterers. We discuss two different cases in this paper. Case 1 is that all the scatterers are free to move inside the circular region (see Figure 4.3). Case 2 is that three cylinders are fixed inside the annulus and are being cloaked by adding scatterers within the cloaking annulus region (see Figure 4.9). Both cases use Equation 3.5 and Equation 3.9 to evaluate the metrics of the optimized configurations.

3.2.2 Geometric Constraints

The constrained optimization problem involves bounds and nonlinear constraints.The nonlinear constraints on a configuration of scatterers are as follows:

1. The positions of cylinders are located within a regions with a radius R_{out} and outside of an inner annulus with a radius R_{in}. Thus,

$$R_{in} < r_k < R_{out} \tag{3.11}$$

where $r_k = |\mathbf{r}_k|$ and $\mathbf{r}_k$ are the k-th scatterer's position vector as described in Figure 3.1.

2. In order to avoid overlapping, the distances between the centers of cylinders are constrained by

$$|\mathbf{r}_i - \mathbf{r}_j| > 2a + \delta \tag{3.12}$$

where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the position vectors of i-th and j-th scatterer, a is the radius of the scatterer, and δ is the minimal allowed distance between the scatterers.

4 GENERATIVE MODELING AND GRADIENT-BASED OPTIMIZATION

4.1 2D-GLOnets Architecture and Overview

The architecture of 2D-GLOnets consists of 8 fully connected networks and uses the *LeakyReLu* for all activation functions, except for the output layer, which uses the *tanh* activation function. Since the *tanh* function always output values between -1 to 1, we simply multiply the output by a scale to adjust the range of output. The input is a vector of random numbers sampled from a noise and has a size of $[K \times M]$, where K is the number of outputs in a batch and M corresponds to the total number of scatterers. The output size is the same as the input size. Figure 4.1 illustrates the process of 2D-GLOnets. A noise vector Z is sampled from the normal Gaussian distribution. The generator takes the noise vector Z as an input and outputs a batch of K unconstrained devices X, where X is in the form:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{y}_1 \\ \mathbf{x}_2 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{x}_k \\ \mathbf{y}_k \end{pmatrix} \tag{4.1}$$

The vertical and horizontal positions of k-th device x with respect to the origin is denoted as x_k and y_k, respectively.

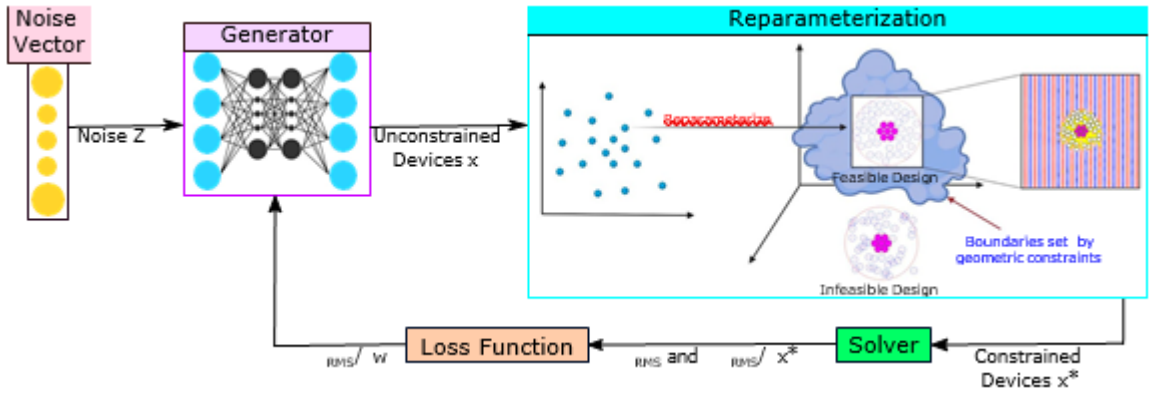


Figure 4.1 Flowchart of 2D-GLOnets with reparameterization.

A method called reparamerization enforces the geometric constraints imposed on cyliders' positions as described by Equation 3.11 and Equation 3.12, and transforms the input X into the constrained devices $X*$. The multiple scattering solver [41] denoted as MS Solver computes σ_{RMS} and $\frac{\partial \sigma RMS}{\partial \mathbf{w}}$. The loss function defined in Equation 4.9 calculates the gradients $\frac{\partial \sigma RMS}{\partial \mathbf{w}}$ from back propagation and updates the generator's weights **w**.

4.2 Reparameterization

Algorithm 1: Procedures of reparameterization

while *not constrained* **do**
  for *device* X^i *in devices* X **do**
    $X^i = x_i, y_i$
    $r_i = \sqrt{x_i^2 + y_i^2}$
    if $B(r_i)$ **then**
      $r_i = r_{out} - a$ if $r_k \geq r_{out}$
      $r_i = r_{in} + a$ if $r_k \leq r_{in}$
    for *other devices* x^j *in devices* X **do**
      $X^j = x_j, y_j$
      $r_j = \sqrt{x_j^2 + y_j^2}$
      $r_{ij} = r_i - r_j$
      $\theta(r_{ij}) = atan2(x_i, y_i, x_j, y_j)$
      if $C(r_i, r_j)$ **then**
        $x_i^* = (2a + \delta) - r_{ij} \cdot cos(\theta(r_{ij}))$
        $y_i^* = (2a + \delta) - r_{ij} \cdot sin(\theta(r_{ij}))$
        $X^{i*} = x_i^*, y_i^*$
  if *all constrains are satisfied* **then**
    break

One method to impose geometric constrain is to add a penalty term in the loss function [42]. This method helps to move devices toward desired design region where geometric constrains are enforced. However, the penalization approach does not assure the enforcement. Therefore, we want to introduce a new method to apply geometric constrains, i.e. 2D reparameterization. The reparameterization denoted as $\varepsilon(x^{(k)})$ is a method that uses a mapping function to transform the illegal devices into constrained design space [43]. Let's define the boundary state, $B(r_k)$ as:

$$B(r_k) = \begin{cases} 1, & \text{if k-th device violates the constraint as defined in Equation 11} \\ 0, & \text{else.} \end{cases} \tag{4.2}$$

Similarly, we define the collision state, $C(r_i, r_j)$ as:

$$C(r_i, r_j) = \begin{cases} 1, & \text{if i-th and j-th device are overlapped as defined in Equation 12} \\ 0, & \text{else.} \end{cases} \tag{4.3}$$

During the process, the position vectors of each device will be updated if they violate the geometric constrains as described in Algorithm 1. This cycle will continues until all constrains are satisfied eventually and return the constrained devices $X*$.

4.3 Loss Function

We frame our objective of the optimization as maximizing the probability of generating the globally optimized device in design space S. Following Jiang and Fan [44], we define the probability function as:

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \int_S \delta(\sigma_{RMS}(X) - \sigma_{RMSmin}) \cdot P_\phi(x)\, dX. \tag{4.4}$$

The derivative of the δ function is nearly zero, thus the δ function can be expressed as a Gaussian approximation [45]:

$$\delta(\sigma_{RMS}(X) - \sigma_{RMSmin}) = \lim_{\beta \to 0} \frac{1}{\sqrt{\pi}\beta} \exp\left[-\left(\frac{\delta(\sigma_{RMS}(X) - \sigma_{RMSmin})}{\beta} \right)^2 \right] \tag{4.5}$$

By substituting Equation 4.4 into Equation 4.5 and leaving β as a tunable parameter, we obtain the new equation as:

$$\phi^* = \underset{\phi}{\operatorname{argmax}} \int_S \exp\left[-\left(\frac{\delta(\sigma_{RMS}(X) - \sigma_{RMSmin})}{\beta} \right)^2 \right] \cdot P_\phi(X)\, dX. \qquad (18) \tag{4.6}$$

where β is an important parameter that stabilizing the model with a finite batch size during training. Note that $P_\varphi(X \,|\sigma_{RMS}(X) < \sigma_{RMSmin}) = 0$. Furthermore, we want to find the argument that give us the maximum probability of generating device. Hence, we can approximate Equation 4.6 with an exponential equation [44] as:

$$\phi^* := \underset{\phi}{\operatorname{argmax}} \int_s \exp\left[-\left(\frac{\sigma_{RMS}(X^{(k)}) - \sigma_{RMSmin}}{\beta} \right) \right] \cdot P_\phi(X)\, dX. \tag{4.7}$$

The objective function depends on σ_{RMSmin}, which is unknown. An ideal acoustic device has a σ_{RMS} of zero. We can assume that the global minimum σ_{RMSmin} is equal to zero. In practice, it is difficult to evaluate the whole design space S. Jiang and Fan [44] suggested to sample a batches of K devices $\left\{X^{(k)}\right\}_{k=1}^{K}$ from P_φ.

We note that the σ_{RMS} of device X is calculated using in-home built multiple scattering solver [41] (MS Solver) implemented on MATLAB, such that σ_{RMS} is not directly differentiable for backpropagation. To overcome this issue, our MATLAB solver [41] computes the gradients with respect to the position for device X: $g = \frac{\partial \sigma RMS}{\partial X}$ as Equation 3.10.

We note that q_j in Equation 3.10 is calculated for single scatterer and g is a list of the q_j for a device with multiple scatterers. Finally, we can use the objective function and gradients to define our loss function $L = L(X, g, \sigma_{RMS})$ such that minimizing L is equivalent to minimize the objective function. With this definition, L must satisfy $\frac{\partial L}{\partial X^{(k)}} = \frac{1}{K}\frac{\partial}{\partial X^{(k)}}\left(\frac{-\sigma_{RMS}(X^{(k)})}{\beta}\right)$ and is defined as:

$$L(X, g, \sigma_{RMS}) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\beta}\exp\left(\frac{-\sigma_{RMS}(X^{(k)})}{\beta}\right)X^{(k)} \cdot g^{(k)} \tag{4.8}$$

We also need to include the reparamerization process to impose the geometric constraints in the loss function. Note that $X^{*(k)}$ is the constrained device and $g^{(k)} = \frac{\partial \sigma}{\partial}$.

Thus, we can further define our final loss function as:

$$L(X, g_\varepsilon, \sigma_{RMS}) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{\beta}\exp\left(\frac{-\sigma_{RMS}(x^{(k)})}{\beta}\right) \cdot X^{*(k)} \cdot g_\varepsilon^{(k)} \tag{4.9}$$

4.4 Training Process

Algorithm 2: Training Process of 2D-GLOnets

Parameters: α, learning rate. ϕ, generator parameters.
initialization
while $i <$ *total iterations* **do**
  sample $\{z^k\}_{k=1}^{K} \sim \mathscr{U}(0, 0.2)$
  generate $\{x^k = G_\phi(z^k)\}_{k=1}^{K}$
  reparameterize $\{x^* = \varepsilon(x^k)\}_{k=1}^{K}$
  compute $\{g_\varepsilon^k\}_{k=1}^{K}$, $\{\sigma_{RMS}^k\}_{k=1}^{K}$
  update $\phi \leftarrow \phi + \alpha \cdot \text{Adam}\,(\phi, g_\varepsilon^k, \sigma_{RMS}^k)$
end

The training process of 2D-GLOnets is shown in Algorithm 2. First, a noise vector Z is sampled from a normal distributed noise and passed into the generator G_φ, which outputs a batches of K unconstrained devices X. The reparameterization maps the illegal devices X into feasible devices X^*. The in-built MATLAB solver [41] computes σ_{RMS} as Equation 3.5 and the corresponding gradients g_ε with respect to the constrained devices X^* as Equation 3.9. Adaptive Moment Estimation (ADAM) is used to update the generator's weights. The hyper-parameters of 2D-GLOnets can be refer to Appendix A.1.

4.5 Model Evaluation for 2D-GLOnets

The *fmincon* is a gradient-based optimization solver that aims to search for the minimum of an nonlinear multi-variable function [46]. To validate the performance of 2D-GLOnets, we evaluated σ_{RMS} of the devices discovered by 2D-GLOnets and compared the results with *fmincon* obtained using the state-of-the-art algorithms, such as sequential quadratic (SQP), SQP-legacy, Interior-point, and Active-set. Since our 2D-GLOnets generated 10 devices at each batch, we considered 10 initial configurations distributed over the range of the possible design space with *MultiStarts*. *MultiStarts* is a MATLAB solver that allows *fmincon* to start with multiple initial configurations. Both *fmincon* and 2D-GLOnets ran 10 attempts and we

chose the results with the lowest σ_{RMS} for each methods. The simulations were performed remotely on Dell Workstation for $2 \leq M \leq 6$ and Docker system for $12 \leq M \leq 24$. We note that 2D-GLOnets use random noise as input to generate the devices. The initial configurations were randomly generated and only used by *fmincon*. Hence, the evaluation of 2D-GLOnets and the initial scatterers positions might not be accurate as 2D-GLOnets did not start from the same initial configurations.

