International Journal of Internet, Broadcasting and Communication Vol.16 No.2 127-135 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.2.127

IJIBC 24-2-16

Deep Learning-Based Inverse Design for Engineering Systems: A Study on Supervised and Unsupervised Learning Models

Seong-Sin Kim

Assistant Professor, School of AI Convergence, Soongsil University, Korea
kss0222@ssu.ac.kr

Abstract

Recent studies have shown that inverse design using deep learning has the potential to rapidly generate the optimal design that satisfies the target performance without the need for iterative optimization processes. Unlike traditional methods, deep learning allows the network to rapidly generate a large number of solution candidates for the same objective after a single training, and enables the generation of diverse designs tailored to the objectives of inverse design. These inverse design techniques are expected to significantly enhance the efficiency and innovation of design processes in various fields such as aerospace, biology, medical, and engineering. We analyzes inverse design models that are mainly utilized in the nano and chemical fields, and proposes inverse design models based on supervised and unsupervised learning that can be applied to the engineering system. It is expected to present the possibility of effectively applying inverse design methodologies to the design optimization problem in the field of engineering according to each specific objective.

Keywords: *Deep Learning, Inverse Design, Design Optimization, Supervised and Unsupervised Learning*

1. Introduction

Inverse design is a methodology that sets a desired result or target performance in advance and finds an optimized design that satisfies these criteria [1]. Conventional methodologies have been focused on optimizing performance to generate output data that matches the targeted results derived from the given input data [2]. However, in actual engineering design optimization problems, there are often cases where it is necessary to instantly determine the optimal design that corresponds to a specific constraint in reverse. This process requires iterative experimentation, which can be time-consuming and costly. To address this issue, recent studies have increasingly utilized data-driven deep learning (DL) approaches. DL has been widely used in nanophotonics, synthesis paths of organic compounds, optimization of solar power generation, and a variety of other solid materials [3, 4]. This paper analyzes the main features and research cases of inverse design based on supervised and unsupervised learning, and proposes inverse design models applicable in the field of engineering.

Manuscript Received: March. 29, 2024 / Revised: April. 8, 2024 / Accepted: April. 15, 2024
Corresponding Author: kss0222@soongsil.ac.kr
Tel: +82-2-820-0956
Assistant Professor, School of AI Convergence, Soongsil University, Korea

2. Related works

The purpose of design optimization is to design a system with maximum performance while satisfying a given constrain. Traditional design optimization methods can be described through methods such as Genetic algorithms(GA) [5] and the gradient-based Sequential quadratic programming (SQP) [6]. GA and SQP often require iterative optimization processes in high-dimensional design spaces, leading to high computational costs. In contrast, DL can efficiently find complex structures and patterns within large datasets, reducing computational costs and time in the inverse design process. DL-based inverse design can be divided into supervised and unsupervised learning; supervised learning is defined as finding complex nonlinear relationships between pairs of pre-labeled data [7]. Unsupervised learning, dealing with unlabeled data, involves discovering significant patterns within the data without answers, posing challenges in learning. However, it excels in generating new patterns in entirely new data.

Recent studies on inverse design using supervised learning models include research that compared known label values for specific optical properties and those using deep neural networks to design complex photonic structures with large datasets of electromagnetic scattering instances required [3, 8]. A method has been proposed that instantly generates optimal designs satisfying multiple objectives, such as APT and drag torque, in automotive brake systems through multidisciplinary inverse design with DL [9]. As for research in inverse design using unsupervised learning models, examples include studies that defined structure-property relationships through Generative Adversarial Network (GAN) to generate optical spectrum in meta-surfaces [10]. A model based on GAN has been proposed to generate new hypothetical materials with desired properties [11]. Additionally, conditional GAN(CGAN) have been used in inverse design techniques to generate structural designs of meta-atom and meta-molecule [12]. Regression and conditionally generative adversarial neural network (RCGAN) have been used to implement the generation of materials with specific target bandwidth values, specifically for graphene and boron-nitrogen hybrids [13]. A variational auto-encoder (VAE) has been utilized to output meta-materials and their spectra corresponding to targeted optical responses [14]. VAE was used for layout design, and unsupervised learning was primarily used to discover new structures for various engineering problems [15]. The intrinsic relationships between microstructures and electromagnetic responses were effectively discovered using conditional VAE (CVAE) [16]. CVAE and GAN were combined to generate airframe shapes corresponding to target wall Mach number distributions that match specified features such as suction peak position, shock, and rear load [17].

