Inverse Design Nanophotonics Structure Using Stable Diffusion

PHN-300 - Lab-Based Project - Report

Priyanshu Maurya (21122036)
B.Tech Engineering Physics (2021-25)

Supervisor: Prof. Vipul Rastogi
Dept. of Physics, IIT Roorkee

May 7, 2024

Priyanshu Maurya

Prof. Vipul Rastogi

Abstract

Creating tiny structures with specific functions, like changing how light behaves, is hard. We can adjust these structures to do different things by picking certain materials and shapes. But current methods need us to know what materials and shapes to start with, which limits our options. So, we came up with a new way using Stable diffusion. This method learns from lots of pictures showing different materials and shapes to find the best ones for a job, like absorbing certain light. It can suggest many different designs that work similarly but look different. Our method, described in the paper "Global Inverse Design across Multiple Photonic Structure Classes Using Generative Deep Learning", is a big step towards making better materials and devices by figuring out what works best, even if we don't know where to start.

1 Project Objective and Scope

The objective of this project is to create a model capable of producing metasurface structures, including their thickness, plasma frequency, and refractive index, given an absorption spectrum. These generated structures will be instrumental in the design and fabrication of optical devices with tailored functionalities, such as filters, sensors, and photonic circuits.

By automating the process of metasurface design based on desired optical characteristics, this project aims to streamline and expedite the development of advanced photonic technologies for various applications in telecommunications, imaging, and sensing.

2 Understanding Deep Learning

Deep learning is a type of machine learning that uses artificial neural networks, inspired by the human brain's structure. It works by feeding large amounts of data into these networks, allowing them to learn complex patterns and relationships. Following are the steps to create a deep learning model

2.1 Creating Dataset

We have used the dataset used in the paper, which contains 20,000 metasurface unit cell designs derived from seven shape templates. These designs represent MIM and hybrid dielectric structures within specific unit cell sizes. Each design is converted into a 64x64x3 pixel RGB image, with each pixel corresponding to a minimum feature size.

Gaussian filtering is applied to enhance device performance and fabricability. Finite-difference time-domain simulations yield an 800-point absorption spectrum vector for each structure. Low-quality designs are filtered out to maximize the model's effectiveness. During color encoding, material properties such as plasma frequencies for metal resonators and refractive indices for dielectric resonators are used to encode the red and green channels, respectively. Dielectric thickness values are encoded in the blue channel, all normalized from 0 to 255 to support the RGB color scheme (as shown in Fig.1).

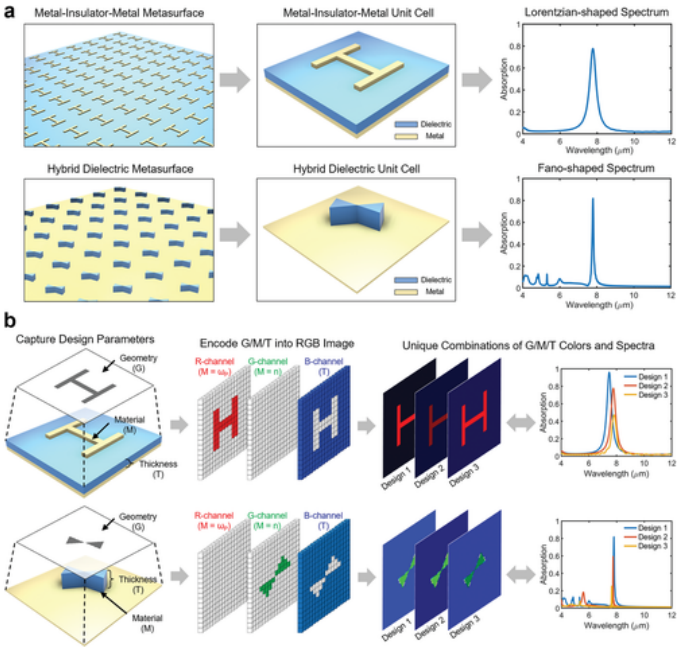

Figure 1: a) Our dataset includes MIM and hybrid dielectric metasurfaces exhibiting Lorentzian-shaped and Fano-shaped absorption responses, respectively. b) We represent different classes of metasurfaces as color-encoded images. These images depict the planar geometries of the metasurfaces, with material properties, thickness values, and metasurface class encoded as varying shades of color. This approach offers greater design flexibility for metasurfaces.