4.6 Results for 2D-GLOnets

In this section, we present the results obtained from 2D-GLOnets and *fmincon*. 2D-GLOnets were implemented using PyTorch by calling MATLAB engines from Python libraries, where *ka* is the non-dimensional wavenumber and σ_{RMS} is the root mean square of the total scattering cross section at a range of wavenumbers. Table 4.1 illustrates the performance of the devices generated by 2D-GLOnet and *fmincon* for M = 2, 3, 4, 5, 6, 12, 18, and 24 at $0.35 \leq ka \leq 0.45$. According to Table 4.1, 2D-GLOnets generated devices with a σ_{RMS} of 0.2202, 0.3031, 0.3196, 0.3366, 0.3048, 0.6396, 0.9998, and 1.7093 for M = 2, 3, 4, 5, 6, 12, 18, and 24, respectively. 2D-GLOnets and all *fmincon's* algorithm yielded the same σ_{RMS} for $2 \leq M \leq 6$, except for SQP-legacy, which generated a slightly higher σ_{RMS} for M = 6. Compared to the initial conditions, 2D-GLOnets reduced the σ_{RMS} by 75.72%, 78.02%, 73.08%, 86.51%, 88.97%, 82.31%, 84.60%, and 80.31% for M = 2, 3, 4, 5, 6, 12, 18, and 24, respectively; however, 2D-GLOnets produced 44.2%, 33.4%, and 71.7% higher σ_{RMS} than the Active-set for M = 12, 18, and 24, respectively.

Table 4.1 Comparison of σ_{RMS} between Initial Conditions (ICs), 2D-GLOnets (GLO), Active-set (A-set), Interior-point (INT), SQP, and SQP-legacy (SQPL) at $0.35 \leq ka \leq 0.45$.

$0.35 \leq ka \leq 0.45$								
	M = 2	M = 3	M = 4	M = 5	M = 6	M = 12	M = 18	M = 24
ICs	0.9071	1.3792	1.8173	2.4958	2.7629	3.6149	6.4931	8.6799
GLO	0.2202	0.3031	0.3196	0.3366	0.3048	0.6396	0.9998	1.7093
A-set	0.2202	0.3031	0.3196	0.3366	0.3048	0.4436	0.7493	0.9954
INT	0.2202	0.3031	0.3196	0.3366	0.3048	0.5568	0.8404	1.0953
SQP	0.2202	0.3031	0.3196	0.3366	0.3048	0.4436	0.8367	1.0164
SQPL	0.2202	0.3031	0.3196	0.3366	0.3671	0.4436	0.7663	1.1138

The performance of 2D-GLOnets and *fmincon* is visualized by plotting the variation of σ versus ka at the desired range of wavenumbers $ka \in [0.35,0.45]$ as shown in Figure 4.2. For M = 2, 3, 4, and 5, 2D-GLOnets have same σ as *fmincon* at every ka from 0.35 to 0.45 as shown in Figure 4.2(a) to (d). For $M = 6$, 2D-GLOnets have lower σ than SQP-legacy at $0.36 \leq ka \leq 0.45$ as shown in Figure 4.2(e). For M = 12, 2D-GLOnets have a smaller σ than Interior point at $0.35 \leq ka \leq 0.36$ as shown in Figure 4.2(f). For M = 18, 2D-GLOnets have the same σ_{RMS} as *fmincon* at $0.375 \leq ka \leq 0.385$ as shown in Figure 4.2(g). For $M = 24$, 2D-GLOnets greatly suppressed σ at all wavenumbers from 0.35 to 0.45 compared to the initial conditions as shown in Figure 4.2(h). Furthermore, the devices generated by *fmincon* and 2D-GLOnets exhibit a small fluctuate or increment in σ as ka increases compared to the initial conditions.

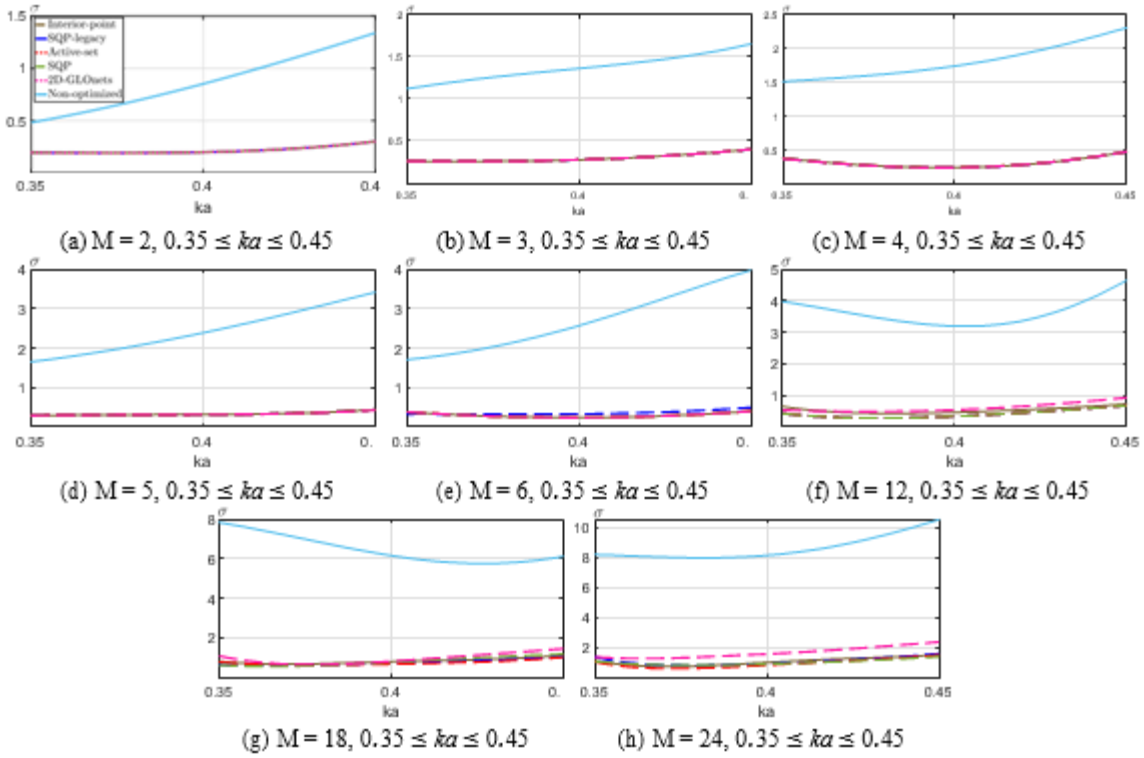
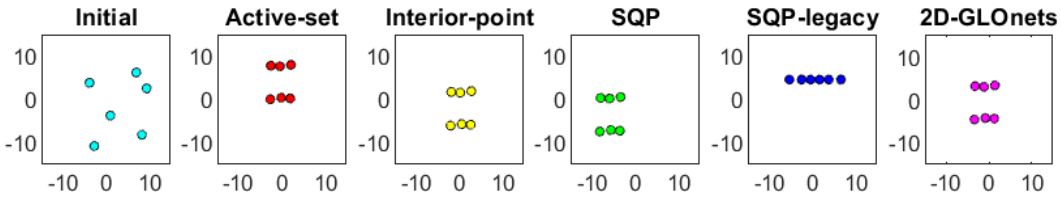
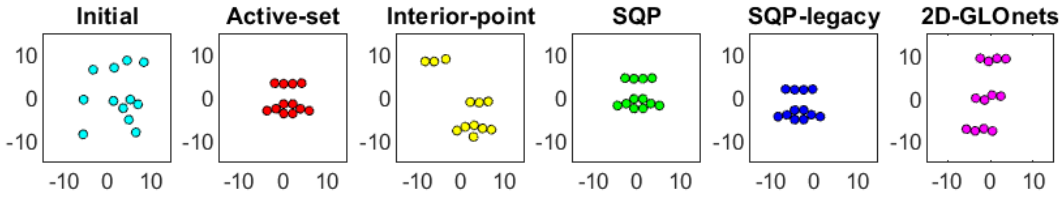


Figure 4.2 Variation of the total scattering cross section σ versus non-dimensional wavenumber ka at $0.35 \leq ka \leq 0.45$.

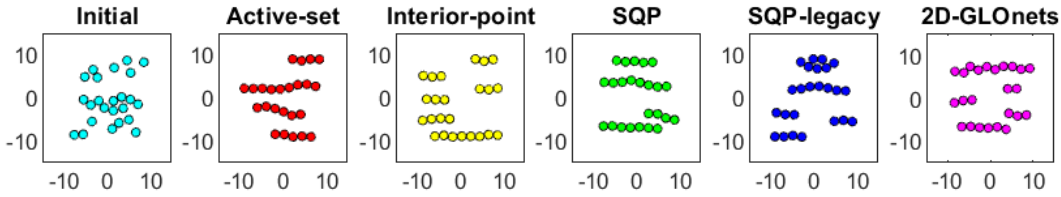
Figure 4.3 displays the devices generated by 2D-GLOnets and *fmincon* for $M = 6$, 12, and 24 at $0.35 \leq ka \leq 0.45$. 2D-GLOnets, Active-set, Interior-point, and SQP yielded the similar pattern for $M = 6$ as shown in Figure 4.3(a). Active-set, SQP, and SQP-legacy agreed on the optimal configuration for $M = 12$, whereas 2D-GLOnets and Interior-point produced different results with a higher σ_{RMS} as shown in Figure 4.3(b). The devices produced by 2D-GLOnets and *fmincon* are different for $M = 24$ as shown in Figure 4.3(c). It indicates that 2D-GLOnets and *fmincon* got stuck in different local optima. Furthermore, 2D-GLOnets found the strategy to suppress σ_{RMS} at $0.35 \leq ka \leq 0.45$, which is aligning scatterers horizontally at multiple rows. 2D-GLOnets' strategy is similar to *fmincon's* strategy as shown in Figure 4.3. The configurations for $M = 2$, 3, 4, 5, and 18 are shown in Appendix B.1.



(a) M=6



(b) M=12



(c) M=24

Figure 4.3 Optimized configurations generated by Active-set (red), Interior-point (yellow), SQP (green), SQP-legacy (blue), and 2D-GLOnets (pink) at $0.3 \leq ka \leq 0.45$. The initial configurations are shown in cyan.

Table 4.2 shows the σ_{RMS} of the devices discovered by 2D-GLOnets and *fmincon* for M = 2, 3, 4, 5, 6, 12, 18, and 24 at $1.00 \leq ka \leq 1.20$. Table 4.2 indicates that 2D-GLOnets output solutions with a σ_{RMS} of 2.7046, 3.0670, 2.8006, 3.3107, 3.7822, 8.2325, 10.8137, and 11.7097 for M = 2, 3, 4, 5, 6, 12, 18, and 24 at $1.00 \leq ka \leq 1.20$, respectively. 2D-GLOnets have the same results as *fmincon* for M = 2 and 3 and produced lower σ_{RMS} than Active-set and SQP-legacy for M = 4, 5, and 6. 2D-GLOnets also outperformed Interior-point for M = 5 and 6, and defeated SQP for M = 6. However, 2D-GLOnets started to struggle to minimize TSCS for M = 12, 18, and 24. All *fmincon* algorithms generated better devices than 2D-GLOnets for M = 12, 18, and 24. Compared to the initial results of TSCS for the initial configurations,

Table 4.2 Comparison of σ_{RMS} between *fmincon* and 2D-GLOnets at $1.00 \leq ka \leq 1.20$

$1.00 \leq ka \leq 1.20$								
	M = 2	M = 3	M = 4	M = 5	M = 6	M = 12	M = 18	M = 24
ICs	4.0284	6.0960	9.5414	10.8903	13.3013	23.1923	31.9988	36.6813
GLO	2.7046	3.0670	2.8006	3.3107	3.7822	8.2353	10.8137	11.7097
A-Set	2.7046	3.0670	3.8500	3.8828	4.0718	6.6489	7.9815	9.8253
INT	2.7046	3.0670	2.8006	3.3107	3.8692	5.7501	7.4785	8.2731
SQP	2.7046	3.0670	2.8006	3.8828	3.8442	6.9226	8.0514	9.5335
SQPL	2.7046	3.0670	4.2610	3.8828	4.4393	5.8122	8.4721	9.5325

2D-GLOnets decreased the σ_{RMS} by 32.86%, 49.69%, 70.65%, 69.60%, 71.57%, 64.49%, 66.21%, and 68.08% for M = 2, 3, 4, 5, 6, 12, 18, and 24, respectively.

The variation of σ versus *ka* for the solutions generated by 2D-GLOnets and *fmincon* is shown in Figure 4.4. Figure 4.4 shows that 2D-GLOnets can sufficiently suppress σ at higher range of wavenumbers $ka \in [1,1.2]$. For $M = 2$ and 3, all algorithms have the same performance. For $M = 4$ and 5, Active-set and SQP-legacy got trapped in the same local minimums and their σ are higher than 2D-GLOnets in every *ka* from 1.0 to 1.2. For $M \geq 6$, all algorithms found different solutions as the objective function exhibits many local optimal and difficult to optimize. Some results might have lower σ than other algorithms in certain range of *ka*, but have higher σ at other range as shown in Figure 4.4(e) to (h). For example, 2D-GLOnets has the smallest σ from $1.04 \leq ka \leq 1.11$ for $M = 6$, but SQP has the smallest σ from $1.12 \leq ka \leq 1.18$.