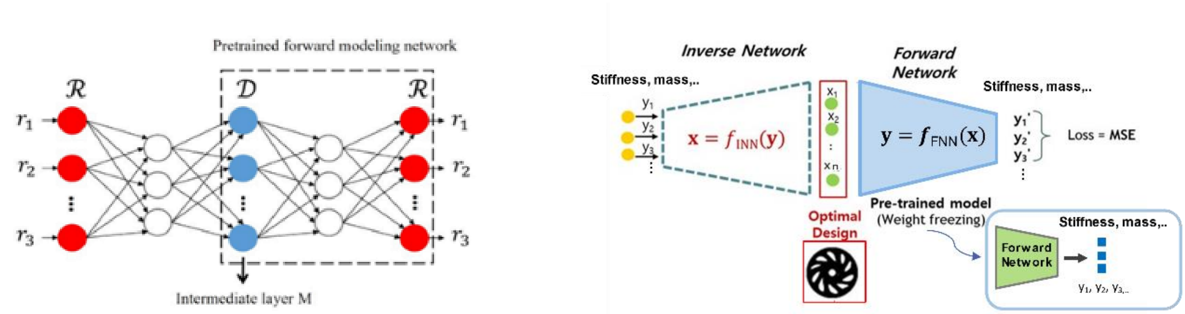
3. Methods

3.1 Inverse design via supervised learning

Supervised training learns the mapping relationship input (x) to output data (y) to find complex and nonlinear relationships between two sets of pre-labeled data. A supervised learning model f: x $\rightarrow y$ is trained to approximate a function that uses given design variables (x) to predict engineering performance (y). However, the inverse mapping f: $y \rightarrow$ x may not always be successful due to the presence of many potential design candidates (x). To resolve this, Liu et al. (2018) first trained a forward DL model (Figure 1(a)) [3]. After training the forward model, the weights of the forward neural network (FNN) are fixed, and the network undergoes retraining. The network is optimized to receive inputs that yield the closest results.

3.1.1. Proposed inverse design model via supervised learning

Figure 1(b) shows the architecture of an inverse design network through supervised learning, which applies a target value to a binary wheel according to the network proposed in [10]. This network combines a pre-trained FNN fFNN(x $\rightarrow y'$) and a subsequently trained inverse network fINN($y \rightarrow$x). After training a forward DL model with good prediction performance, the weights of fFNN are fixed, and the network is retrained. As a result, the x of fINN is passed to the frozen forward network fFNN, which outputs y', and fINN is trained to minimize the loss between y and y'. Thus, fINN can obtain the optimal design x for y. FINN that generates the design parameters that satisfy the target performance is attached in front of the fFNN and retrained, after learning the fFNN that predicts the performance of the wheel design parameters.



(a) Structure of tandem network

(b) Inverse design architecture model for wheel images via supervised learning

Figure 1. Architectures of inverse design via supervised learning

When the training is completed, fINN can generate optimized design parameters as output by inputting the target performance. The parameters of the fFNN model were optimized to minimize the mean square error (MSE) shown in Equation (1).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

In Equation (1), y and $\hat{y}$ are the ground truth and the estimation by a model, respectively. MSE is commonly used to evaluate the performance of a regression model, with lower values indicating a more accurate model prediction.

Table 1 outlines a process for generating wheel designs using supervised learning. The method starts with a training dataset and iteratively adjusts the design parameters using gradient descent to minimize the mean squared error (MSE) between the predicted and actual data, until the error is within an acceptable range. The process utilizes two functions, fFNN and fINN, where fINN generates the final design parameters based on the optimized input.

Table 1. Iterative optimization process with supervised learning

Algorithm :	Generating for wheel design via supervised learning
1. Training data set $\{\mathbf{x}_n, y_n \mid n = 1, \cdots, N\}$, $MSE(y, f_{FNN}(\mathbf{x}))$	
2. **repeat**:	
3. if e > δ then (δ : best error, e = $MSE(y, f_{INN}(\mathbf{x}))$)	
x' initialization for gradient descent	
Define error: E = $MSE(y, f_{FNN}(x'))$	

Gradient descent $x'_{(i+1)} = x'_i - \eta \nabla E(x'_i)$
4. end if
5. **until** $e <= \delta$
6. $f_{INN}(y) \rightarrow x'_{new}$

* f_{FNN} : FNN function, f_{INN} : INN function

3.2 Inverse design via unsupervised learning

Unsupervised learning is a machine learning technique that learns patterns and characteristics from unlabeled data. It is commonly used for tasks such as clustering or dimensionality reduction, and is also used for generative modeling. The most representative unsupervised deep inverse design models are GAN [18] and VAE [19], which generate new data by learning the characteristics and distribution of the data. The goal of unsupervised inverse design is to learn the relationship between design variables and engineering performance by incorporating conditions. These unsupervised inverse design models can generate many unique design candidates based on conditional inputs of target performance.

3.2.1. Generative Adversarial Network (GAN)

GAN uses an adversarial training approach, consisting of two networks, a generator and a discriminator that compete in a zero-sum game and learn simultaneously. The generator takes random noise and produces an image that needs the characteristics of the desired object, while the discriminator determines if the generated image comes from the data we want. Many variants of GAN have been developed for different purposes, most notably CGAN. The objective function of CGAN is as follows (Equation (2))[20].

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathrm{x}\sim p_{data}(\mathrm{x})}[\log D(\mathrm{x}|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{2}$$

CGAN includes additional conditional information as an input to G. It typically generates data that is trained using a modified objective function, which includes an added condition y, to achieve a target performance.

Figure 2(a) shows an example of generating an optical spectrum from a meta-surface [10]. The generator makes an image of a structure that satisfies the optical properties of the transmission spectrum from the spectrum T and random noise z and provides a binarized image. The critic classifies the geometric data and the generator with images of real structures. The simulator is a pre-trained network that approximates the transmission spectrum T for a given pattern in the input, and the provided structure images are validated by the simulator's optical property predictions.

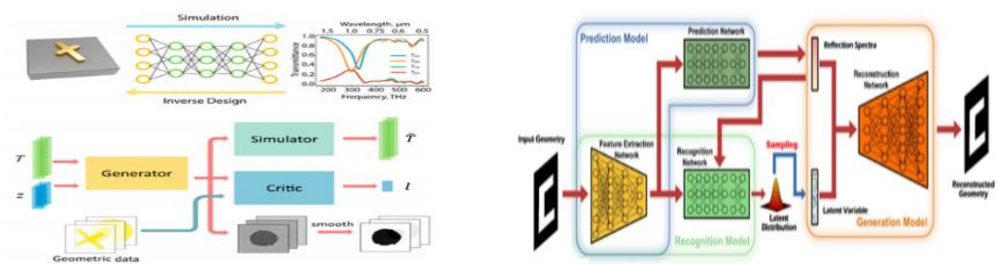
3.2.2. Variational Auto-Encoder (VAE)

Unlike a GAN, which generates an image from random noise, the structure of a VAE receives an image of a structure as input through an encoder. It creates a latent variable z through encoding that compresses the input data into low dimensions. It also reconstructs the original data by probabilistically extracting and decoding z from the latent space. Conditional Variational Autoencoder CVAE is a variant of VAE that learns constraints and uses VAE's generation function to generate design candidates that automatically satisfy all constraints. Here is the objective function of a CVAE (Equation (3))[19].

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right] \quad (3)$$

The DKL term computes the Kullback-Leibler divergence between the encoder's distribution $q\phi(z|\text{x}(i))$ of the latent variable z given the input data $\text{x}(i)$ and the prior distribution $p\theta(z)$, measuring how close the encoder distribution is to the prior. The $Eq\phi(z|\text{x}(i))$ term calculates the expected log likelihood of the reconstructed data $\text{x}(i)$, evaluating how well the model reconstructs the data. Optimizing these terms allows the VAE to effectively encode and reconstruct the input data.

Figure 2(b) shows a network structure that generate optimal metamaterials through the relationship between physical structure and optical response [14]. It consists of a recognition model that encodes the metamaterial into latent space, a prediction model that predicts the reflection spectrum forward for the metamaterial, and a generation model that generates new metamaterials that satisfy the desired optical properties by applying various latent variable values.