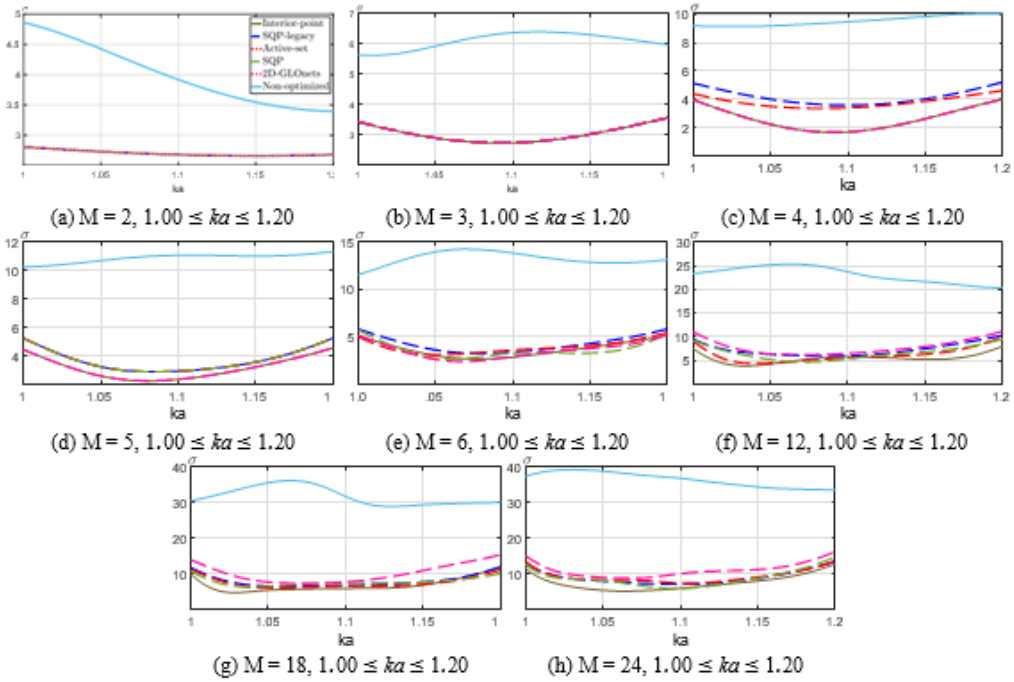
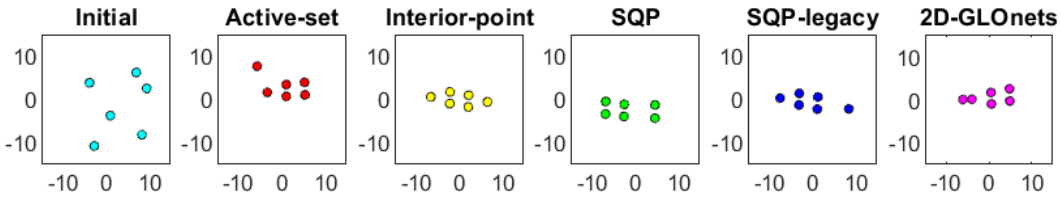
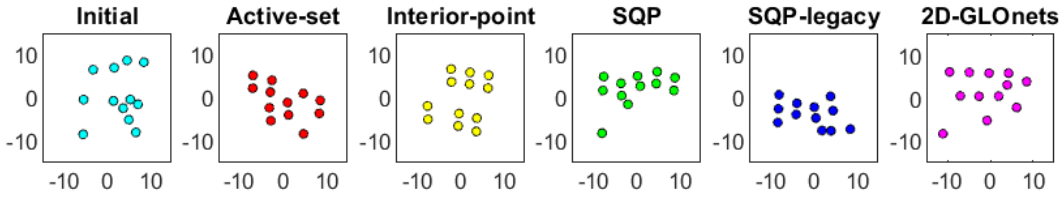


Figure 4.4 Variation of the total scattering cross section, σ , versus non-dimensional wavenumber *ka* at $1.00 \leq ka \leq 1.20$.

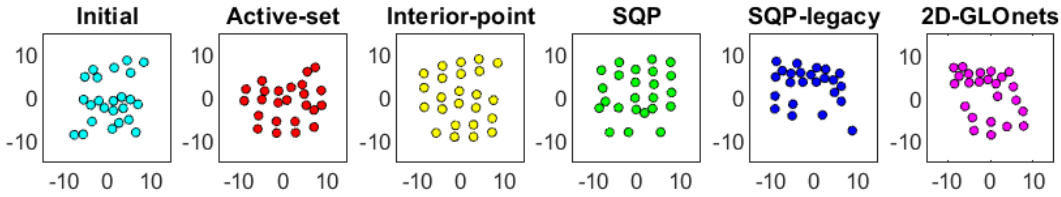
Figure 4.5 displays the devices generated by 2D-GLOnets and *fmincon* for $M = 6$, 12, and 24 at $1.00 \leq ka \leq 1.20$. For $M = 6$, 2D-GLOnets outperformed all *fmincon* algorithms and found the unique patterns with the lowest σ_{RMS} of 3.7822 as shown in Figure 4.3(a). For $M =$ 18 and 24, the configurations have no obvious patterns. SQP-legacy's and 2D-GLOnets' scatterers did not aligned or formed a particular shape. Instead, they were scattered over the design space as shown in 4(b) and (c). However, their configurations produced by 2D-GLOnets show a great reduction of σ_{RMS} compared to the initial configurations. In contrast, Interior-point placed scatterers vertically at multiple columns as it has the lowest σ_{RMS} of 7.4785 and 8.2731 for $M = 12$ and 24, respectively. The configurations for $M = 2$, 3, 4, 5, and 18 are shown in Appendix B.2.



(a) M=6



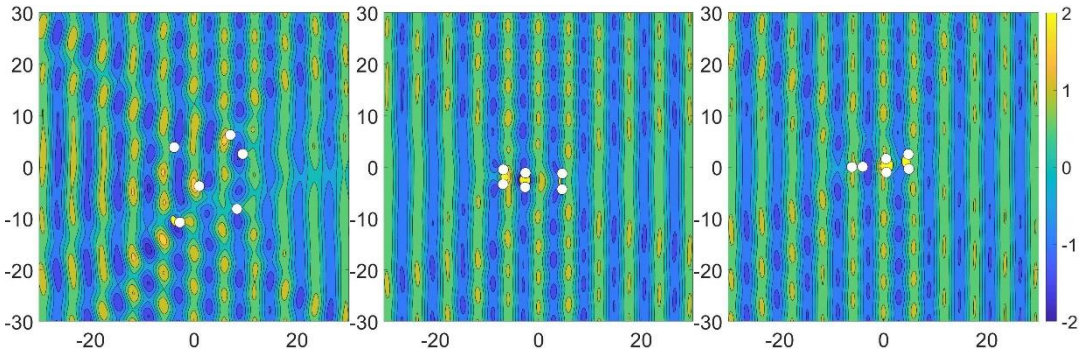
(b) M=12



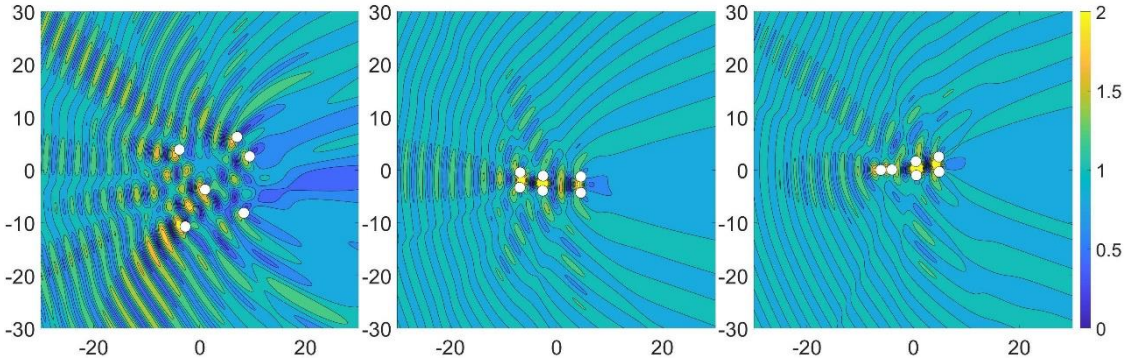
(c) M=24

Figure 4.5 Optimal configurations of rigid cylinders discovered by Active-set, Interior-point, SQP, SQP-legacy, and 2D-GLOnets at $1.00 \leq ka \leq 1.20$ and their initial configurations.

Figure 4.6 illustrates the comparison between the non-optimized, SQP's, and 2D-GLOnets' device for the total pressure field. Figure 4.6(a) displays the real part of acoustic total pressure distribution denoted as *Rep* and Figure 4.6(b) depicts the absolute part of the total pressure denoted as $|p|$. The incident wave propagates from left to the right horizontally and causes the interaction between the scatterers and the scattering response as shown in the total pressure field. The high pressure is colored in yellow and the low pressure is colored in dark blue as shown in Figure 4.6(a). The initial configuration has many high pressure areas in front of the scatterers in Figure 4.6(a). SQP performed better at suppressing the *Rep* on the front of



(a) The real part of total acoustic pressure field (*Rep*)



(b) The absolute part of total pressure ($|p|$)

Figure 4.6 The real part of total acoustic pressure field and the absolute total pressure at a normalized wavenumber $ka = 1.20$ for the initial configurations (Left), SQP (Middle), and 2D-GLOnets (Right) shown in Figure 4.5(a).

the scatterers and 2D-GLOnets better reduced the *Rep* on the rear of the cylinders configuration as shown in Figure 4.6(a). According to Figure 4.6(b), the initial configuration shows many regions with high $|p|$ in the front of the scatterers. The configurations discovered by SQP and 2D-GLOnets both suppressed these high pressure regions as shown in Figure 4.6(b).

The performance of optimized devices were evaluated over a larger range of wavenumbers from 0.10 to 2.00 as shown in Figure 4.7. Note that the devices were optimized only at the target range $ka \in [0.35,0.45]$ and $ka \in [1.00,1.20]$, but performance is evaluated at

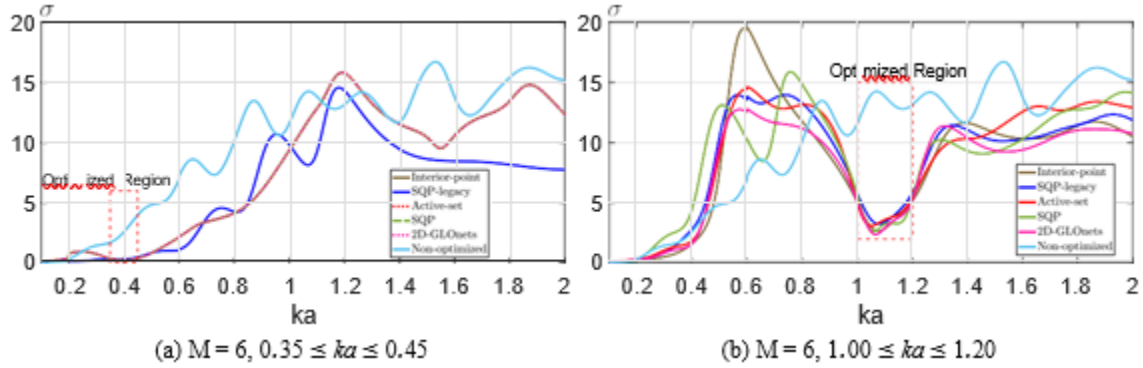


Figure 4.7 Variation of σ_{RMS} versus non-dimensional wavenumber ka for the optimized configurations when $M = 6$ rigid cylinders at $0.10 \leq ka \leq 2.00$.

a larger range $ka \in [0.1,2]$. The devices optimized at $0.35 \leq ka \leq 0.45$ by all algorithms can further suppress σ_{RMS} in larger ka compared to the non-optimized device as shown in Figure 4.7(a). SQP-legacy outperformed the other four algorithms for $ka \leq 0.2$ and $1.00 \leq ka \leq 2.00$ as shown in Figure 4.7(a). Figure 4.7(b) demonstrates that the devices optimized at $1.00 \leq ka \leq 1.20$ by all algorithms greatly suppressed σ_{RMS} at the desired range of wavenumbers $ka \in [1.00,1.20]$ compared to the non-optimized device. When outside of the desired range, 2D-GLOnets' optimized device has a better suppression of σ compared to *fmincon* at $1.60 \leq ka \leq 2.00$ as shown in Figure 4.7(b). Additionally, 2D-GLOnets produced lower σ than Active-set, Interior-point, and SQP-legacy at $0.48 \leq ka \leq 0.8$, although all optimized devices have higher σ than the initial configurations as shown in Figure 4.7(b).