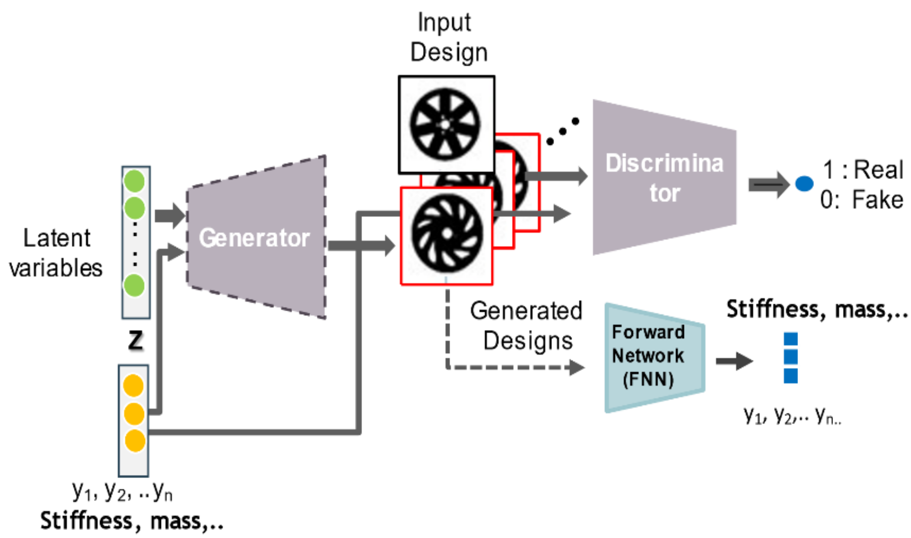
(a) Inverse design generating for an optical spectrum

(b) Architecture of the deep generative model for metamaterial designs

Figure 2. Inverse design architectures via unsupervised learning

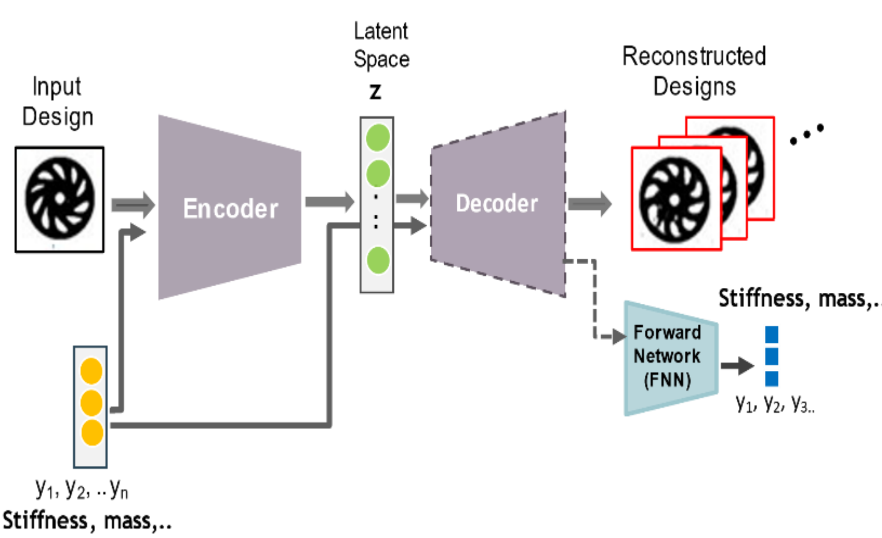
3.2.3. Proposed inverse design model via unsupervised learning

This paper proposes a model for generating wheel images using CGAN and CVAE, which are variations of GAN and VAE. The wheel image shape to be trained is represented as a binary black and white image of 128 x128 pixels. Figure 3(a) shows CGAN to generate wheel images that satisfy constraints such as desired stiffness and mass. The constructor consists of one fully connected layer and four deconvolution layers, while the discriminator consists of five convolutional layers followed by a fully connected layer. The output layer of the constructor uses the Sigmoid function as the activation function, while all other batch normalization layers use LeakyReLU as the activation function. The CGAN model is trained to minimize both the constructor loss and the discriminator loss. The generated wheel images are validated with additional FNN that predict stiffness and mass, among other things. Since the constructor was trained using known unsupervised learning, it shows the potential to design wheel images with a high degree of design freedom for a wide range of stiffnesses and mass. A CGAN model trained with topology-optimized wheel images can generate many new designs that do not exist in the training dataset.



(a) CGAN model for wheel designs

Figure 3(b) shows a schematic diagram of CVAE that add additional conditions to the wheel image generation process. CVAE receives input data along with wheel performance conditions, such as stiffness and mass, which are integrated into both the encoder and decoder. The model restores the data from the latent variable z, guided by the constraints c. This shows that there are many different wheel images that can represent constraints. In this model, the latent variables are selected probabilistically to reconstruct the image, which allows different values of the latent variables to be applied to generate new wheel geometries that satisfy the desired target conditions.



(b) CVAE model for wheel designs

Figure 3. Proposed architecture of inverse design for wheel images using unsupervised learning

Table 2 and 3 show the training process of CGAN and CVAE models. Table 2 describes a process for generating wheel designs using a CGAN. The algorithm trains a generator (G) to create fake wheel designs based on noise inputs and target performance or conditions, while a discriminator (D) learns to differentiate between the generated and real designs. Training continues until the desired accuracy is achieved through an early stopping criterion, after which the generator produces new wheel designs optimized to meet the specified conditions.

Table 2. Iterative optimization process with unsupervised learning (CGAN)

Algorithm : Generating for wheel design via CGAN	
1. Training data set $\{\mathbf{x}_n, c_n \mid n = 1, \cdots, N\}$,	c_n : Target performance or condition
2. **repeat**:	
3. G: $\mathbf{x}_{fake} = G(z, c)$, $D(\mathbf{x}_{fake}, c) \approx 1$,	G : Generator
4. D: $D(\mathbf{x}, c) \approx 1$, $D(\mathbf{x}_{fake}, c) \approx 0$,	D : Discriminator
$L_G = -\log D(G(z, c), c)$	
$L_D = -\log D(\mathbf{x}, c) - \log(1 - D(G(z, c), c))$	
5. **until** n_epoch (Early Stopping)	
6. $G(z, c) \rightarrow \mathbf{x}_{new}$	

Table 3 and the described algorithm illustrate an iterative optimization process for wheel design using a CVAE. This unsupervised learning method involves encoding the input data and conditions into a latent space, and then reconstructing it to minimize reconstruction loss and the Kullback-Leibler divergence, adjusting the balance between these losses with a hyperparameter (β). The process continues until a specified number of epochs is reached through early stopping, ultimately generating optimized wheel designs.

Table 3. Iterative optimization process with unsupervised learning (CVAE)

Algorithm : Generating for wheel design via CVAE	
1. Training data set $\{\mathbf{x}_n, c_n \mid n = 1, \cdots, N\}$,	c_n : Target performance or condition
2. **repeat**:	
3. $z = E(\mathbf{x}, c)$,	E : Encoder
4. $\mathbf{x}_{recon} = D(z, c)$,	D : Decoder
5. $L_{recon} = \lVert \mathbf{x} - D(E(\mathbf{x}_{recon}, c), c) \rVert^2$	
$L_{KL} = D_{KL}(E(\mathbf{x}, c), c) \mid\mid N(0, I)$	
$L = L_{recon} + \beta L_{KL}$,	β: Hyperparameters to balance the two losses
6. **until** n_epoch (Early Stopping)	
7. $D(z, c) \rightarrow \mathbf{x}_{new}$	

4. Conclusion

In conclusion, we propose a new approach to engineering problems by introducing the concept of inverse design. DL-based inverse design is possible with two approaches: supervised and unsupervised learning, and the appropriate method must be selected according to the nature and purpose of the design problem. Supervised inverse design approaches learn complex patterns and perform well but they are characterized by generating one optimal design at a time. On the other hand, inverse design with unsupervised learning can infer and generate new patterns from new data and can generate many different designs simultaneously. However, unsupervised learning can be difficult to train for the optimization of high-dimensional and complex problems. If these challenges are overcome, it would be a design method capable of generating many different optimal design candidates. In future work, we plan to extend the application of supervised and unsupervised inverse design techniques to various domains in real-world engineering.

In summary, the proposed inverse design model with supervised and unsupervised learning is expected to provide efficient design optimization solutions and useful guidelines for applying inverse design methodologies to real-world optimization problems in engineering system.

References

[1] S. Molesky, Z. Lin, A. Y. Piggott, W. Jin, J. Vucković, and A.W. Rodriguez, "Inverse design in nanophotonics," *Nature Photonics*, *12*(11), 659-670, 2018. DOI: https://doi.org/10.1038/s41566-018-0246-9

[2] A. Shrestha, and A. Mahmood, "Review of deep learning algorithms and architectures," IEEE Access, 7, 53040-53065, 2019. DOI: https://doi.org/10.1109/ACCESS.2019.2912200

[3] D. Liu, Y. Tan, E. Khoram, and Z. Yu, "Training deep neural networks for the inverse design of nanophotonic structures," *ACS Photonics*, *5*(4), 1365-1369, 2018. DOI: https://doi.org/10.1021/acsphotonics.7b01377