Next, we will discuss Case 2 where the initial configuration has three scatterers M_{fixed} fixed inside an annulus region. We can suppress the σ_{RMS} by adding multiple scatterers on the proper positions around the fixed scatterers. According to Table 4.3, the initial configuration had a σ_{RMS} of 4.9400. 2D-GLOnets reduced the σ_{RMS} to 0.7790, 0.7057, 0.8405, 1.1327, and 1.9610 by adding 2, 4, 6, 12, and 18 scatterers, respectively. 2D-GLOnets have the best

Table 4.3 Comparison of σ_{RMS} between *fmincon* and 2D-GLOnets with three fixed scatterers at $0.35 \leq ka \leq 0.45$

$M_{fixed} = 3$ *and* $0.35 \leq ka \leq 0.45$							
Algorithm	$M_o = 0$	$M_o = 2$	$M_o = 4$	$M_o = 6$	$M_o = 12$	$M_o = 18$	$M_o = 24$
2D-GLOnets	2.4199	0.7790	0.7057	0.8405	1.1327	1.9610	2.8835
Active-set	2.4199	0.7790	0.8411	0.8974	1.1868	1.3097	1.7458
Interior-point	2.4199	0.7790	1.0211	1.0614	1.2680	1.6557	1.8105
SQP	2.4199	0.7790	0.8132	1.0329	1.1763	1.3139	1.7783
SQP-legacy	2.4199	0.7790	0.7050	0.8411	1.2037	1.5596	1.8725

results among all the algorithms for M_o = 6 and 12, the second lowest σ_{RMS} for M_o = 4, and same results as *fmincon* for M_o = 2; For M_o = 18, 2D-GLOnets have higher σ_{RMS} than *fmincon*; For M_o = 24, 2D-GLOnets failed to reduce the σ_{RMS} compared to the initial configurations as it produced higher σ_{RMS}.

Figure 4.8 shows the performance of 2D-GLOnets and *fmincon* with three fixed scatterer inside the annulus at every *ka* from 0.35 to 0.45. 2D-GLOnets have lower σ at every *ka* from 0.35 to 0.45 compared to the initial configurations for M_o = 2, 4, 6, and 12 as shown in Figure 4.8(a), (b), (c), and (d). For M_o = 2, all algorithms have the same curve as they yielded the same σ_{RMS} as shown in Figure 4.8(a). For M_o = 4, the curve of 2D-GLOnets almost overlapped with SQP-legacy as they only have a small difference in σ_{RMS} and 2D-GLOnets defeat Active-set, Interior-point, and SQP in every *ka* from 0.36 to 0.45 as shown in Figure 4.8(b); For M_o = 6, 2D-GLOnets have the lowest σ at $0.35 \leq ka \leq 0.36$ while mantaining a relatively low σ at other *ka* as shown in Figure 4.8(c). For M_o = 12, 2D-GLOnets have the lowest σ at $0.425 \leq ka \leq 0.45$ as shown in Figure 4.8(d). For M_o = 18 and 24, 2D-GLOnets do

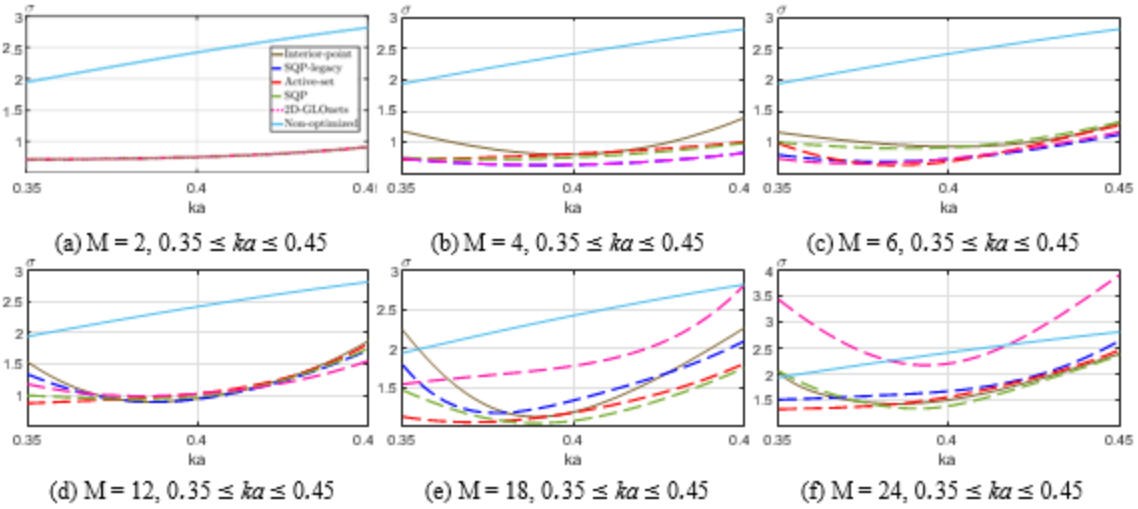
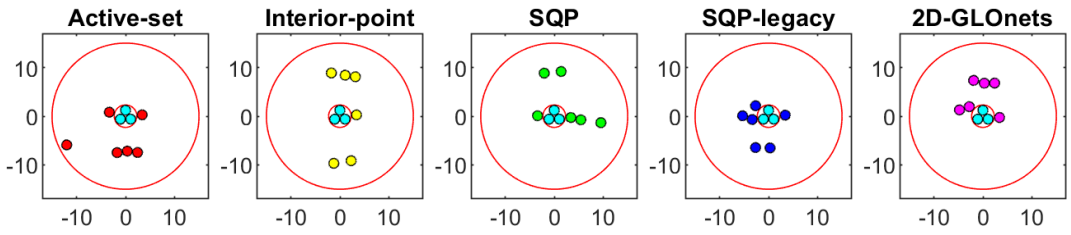


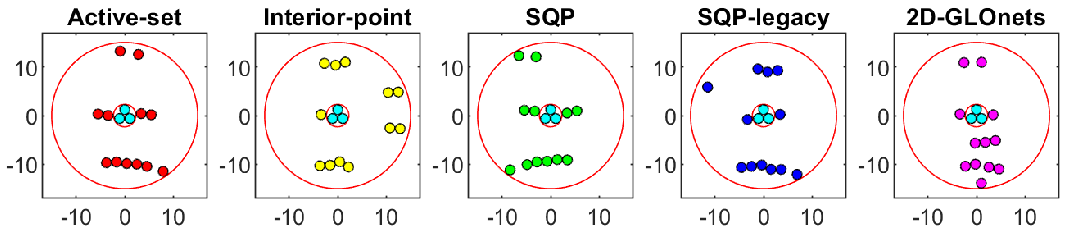
Figure 4.8 Variation of σ versus non-dimensional wavenumber *ka* with three fixed scatterers at $0.35 \leq ka \leq 0.45$.

not sufficiently reduce the σ at every *ka* compared to the initial configurations as shown in Figure 4.8(e) and (f).

Figure 4.9 displays the various configuration generated by 2D-GLOnets and *fmincon* with three fixed scatterers inside the annulus region for $M = 6$ and 12 at $0.35 \leq ka \leq 0.45$. For M_o = 6, 2D-GLOnets discovered the configuration with the smallest σ_{RMS} of 0.8405 as shown in Figure 4.9(a). Furthermore, SQP-legacy's configuration with a σ_{RMS} of 0.8411 is different from 2D-GLOnets, but their σ_{RMS} have a small difference, showing that there are multiple local optima with a similar performance when we increase the number of scatterers. For M_o = 12, 2D-GLOnets' solution also have the smallest σ_{RMS} of 1.1327 and its configuration is different from *fmincon* as shown in Figure 4.9(b). We can see that 2D-GLOnets also learned to align the scatterers horizontally at multiple rows to suppress the scattering response when we added three fixed scatterers at the center. The configurations for M_o = 2, 4, 18, and 24 are shown in Appendix B.3.



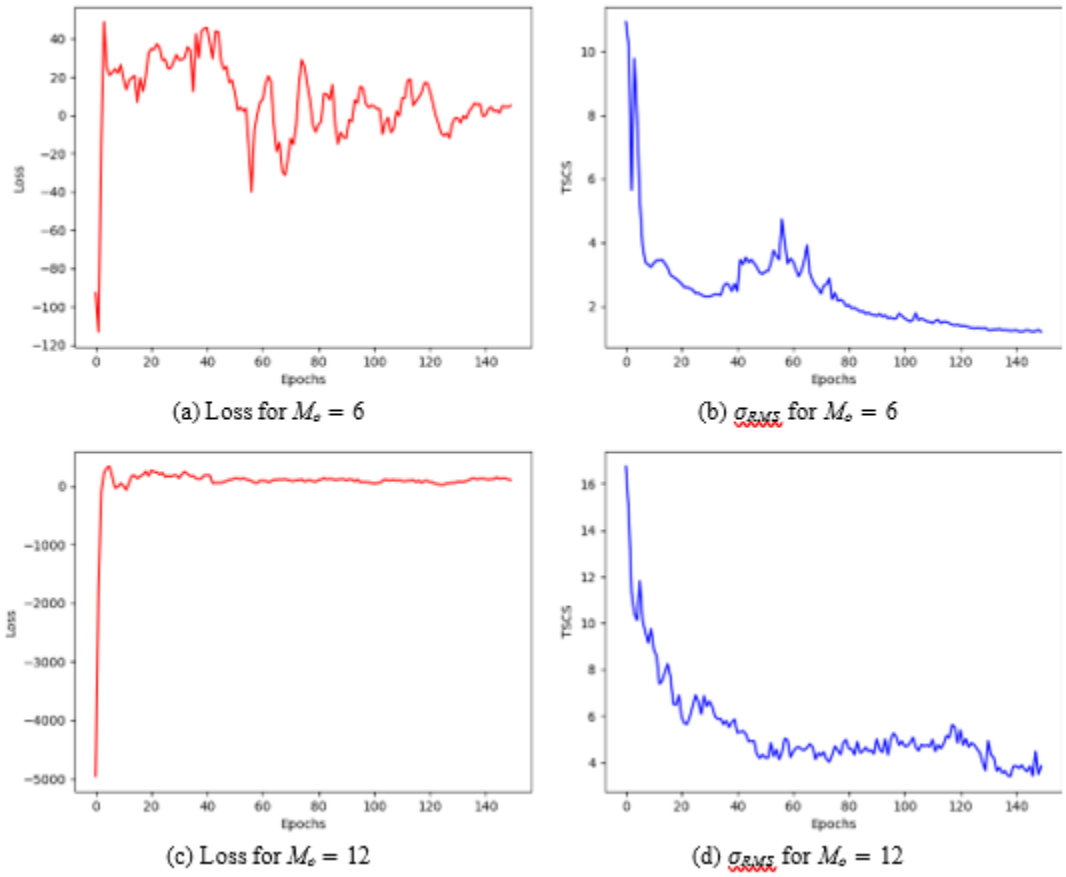
(a) M_o =6



(b) M_o =12

Figure 4.9 Optimized configurations discovered by fmincon and 2D-GLOnets with three fixed rigid cylinders inside an annulus for M_o = 2, 6, and 12.

Figure 4.10 shows the loss and σ_{RMS} versus epochs for M_o = 12 and 24. Unlike the traditional loss function where we want our loss as small as possible, we provided analytical gradients g_ε to guide our defined loss function defined in Equation 4.9 toward the ideal devices, which has zero σ_{RMS} and zero gradients. In other words, we want our loss function converges to zero. The loss functions for M_o = 12 and 24 were negatively large at the first epoch. They indicates that our gradients were also negatively large initially and the devices were far away from the optimal region. Hence, the gradients guided the devices rapidly toward the positive direction from epoch 0 to 10 as shown in Figure 4.10(a) and (c). In Figure 4.10(b) and (d), there were a huge reduction in TSCS from epoch 0 to 20 as the loss function moves toward zero. In Figure 4.10(a), the loss function shows that the generator was searching for the optimal solution at a particular region and landed into another design region



(a) Loss for $M_o = 6$ (b) σ_{RMS} for $M_o = 6$

(c) Loss for $M_o = 12$ (d) σ_{RMS} for $M_o = 12$

Figure 4.10 Loss and σ_{RMS} versus epochs for $M_o = 12$ and 24 with three fixed scatterers at the center.

as the loss suddenly dropped at epoch 40 and caused the TSCS increased at epoch 40 as shown in Figure 4.10(b).

From epoch 60 to 150, the loss functions slowly converged to zero as the TSCS steady decreased to a local minimum as shown in Figure 4.10(a) and (b). Compared to the loss function for M_o = 12, the loss function converged faster and more steady to zero for M_o = 24. However, the TSCS rapidly decreased from epoch 0 to 60, but slowly decreased from epoch 60 to 120 as shown in Figure 4.10(d). It is because there are many extremum points and saddle points when the number of scatterer is large.

According to Figure 4.10(c), the loss converged to zero at epoch 20 and remained close to zero toward to end of the epochs, which indicates that the generator reached the regions where the generated devices have very small gradients (i.e. the local optima). Because the loss is so small, the back propagation only slightly update the weights of the generator, which makes the generator hard to get out of the local optima. Therefore, the generator stuck in a local optima from epoch 40 to 120 and finally got out of the unwanted regions at epoch 120. However, it quickly fell into another local optima at epoch 125 as shown in Figure 4.10(d). The analytical gradients can guide the loss function shifts toward the optimal design space and reached the minimum points at the regions after exploring and examining many possible solutions, but it is difficult for the gradients to find the optimal solution when there are too many local optima. The loss function for $M_o = 24$ swiftly converged and get trapped in local optima too and caused the generator only examined the solutions in the undesirable design region. To have a good performance, our generator need to be examine the design space well.

5 DEEP REINFORCEMENT LEARNING MODEL

In this section, we present our deep RL model which is capable of predicting the configuration with lower values of TSCS by adjusting positions of scatterers. The purpose of RL is to create a self-learning agent which interacts with an environment and finds an optimal policy through its own reward function and experience. An agent is an entity, which made of neural networks, selects actions and interacts with the environment, which represents a task or simulation [36]. As the agent outputs actions to the environment, it receives feedback from the environment and save them in a relay buffer. During the training process, the agent randomly selects a batch of data from the relay buffer and uses them to maximize its reward function. These process will continue in a cycle until an optimal policy is discovered.

There are two branches of RL algorithms: Policy Gradients [47] and Q-learning [48]. These two methods represent the actions space differently in the environment. They are both enforced by the Markov Decision Process (MDP) described in [49, pp.47-71]. The MDP provides a mathematical framework for policy iteration and improvement.

The environment has an initial state S_t. The agent selects an action A_t to interact with the environment and receives a reward R_t and a next state S_{t+1} as shown in the Figure 5.1. This process is repeated until the environment satisfy the terminal state. resulting in an episode trajectory [49]:

$$S_0,A_0,R_1,S_1,A_1,R_2,S_2,A_2,R_3,.... \tag{5.1}$$

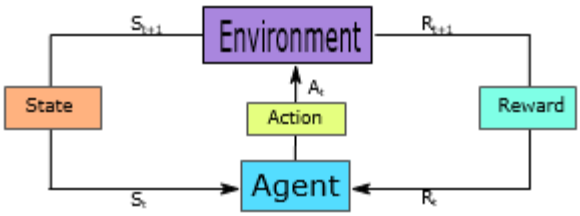


Figure 5.1 Agent interacting with environment in a Markov Decision Process (MDP).

In this setting, the goal of the agent is to maximize the sum of rewards obtained from an episode and can be defined as G_t [36]:

$$G_t \doteq R_{t+1} + R_{t+2} + \ldots + R_T \tag{5.2}$$

where T is the final time step and t is the current time step. In order to encourage the agent to find the optimized solution, we introduce a discount factor parameter γ, which values the future rewards [49]:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=t}^{T} \gamma^k R_{t+k+1}, \ \textit{where } 0 \leq \gamma \leq 1. \tag{5.3}$$

The discounted sum of rewards G_t represents the value of the action taken in the state. We can use this reward function G_t to train the agent to estimate the value of each action. The Bellman optimality equation for Q_* estimates the reward function of the optimal policy when actions are repeatedly applied as an update [49]:

$$Q_*(s, a) = E[R_{t+1} + \gamma \max_a Q_*(S_{t+1}, a) | S_t = s, A_t = a] \tag{5.4}$$

5.1 Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient (DDPG) is a model-free, actor-critic algorithm based on the deterministic policy gradient that allows the agent to operate over continuous action spaces [47]. We chose DDPG because a continuous action space better represents the possible solutions in our acoustic problem compared to a discrete action space. DDPG has four networks: actor, critic, target actor, and target critic. For our problem, the actor network takes the state from the environment and output $2M$ actions for M scatterers. One for the horizontal direction and the other one for the vertical direction. To prevent the scatterers from moving its position rapidly, we bound the action output to a reasonable range. The critic network is used to predict the Q-value for each state and action pairs. The target actor network takes the next state and output the next action. The target critic network is used to predict the next Q-values for each next state and next action pairs. The critic network will be trained to accurately predict the Q-values and the actor network will be trained to maximize the critic output. To train the target networks, we copy a portion of the parameters from the base networks using a soft update $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$ as shown in Figure 5.2. The observation is $(S_t, A_t, R_t, S_{t+1}, is\ \ Done)$. We define our reward function as $R_t = -\sigma_{RMS}$, where σ_{RMS} is defined in Equation 3.5.

5.2 Environment

In our recent RL model [36], our agent was able to discover the configuration which suppress σ and σ_{RMS} to a local minimum over a range of wave numbers for $M \leq 4$. However, our agent struggled when we increased the number of scatterers to $M \geq 8$. The issue with our original environment is that the configuration reverts back to its previous state when the geometric constrains are not satisfied. This causes the environment rejecting too many

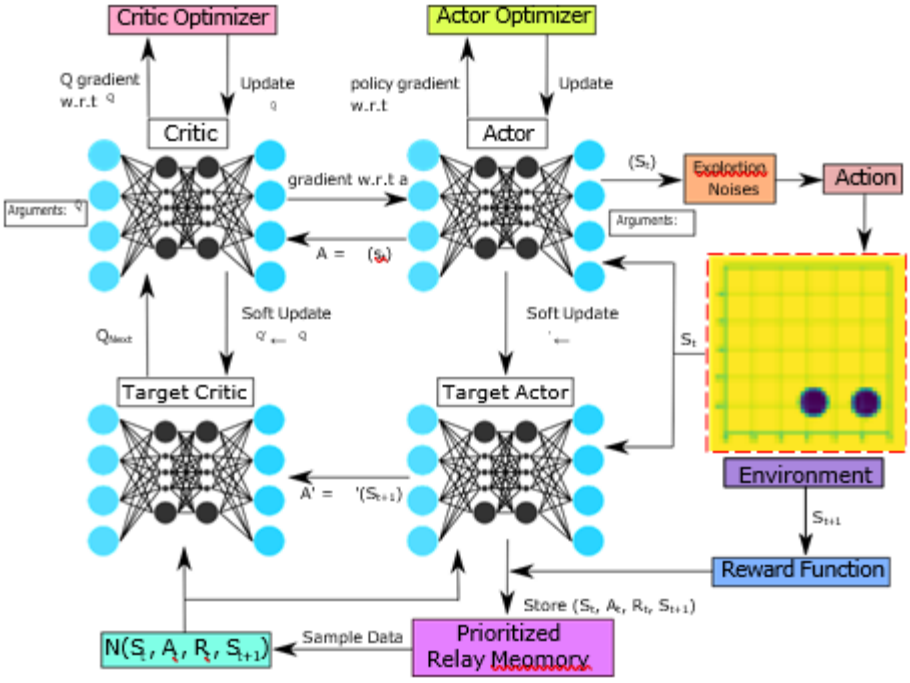


Figure 5.2 Diagram of our *DDPG* agent interacting with the environment through adjusting scatterer position.

actions when M is large. In this work, we propose to use a physics-based environment [50] which does not reject actions to overcome this issue.

Our environment allow scatterers to move in both x and y direction as long as they satisfy the constrains. To make our environment behave like a real world, we will use Newton's Second Law to define the equation of motions for all scatterers. Let's define $r_i(t)$, $v_i(t)$, and $a_i(t)$ as the position, velocity, and acceleration vector of the i-th scatterer at the current time step t, respectively. We use Equation 4.2 and Equation 4.3 to represent the collision state. Note that 1 indicates collision and 0 indicates no collision. We also want to define $r_{i,bound}$ as the distance vector between i-th device and the boundary, and $r_{i,j}$ as the distance vector between i-th and j-th device. Thus, we can define the current force $F(r_i(t)$ acting on the scatterers as below:

$$F(r_i(t)) = B(r_i(t)) \cdot (r_{i,bound}(t) \cdot k_w + c_w \cdot v_i(t)) + C(r_i(t), r_j(t)) \cdot r_{i,j} \cdot k_i \cdot v_i(t). \quad (5.5)$$

where c_w and k_w are the contact stiffness and damping of the boundary, respectively and k_i is the contact stiffness of i-th scatterer. Finally, we can sum all the forces to calculate the acceleration for each scatterers by using the equation of motions:

$$a_i(t) = \frac{A_i(t) - c_i \cdot v_i(t) + F(r_i(t))}{m_i}, \quad (5.6)$$

where m_i and c_i are the mass and damping coefficient of the scatterers, respectively. $A_i(t)$ is the actions or the forces selected by the agents to move the scatterers. By taking derivative of $a_i(t)$, we can obtain the velocity $v_i(t+1)$ and position $x_i(t+1)$ for the next time step. This cycle continues until we reached the termination state.

5.3 Actor and Critic Networks

In our model, the actor network consists of 2 fully connected layers with 128 neurons on each layer. The critic network has 8 fully connected layer with 128 neurons on each layers. Both networks apply *ReLu* as the activation function, but use *tanh* for the output layer. We also apply *BatchNorm* layer to all the layers to stabilize and normalize the input [51], except for the output layer. The learning rate for actor and critic networks are 0.0001 and 0.001, respectively. The noise scale starts with 1.2 and decays to 0.02 with 8000 steps [36]. The target networks will have the same architecture as the base networks. The parameters for the RL models are shown in Appendix A.2.

5.4 Model Evaluation for Reinforcement Learning Model

To evaluate the performance of the RL models, we compared our obtained results with ones produced by *fmincon*'s SQP algorithm. Because both RL and *fmincon* are very sensitive to the initial points, we started the optimization process with the same initial configurations to ensure accurate performance evaluation.

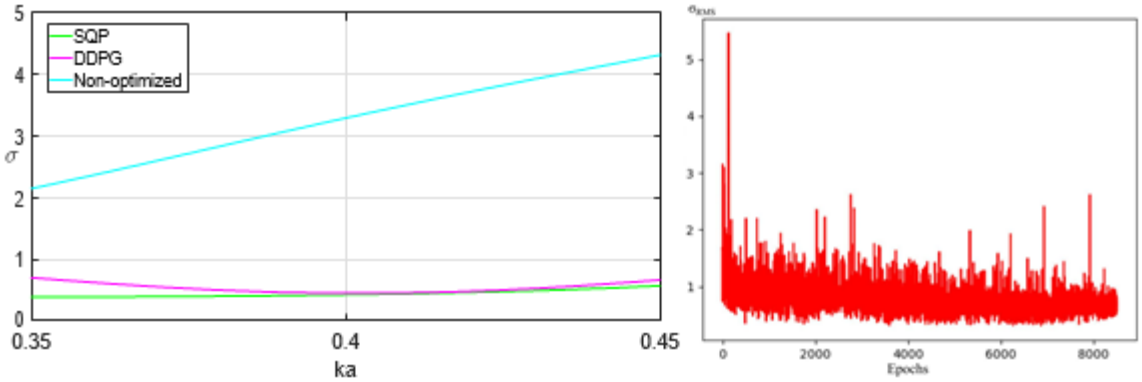
5.5 Results on Reinforcement Learning Model

Table 5.1 shows the comparison of TSCS for the initial configurations, SQP, and the RL model. Table 5.1 displays that the RL model decreased σ_{RMS} to 0.5529, 0.5438, 0.8382, and 1.4444 with the given initial configurations with a σ_{RMS} of 3.3380, 2.3788, 3.9437, and 10.8117 for M = 6, 8, 10, and 12, respectively. Our RL model has no prior experience or knowledge about TSCS, but it is able to discover its own strategy to suppress σ_{RMS} for various numbers of scatterers. Table 5.1 also illustrates that SQP has better suppression than DDPG, especially for M = 10 and 12. However, our RL model works well for lower numbers of scatterers [36], we need further improvement and better parameter tuning for larger numbers of scatterers.

Table 5.1 Comparison of TSCS for non-optimized device, the RL model and SQP at $0.35 \leq ka \leq 0.45$

Algorithm	M = 6	M = 8	M = 10	M = 12
Initial	3.3380	2.3788	3.9437	10.8117
RL Model	0.5529	0.5438	0.8382	1.4444
SQP	0.4468	0.3833	0.5845	0.5717

Figure 5.3(a) shows the performance of the non-optimized device, SQP's final device, and The RL model's optimized device at $0.35 \leq ka \leq 0.45$. The RL model achieves similar performance as SQP at $0.40 \leq ka \leq 0.45$. However, SQP's final configuration behaves a lower



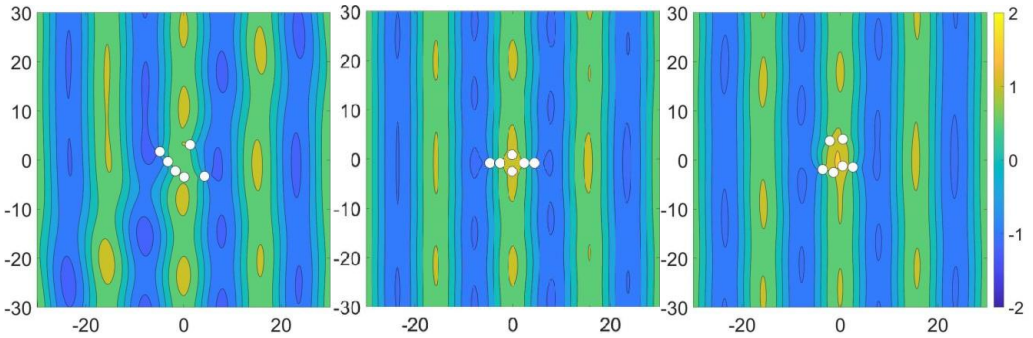
(a) The Performances of non-optimized, SQP's, and DDPG's device at $0.35 \leq ka \leq 0.45$ and $M = 6$ over 11 frequency

(b) Training Curve for DDPG over 8500 epochs

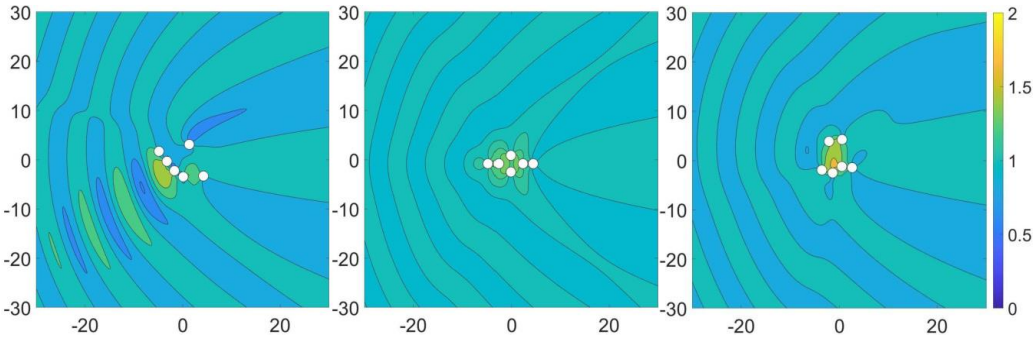
Figure 5.3 The variation of σ_{RMS} versus non-dimensional wavenumber ka for optimized configurations generated by the RL model and SQP for $M = 6$ at $0.35 \leq ka \leq 0.45$ and the σ_{RMS} versus the numbers of epochs.