[4] A. S. Khazaal, M. Springborg, C. Fan, K. Huwig, "Optimizing small conjugated molecules for solar-cell applications using an inverse-design method," *Journal of Molecular Graphics and Modelling*, *100*, 107654, 2020. DOI: https://doi.org/10.1016/j.jmgm.2020.107654

[5] T. Zhang, J. Wang, Q. Liu, J. Zhou, J. Dai, X. Han, Y. Zhou, and K. Xu, "Efficient spectrum prediction and inverse design for plasmonic waveguide systems based on artificial neural networks," *Photonics Research*, 7(3) 368-380, 2019. DOI: https://doi.org/10.1364/PRJ.7.000368

[6] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimizatio n," *SIAM review*, *47*(1), 99-131, 2005. DOI: https://doi.org/10.1137/S003614450444609

[7] R. Caruana, and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," In *Proceedings of the 23rd international conference on Machine learning*, 161-168, 2006. DOI : https://doi.org/10.1145/1143844.1143865

[8] J. Peurifoy, Y. Shen, L. Jing, Y. Yang, F. Cano-Renteria, B. G. DeLacy, ... and M. Soljačić, "Nanophotonic particle simulation and inverse design using artificial neural networks," *Science advances*, *4*(6), eaar4206, 2018. DOI: https://doi.org/10.1126/sciadv.aar4206

[9] S. Kim, M. Jwa, S. Lee, S. Park, and N. Kang, "Deep learning-based inverse design for engineering systems: multi disciplinary design optimization of automotive brakes," *Structural and Multidisciplinary Optimization*, *65*(11), 323, 2022. DOI: https://doi.org/10.1007/s00158-022-03386-8

[10] Z. Liu, D. Zhu, S. P. Rodrigues, K. T. Lee, and W. Cai, "Generative model for the inverse design of metasurfaces," *Nano letters*, *18*(10), 6570-6576, 2018. DOI: https://doi.org/10.1021/acs.nanolett.8b03171

[11] Y. Dan, Y. Zhao, X. Li, S. Li, M. Hu, and J. Hu, "Generative adversarial networks (GAN) based efficient sampling of chemical composition space for inverse design of inorganic materials," *npj Computational Materials*, *6*(1), 1-7, 2020. DOI: https://doi.org/10.1038/s41524-020-00352-0

[12] A. Mall, A. Patil, A. Sethi, and A. Kumar, "A cyclical deep learning based framework for simultaneous inverse an d forward design of nanophotonic metasurfaces," *Scientific reports*, *10*(1), 19427, 2020.

[13] Y. Dong, D. Li, C. Zhang, C. Wu, H. Wang, M. Xin, J. Cheng, and J. Lin, "Inverse design of two-dimensional grap hene/h-BN hybrids by a regressional and conditional GAN," *Carbon*, *169*, 9-16, 2020. DOI: https://doi.org/10.101 6/j.carbon.2020.07.013

[14] W. Ma, F. Cheng, Y. Xu, Q. Wen, and Y. Liu, "Probabilistic Representation and Inverse Design of Metamaterials Based on a Deep Generative Model with Semi-Supervised Learning Strategy," *Advanced Materials*, *31*(35), 1901111, 2019. DOI: https://doi.org/10.1002/adma.201901111

[15] Y. Zhang, and W. Ye,"Deep Learning-based inverse method for layout design," *Structural and Multidisciplinary Optimization*, 60, 527-536, 2019. DOI: https://doi.org/10.1007/s00158-019-02222-w

[16] C. Wang, Y. Fu, K. Deng, and C. Ji, "Functional Response Conditional Variational Auto-Encoders for Inverse Design of Metamaterials," In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021.

[17] W. Jing, LI. Runze, H. E. Cheng, C. Haixin, R. Cheng, Z. Chen, and M. Zhang, "An inverse design method for supercritical airfoil based on conditional generative models," *Chinese Journal of Aeronautics*, *35*(3), 62-74, 2022. DOI: https://doi.org/10.1016/j.cja.2021.03.006

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, ... and Y. Bengio, "Generative adve rsarial nets," *Advances in neural information processing systems*, *27*, 2014.

[19] D. P. Kingma, and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013. DOI: https://doi.org/10.48550/arXiv.1312.6114

[20] M. Mirza, and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014. DOI: https://doi.org/10.48550/arXiv.1411.1784