σ_{RMS} at $0.35 \leq ka \leq 0.38$ than the RL's device. Figure 5.3(b) shows the σ_{RMS} discovered by the RL models versus epochs. Figure 5.3(b) presents that the RL model had a noisy convergence after the iteration of 8500 epochs. It indicates that our RL model still need to improve to achieve better and stable performance.

Figure 5.4 depicts the total acoustic pressure field for $M = 6$ at the normalized wavenumber $ka = 0.40$. Figure 5.4 (a) and (b) shows the real and absolute part of the total pressure, respectively. The RL model's device shows an overall reduction in the real part of the total pressure compared to the initial configuration, but there is a high pressure region presented between the scatterers as shown in Figure 5.4 (a). The RL model's device also greatly suppress the scattering response on the front compared to the initial configurations as shown in Figure 5.4 (b) In contrast, SQP has a better suppression on the front and rear of the scatterers. Moreover, SQP does not have any high pressure regions in between the scatterers.



(a) The real part of total acoustic pressure field (*Rep*)



(b) The absolute total pressure ($|p|$)

Figure 5.4 The pressure field produced by the initial non-optimized configurations (Left), the configuration discoverd by SQP (Middle), and the configuration obtained by the RL model (Right) at $ka = 0.40$.

Figure 5.5 displays the total pressure field of final configurations generated by the RL model and SQP as well as the initial configurations. Figure 5.5(a) is the initial configuration that SQP and the RL model optimized. Figure 5.5(b) illustrates that SQP grouped the scatterers to form a rhombus pattern. In contrast, RL mode divided six scatterers into two rows, two on the top row, and four on the bottom with an offset in the vertical direction as shown in Figure 5.5(c). Our 2D environment, where the scatterer are continuously moving and colliding with each other to avoid overlapping, can perfectly enforce the geometric constraints, but the collision between scatterers makes the RL model hard to closely align scatterers horizontally at multiple rows like 2D-GLOnets. Furthermore, this dynamic

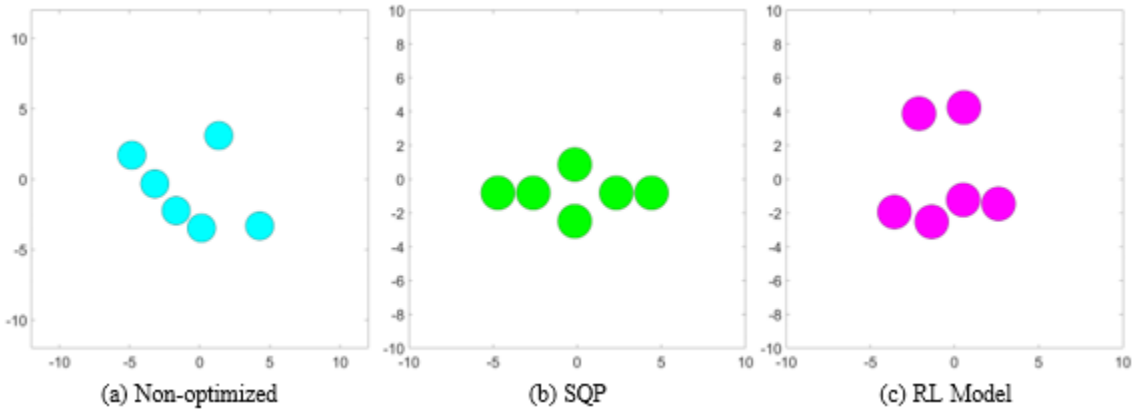


Figure 5.5 Final optimal configurations generated by SQP (green color) and the RL Model (pink color) at $0.35 \leq ka \leq 0.45$ and $M = 6$. The non-optimized configuration is shown in cyan color.

environment requires too many computation resources as it needs to constantly estimates the dynamics of each scatterers. A modified environment can help our RL model to discover better strategy to minimize the TSCS. Appendix C shows the optimal configurations discovered by RL models and SQP for $M = 8$, 10, and 12.

6 DISCUSSION

In this thesis, we implemented two models to discover the optimal configurations with the reduced or minimized TSCS. The first one is the 2D-GLOnets and the second one is the RL model.

6.1 2D-GLOnets Discussion

Our developed 2D-GLOnets can generate the optimized configurations that can efficiently suppress the effect of plane wave for $0.35 \leq ka \leq 0.45$ and $1.00 \leq ka \leq 1.20$ by rearranging the positions of each scatterer. 2D-GLOnets found the optimal solutions for $2 \leq M \leq 6$ at $0.35 \leq 0.45$ and for $2 \leq M \leq 6$ at $1.00 \leq ka \leq 1.20$, showing its ability to handle acoustic optimization for small numbers of scatterers at various range of wavenumbers. 2D-GLOnets also proves its capability to outperform *fmincon* as it yielded better results for M = 2, 6, and 12 with three fixed scatterers in the inner annulus.

When the number of scatterers increases, 2D-GLOnets started to struggle to optimize the objective function because 2D-GLOnets is less stable than *fmincon*. Unlike *fmincon*, 2D-GLOnets required a long iteration of parameters tuning. The performance of 2D-GLOnets fluctuates with the changes of parameters, especially for larger numbers of scatterers. In contrast, *fmincon* has robust and stable performance in optimizing nonlinear objective function with small dependence on the parameters. However, 2D-GLOnets is capable of outperforming *fmincon* if it is properly tuned. Additionally, we only use a batch size of 10 because we do not have enough computation resources. If we can increase the batch size to 100 or even larger number, we can better explore the design space and avoid the unfavorable local solutions. Moreover, our loss function quickly converged to zero when the gradients

were small or got trapped in local optima. We can modify the loss function to reduce the effect of gradients. For example, we can rescale the gradients such that the loss function will not be updated too rapidly, allowig the generator to further examine the design space before convegred to the minimum point. Furthermore, our results indicate that reparameterization guarantees the enforcement of geometric constraints and allows the generator to push the devices toward the favorable design regions.

The optimized devices do not guarantee to have a good performance if they are outside the target range of wavenumbers. An optimized device can sufficiently suppress the σ_{RMS} at the given range of wavenumbers, but it might increase the σ_{RMS} at the other range of wavenumbers compared to the initial configurations.

6.2 RL Model Discussion

The RL model shows its ability to decrease σ_{RMS} for a given initial configuration. The RL model outperformed SQP at lower numbers of scatterers [36], but it did not perform better than SQP consider a larger number of scatterers. Moreover, the RL model takes more than 10 hours to complete 8500 epochs for $M = 6$ on the Dell workstation and the result still did not converge well. Note that the computation time varies depend on the initialization. On the other hand, SQP takes less than 1 minutes to optimize 6 scatterers on the Dell workstation. 2D-GLOnets requires about 30 minutes to run 150 epochs on the Dell workstation and converged to a better result compared to the RL model. 2D-GLOnets and SQP have better performance than the RL model in term of lower σ_{RMS} and the computation time. The RL model's computation time and convergence took too longer, which makes the RL model difficult to tune the parameters when the number of scatterer is large. Our environment is the

cause of the long computation time as it repeatably computes the dynamics of the scatterers for a long iteration. Moreover, there are collisions between scatterers to avoid overlapping, but these behaviors are undesirable, which makes the scatterers difficult to align closely to each other as shown in Appendix B.1. One solution for this problem is to modify and transfer my reparameterization into a 2D environment. The reparameterization is an efficient and robust method for geometric constraints and it is a more suitable environment for our agent to interact with. Additionally, our reward function only includes the σ_{RMS}. We can include the analytical gradients as described in Equation 3.10 to guide our agent toward the optimal region in the reward function.

7 CONCLUSIONS

An ideal acoustic device should have zero σ_{RMS} for a broad range of wavenumbers. Currently optimization methods are still inadequate to solve this complicated inverse design problems. In this thesis, we proposed two methods that have the potential to discover the optimal acoustic cloaking devices. The RL model is a doable approach, but its computation makes this approach impractical to implement. The RL model's ability is undermined by the computation resources on the current computers (Dell remote workstation and Docker system). Improvement and modification are needed for the RL model to perform better in solving the multiple scattering problems. A better computation machine, environment, and reward function can enhance the performance of the RL model.

On the other hand, 2D-GLOnets with reparametrization is capable of finding devices that can greatly suppress σ_{RMS} at various ranges of wavenumbers. This method transforms the illegal devices into the constrained design space in a manner that naturally enforces geometric requirements without adding a penalty term in the loss function. Unlike the original GLOnet model [12] which only can deal with 1D photonic problems, our model can handle 2D acoustic response taking into account the geometric constraints. Moreover, 2D-GLOnets has the capability to outperform *fmincon*, which makes it a promising approach for inverse design problems; however, 2D-GLOnets is still underdeveloped as it has poor performance for larger numbers of scatterers and less vulnerable to the parameters. A well-tunned 2D-GLOnets will be a promising optimizer for acoustic cloaking problems and other matematerial design problems. Moreover, 2D-GLOnets will have a robust performance with a better exploration in the design space for a larger numbers of scatterers. A modified loss

function and a large batch size could enhance 2D-GLonets' ability to examine the design space.

Although 2D-GLOnets and the RL model do not perform well at a larger number of scatterers, further development and modification can make our model perform better for more complex and challenging problems. Moreover, our models provide valuable insights that inspire other researchers in this field to use DL to develop new optimization techniques that solve different types of engineering design problems.

7.1 Future Work

Future work for 2D-GLOnets should focus on larger numbers of scatterers. We need to make proper adjustment and modification on 2D-GLOnets to avoid the local optima when dealing with a large number of scatterers. We can also add more design parameters, such as materials, radius, and shape of the scatterers. Furthermore, we can apply this model to other types of material design problems, such as acoustic lens. One suggestion to improve our RL model is to use a high performance computation machine, so we can better tune the hyper parameters. Another suggestion is to combine the RL model and 2D-GLOnets [39], [40]. 2D-GLOnets will be used to locate the potential optimal design regions and the RL model with a well-defined reward function and environment will be utilized to search for a better solution. The proposed generative modeling can be extended and employed in different design of metamaterials when physical gradients can be supplied to the model.

LITERATURE CITED

[1] J. Wagner and X. Liu, "An active vibration isolation system for vehicle seats," *SAE Transactions*, vol. 109, pp. 1076–1087, 2000.

[2] S. M. Dimitrijevic, V. M. Garcia-Chocano, F. Cervera, E. Roth, and J. Sanchez-Dehesa, "Sound insulation and reflection properties of sonic crystal barrier based on micro-perforated cylinders," *Materials*, vol. 12, pp. 1–25, 2019.

[3] S.-H. Kim, "Sound focusing by acoustic luneburg lens," Cornell University. https://doi.org/10.48550/arXiv.1409.5489, 2014.

[4] F. A. Amirkulova and A. N. Norris, "The gradient of total multiple scattering cross-section and its application to acoustic cloaking," *Journal of Theoretical and Computational Acoustics*, vol. 28, p. 1950016, Mar. 2020.

[5] A. N. Norris, "Acoustic cloaking theory," *Proceedings of the Royal Society A*, vol. 464, pp. 2411–2434, 2008.

[6] A. N. Norris, F. A. Amirkulova, and W. J. Parnell, "Source amplitudes for active exterior cloaking," *Inverse Problems*, vol. 28, p. 105002, 2012.

[7] V. M. García-Chocano, L. Sanchis, A. Díaz-Rubio, J. Martínez-Pastor, F. Cervera, R. Llopis-Pontiveros, and J. Sánchez-Dehesa, "Acoustic cloak for airborne sound by inverse design," *Applied Physics Letters*, vol. 99, p. 074102, Aug 2011.

[8] L. Sanchis, V. García-Chocano, R. Llopis-Pontiveros, A. Climente, J. Martínez-Pastor, F. Cervera, and J. Sánchez-Dehesa, "Three-dimensional axisymmetric cloak based on the cancellation of acoustic scattering from a sphere," *Physical Review Letters*, vol. 110, pp. 124301, Mar 2013.

[9] Z. Lu, L. Sanchis, J. Wen, L. Cai, Y. Bi, and J. Sánchez-Dehesa, "Acoustic cloak based on Bézier scatterers," *Scientific Reports*, vol. 8, p. 12924, Aug 2018.

[10] S. Ruder, "An overview of gradient descent optimization algorithms," https://doi.org/10.48550/arXiv.1609.04747, 2016.

[11] O. Sigmund and K. Maute, "Topology optimization approaches," *Structural and Multidisciplinary Optimization*, vol. 48, pp. 1031–1055, 2013.

[12] J. Jiang and J. A. Fan, "Global optimization of dielectric metasurfaces using a physics-driven neural network," *Nano Letters*, vol. 19, pp. 5366–5372, Jul 2019.

[13] P. Andersen, V. Henriquez, L. Sanchis, and J. Sánchez-Dehesa, "Design of multi-directional acoustic cloaks using two-dimensional shape optimization and the boundary

element method," in *Proceedings of the23rd International Congress of Acoustics*, pp. 5600–5606, Department of Electrical Engineering, Acoustic Technology, Technical University of Denmark, 2019.

[14] J. Andkjær and O. Sigmund, "Topology optimized low-contrast all-dielectric optical cloak," *Applied Physics Letters*, vol. 98, p. 021112, Jan 2011.

[15] M. P. Bendsoe and O. Sigmund, *Topology Optimization: Theory, Methods, and Applications*, Springer Science & Business Media, 2013.

[16] S. Sivanandam and S. Deepa, "Genetic algorithms," in *Introduction to Genetic Algorithms*, pp. 15–37, Springer, 2008.

[17] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*, pp. 7–15, Springer, 1987.

[18] J. Rho and J. A. Fan, "Freeform metasurface design based on topology optimization," *MRS Bulletin*, vol. 45, pp. 196–201, 2020.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS Proceedings of Advances in Neural Information Processing Systems 27*, pp. 1-9, 2014.

[20] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," Cornell University, https://doi.org/10.48550/arXiv.1812.04948, 2019.

[21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros,"Context encoders: Feature learning by inpainting," Cornell University, https://doi.org/10.48550/arXiv. 1604.07379, 2016.

[22] C. Gurbuz, F. Kronowetter, C. Dietz, M. Eser, J. Schmid, and S. Marburg, "Generative adversarial networks for the design of acoustic metamaterials," *The Journal of the Acoustical Society of America*, vol. 149, pp. 1162–1174, Feb 2021.

[23] J. A. Hodge, K. V. Mishra, and A. I. Zaghloul, "Joint multi-layer gan-based design of tensorial rf metasurfaces," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, 2019.

[24] A.-P. Blanchard-Dionne and O. J. F. Martin, "Successive training of a generative adversarial network for the design of an optical cloak," *OSA Continuum*, vol. 4, p. 87, Dec 2020.

[25] X. Han, Z. Fan, Z. Liu, C. Li, and L. J. Guo, “Inverse design of metasurface optical filters using deep neural network with high degrees of freedom,” *InfoMat*, vol. 3, pp. 432-442, Jun 2020.

[26] S. An, B. Zheng, H. Tang, M. Y. Shalaginov, L. Zhou, H. Li, M. Kang, K. A. Richardson, T. Gu, J. Hu, C. Fowler, and H. Zhang, “Multifunctional metasurface design with a generative adversarial network,” *Advanced Optical Materials*, vol. 9, p. 2001433, Jan 2021.

[27] A. Challapalli, D. Patel, and G. Li, “Inverse machine learning framework for optimizing lightweight metamaterials,” *Materials & Design*, vol. 208, p. 109937, 2021.

[28] Z. Hou, P. Zhang, M. Ge, J. Li, T. Tang, J. Shen, and C. Li, “Metamaterial reverse multiple prediction method based on deep learning,” *Nanomaterials*, vol. 11, p. 2672, 2021.

[29] A. H. Nobari, W. Chen, and F. Ahmed, “PcDGAN: A continuous conditional diverse generative adversarial network for inverse design,” Cornell University, https://doi.org/10.48550/arXiv.2106.03620, 2021.

[30] L. Wang, Y.-C. Chan, F. Ahmed, Z. Liu, P. Zhu, and W. Chen, “Deep generative modeling for mechanistic-based learning and design of metamaterial systems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113377, Dec 2020.

[31] X. Han, L. Zhang, K. Zhou, and X. Wang, “Deep learning framework dnn with conditional wgan for protein solubility prediction,” Cornell University, https://doi.org/10.48550/arXiv.1811.07140, 2018.

[32] M. Khodaee, M. Banakermani, and H. Baghban, “Gan-based metamaterial terahertz bandpass filter design: tunability and ultra-broad passband attainment,” *Applied Optics*, vol. 54, pp. 8617–8624, 2015.

[33] T. Tran, F. Amirkulova, and E. Khatami, “Acoustic cloak design via machine learning,” Cornell University, https://doi.org/10.48550/arXiv.2111.01230, 2021.

[34] I. Sajedian, T. Badloe, and J. Rho, “Optimisation of colour generation from dielectric nanostructures using reinforcement learning,” *Optics Express*, vol. 27, p. 5874, Feb 2019.

[35] I. Sajedian, H. Lee, and J. Rho, “Double-deep q-learning to increase the efficiency of metasurface holograms,” *Scientific Reports*, vol. 9, Jul 2019.

[36] T. Shah, L. Zhuo, P. Lai, A. De La Rosa-Moreno, F. Amirkulova, and P. Gerstoft, "Reinforcement learning applied to metamaterial design," *The Journal of the Acoustical Society of America*, vol. 150, pp. 321–338, 2021.

[37] S. Kim, I. Kim, and D. You, "Multi-condition multi-objective optimization using deep reinforcement learning," Cornell University, https://doi.org/10.48550/arXiv.2110.05945, 2021.

[38] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem, "Direct shape optimization through deep reinforcement learning," *Journal of Computational Physics*, vol. 428, p. 110080, 2021.

[39] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (organ) for sequence generation models," https://doi.org/10.48550/arXiv.1705.10843, 2017.

[40] E. Putin, A. Asadulaev, Y. Ivanenkov, V. Aladinskiy, B. Sanchez-Lengeling, A. Aspuru-Guzik, and A. Zhavoronkov, "Reinforced adversarial neural computer for de novo molecular design," *Journal of Chemical Information and Modeling*, vol. 58, pp. 1194–1204, May 2018.

[41] F. A. Amirkulova and A. N. Norris, "Acoustic multiple scattering using fast iterative techniques," in *Proceedings of the International Mechanical Engineering Congress and Exposition*, vol. 13, pp. V013T01A005, 2017.

[42] Z. Yang, J.-L. Wu, and H. Xiao, "Enforcing deterministic constraints on generative adversarial networks for emulating physical systems," Cornell University, https://doi.org/10.48550/arXiv.1911.06671, 2019.

[43] M. Chen, J. Jiang, and J. Fan, "Design space reparameterization enforces hard geometric constraints in inverse-designed nanophotonic devices," Cornell University, https://doi.org/10.48550/arXiv.2007.12991, 2020.

[44] J. Jiang and J. A. Fan, "Simulator-based training of generative neural networks for the inverse design of metasurfaces," *Nanophotonics*, vol. 9, pp. 1059–1069, Nov 2019.

[45] R. Shankar, *Principles of quantum mechanics*. Springer Science & Business Media, 2012.

[46] The MathWorks Inc, "When the solver fails. mathworks matlab documentation," MathWorks, https://www.mathworks.com/help/optim/ug/when-the-solver-fails.html.

[47] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," Cornell University, https://doi.org/10.48550/arXiv.1509.02971, 2015.

[48] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” https://doi.org/10.48550/arXiv.1312.5602, 2013.

[49] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, MIT Press Ltd, 2018.

[50] The MathWorks Inc, “Train multiple agents to perform collaborative task, ” MathWorks, https://www.mathworks.com/help/reinforcement-learning/ug/train-2-agents-to-collaborate.html, 2021.

[51] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” Cornell University, https://doi.org/10.48550/arXiv.1502.03167, 2015.

APPENDIX A

HYPER PARAMETERS

A.1 Hyper Parameters of 2D-GLOnets

Table A.1 Hyper parameters of 2D-GLOnets

Name	Value	Description
Noise Dimensions	depends on M	Determine the size of the noise vector
Noise scale	1.0	Determine the amplitude of noise
a	1.0	Radius of scatterer
numIter	2000	Control the number of epochs
lr	0.001	Learning rate
Optimizer	Adam	Optimizer used for both networks
scale	50	Multiplier at the output layers
Rout	15	Radius of outer boundary in meters
Rin	2.31	Radius of inner annulus in meters
β	10	Tunable parameter in loss function
Hidden size	128	Dimension of hidden layers

A.2 Hyper Parameters of RL Model

Table A.2 Hyper parameters of the RL Model

Name	Value	Description
γ	0.9	future reward
Initial noise scale	0.5	Scale of normally distributed noise
Final noise scale	0.01	Final scale of noise at end of training
τ	0.001	target network update rate
α	0.7	How much to use prioritized sampling
β	0.5	How aggressively to apply importance-sampling weights
Optimizer	Adam	Optimizer used for both networks
Actor learning rate	1e−4	Learning rate of actor optimizer
Critic learning rate	1e−3	Learning rate of critic optimizer
Critic weight decay	1e−2	Regularization term to prevent overfitting
Batch size	64	Number of samples per batch of gradient descent
m_i	2.0	mass of scatterer
c_i	1.0	Damping coefficient of scatterer
k_i	1.0	stiffness coefficient of scatterer
c_w	2.0	Damping coefficient of boundary
k_w	1.08	stiffness coefficient of boundary

APPENDIX B

OPTIMAL CONFIGURATIONS GENERATED BY 2D-GLONETS AND *FMINCON*

B.1 Final Devices Discovered by 2D-GLOnets and *fmincon* at $0.35 \leq ka \leq 0.45$

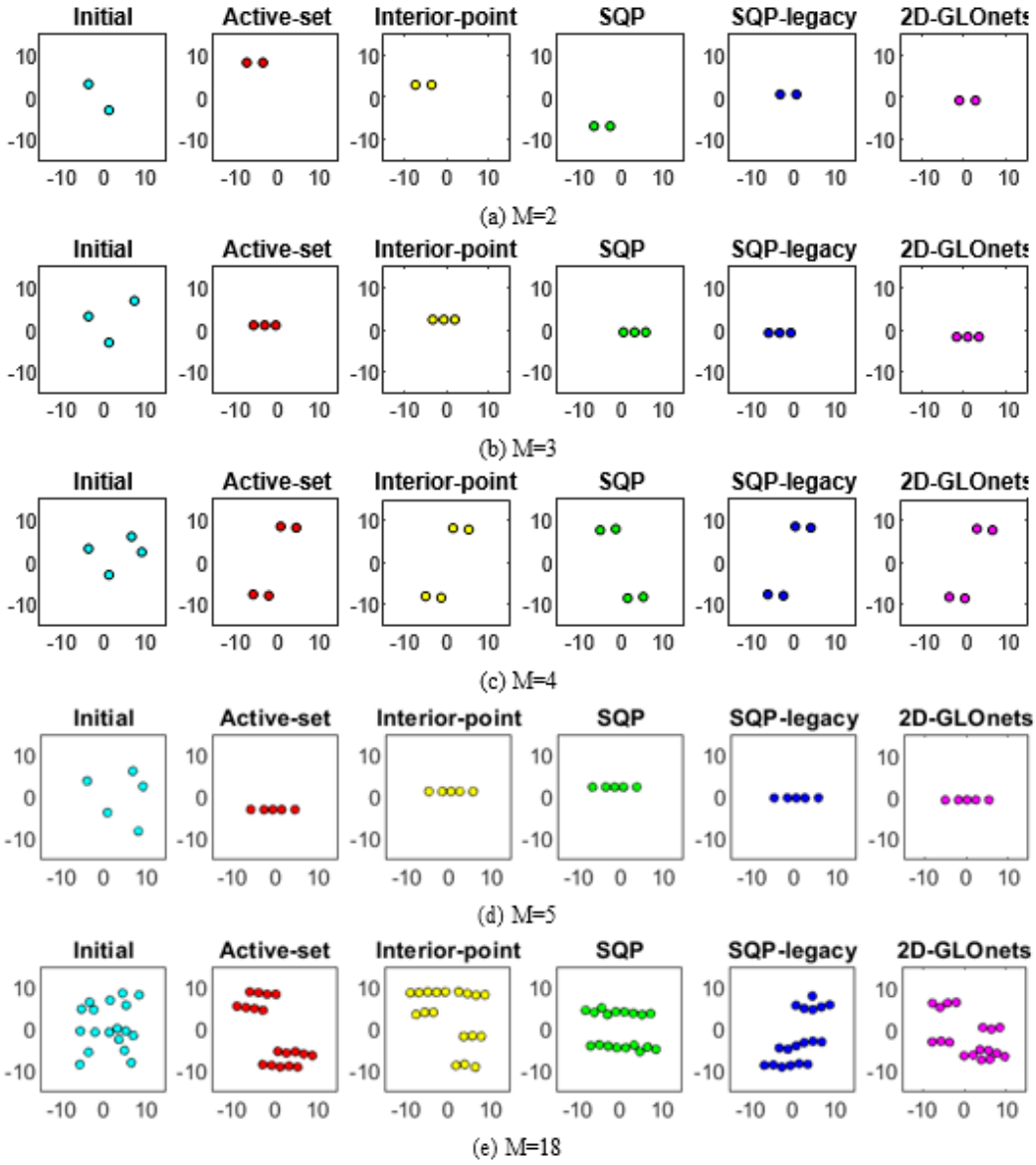


Figure B.1 Optimized configurations generated by Active-set (red corlor), Interior-point (yellow color), SQP (green color), SQP-legacy (blue color),and 2D GLOnet (pink color) at $0.35 \leq ka \leq 0.45$.

B.2 Final Devices Generated by 2D-GLOnets and *fmincon* at $1.00 \leq ka \leq 1.20$

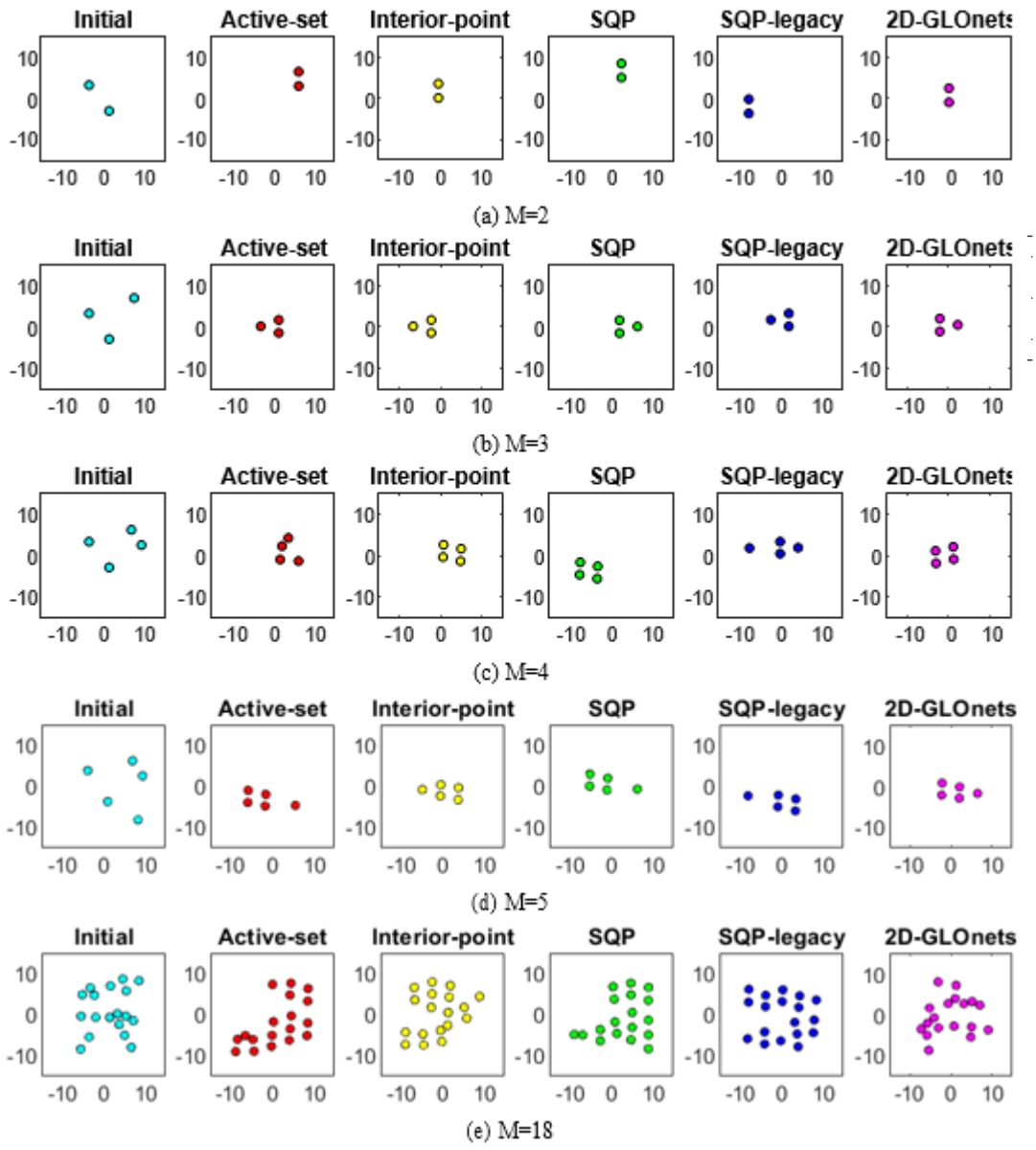


Figure B.2 Optimal configurations of rigid cylinders discovered by Active-set, Interior-point, SQP, SQP-legacy, and 2D-GLOnets at $1.00 \leq ka \leq 1.20$. The initial configurations are denoted by cyan color.

B.3 Optimal Devices Discovered by 2D-GLOnets and *fmincon* with Three Fixed Scatterers Inside An Annulus Region at $0.35 \leq ka \leq 0.45$

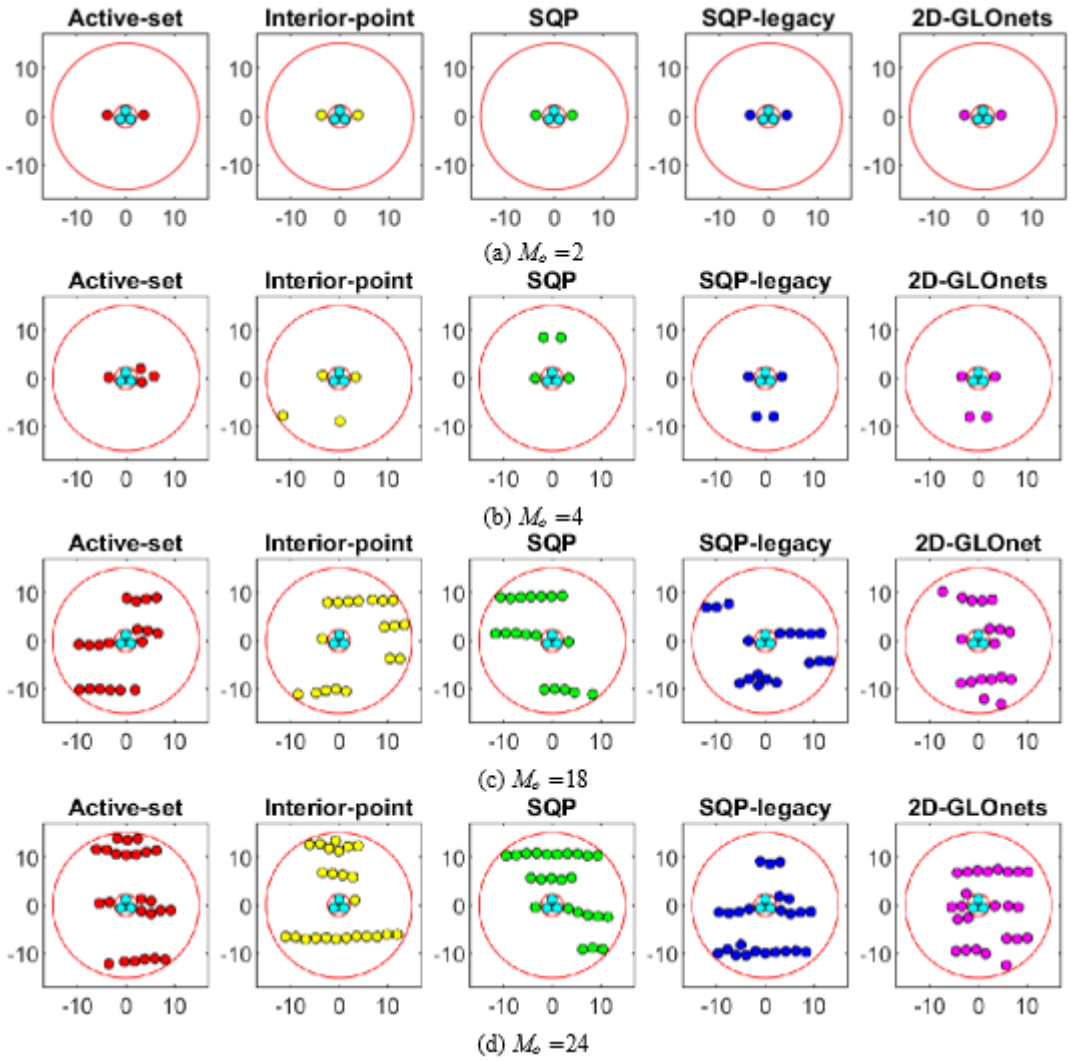


Figure B.3 Optimized configuration discovered by *fmincon* and 2D-GLOnets with three fixed rigid cylinders inside an annulus for $M_o = 4$, 18, and 24.

APPENDIX C

OPTIMAL DEVICES DISCOVERED BY THE RL MODEL AND *FMINCON* AT $0.35 \leq KA \leq 0.45$

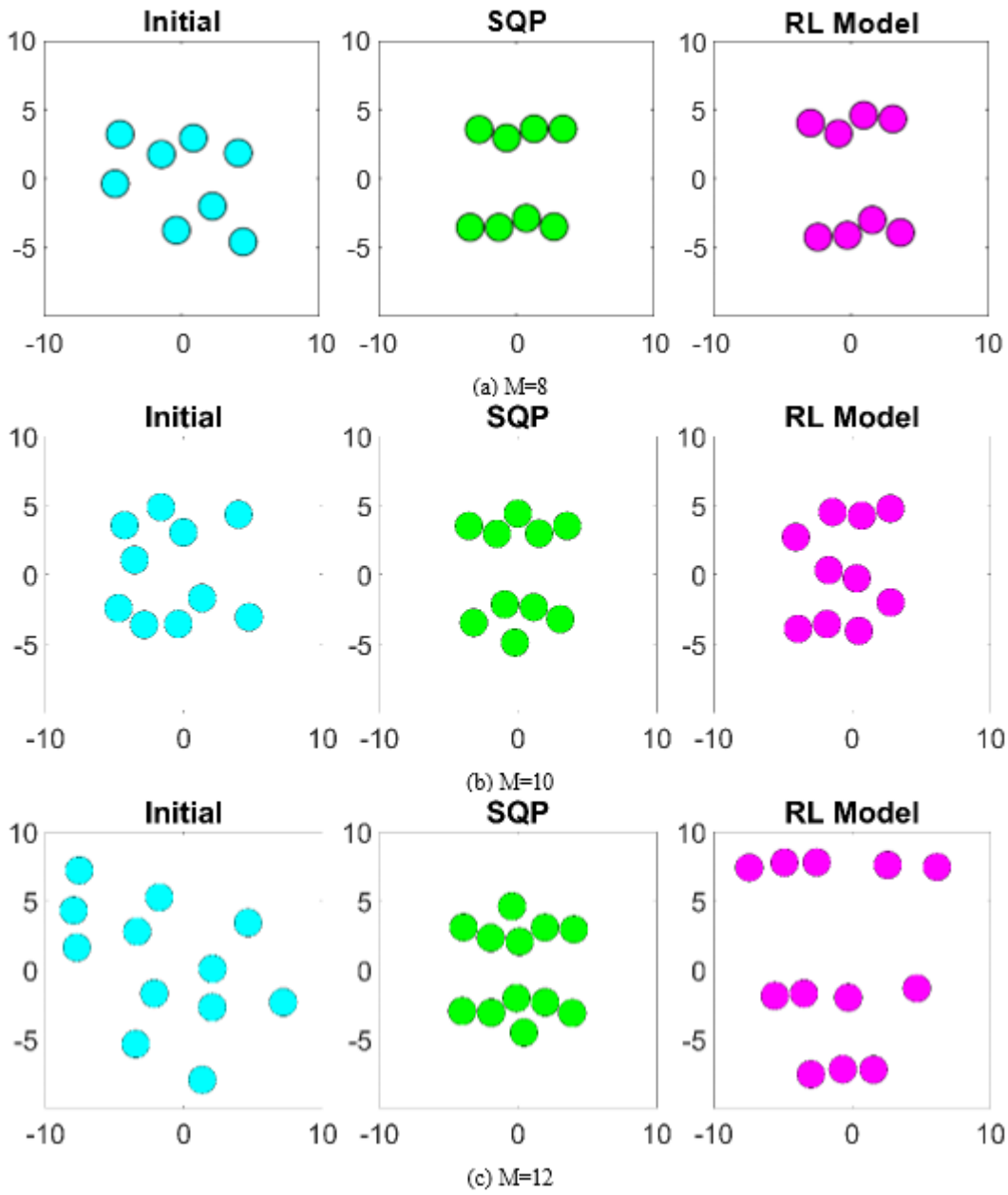


Figure C.1 Optimal configurations found by the RL models and *fmincon* for $M = 8$, 10, and 12.