2.2 Model Architecture

The neural network architecture presented, comprises two key components: a spectra encoder and a conditional U-Net model. The spectra encoder, denoted as "spectra encoder", is responsible for processing absorption spectra data. It employs a feedforward neural network structure, consisting of a single hidden layer with a linear transformation followed by a hyperbolic tangent activation function. This network reduces the dimensionality of the input absorption spectrum, which typically consists of 800 points, into a lower-dimensional representation of size 784.

On the other hand, the conditional U-Net model, referred to as "model," is designed for image generation tasks. It is a modified version of the U-Net architecture that incorporates both an input image and a conditional input—in this case, the encoded spectrum—from the spectra encoder. The model architecture comprises encoder and decoder blocks interconnected by skip connections to preserve spatial information during feature extraction and reconstruction. Notably, spatial self-attention blocks are integrated into both the downsampling and upsampling pathways to enhance feature learning and image reconstruction quality.

The input size for the U-Net model is specified as 64x64 pixels with three color channels (RGB images), matching typical image dimensions. The output size is identical to the input size. During operation, the encoded spectrum from the spectra encoder is concatenated with randomly generated noise of the same size as the input image. This combined input is then passed to the conditional U-Net model for generating output images with desired characteristics, as depicted in Figure 2.

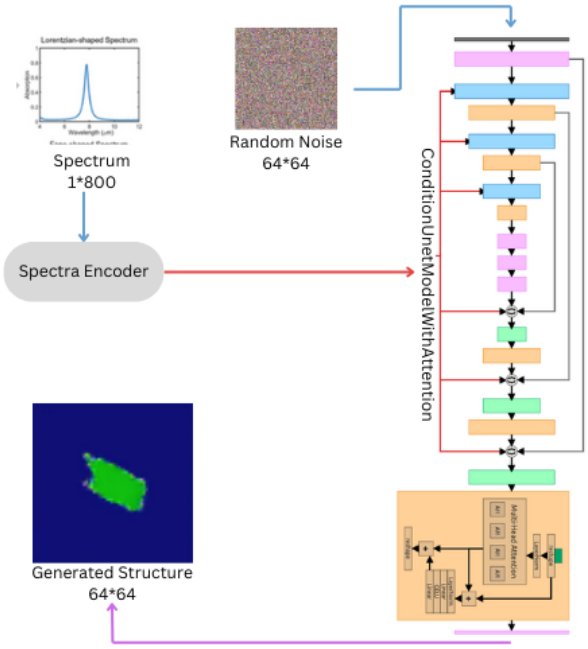

Figure 2: The image illustrates the architecture of the neural network model used in the study. It consists of a spectra encoder, responsible for processing absorption spectra data, and a conditional U-Net model for image generation tasks. The U-Net model incorporates both an input image and a conditional input—the encoded spectrum—into its architecture to generate output images with desired characteristics.

2.3 Choosing loss function

The provided loss function calculates the L1 loss between the predicted noise and the actual noise.

The loss function, denoted as $\mathcal{L}$, is defined as the mean absolute error (L1 loss) between the predicted noise $\hat{N}$ and the actual noise N, given by:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} |N_i - \hat{N}_i|$$

where n is the number of elements in the noise vectors N and $\hat{N}$. The loss function is used to evaluate the discrepancy between the predicted noise and the ground truth noise, providing a measure of how well the model reconstructs the noise in the input data. Minimizing this loss helps in training the model to generate accurate predictions of the noise component.

2.4 Optimizer

We use Adam optimizer,it is a variant of stochastic gradient descent (SGD) that combines adaptive learning rates with momentum. It adapts the learning rate for each parameter based on estimates of the first and second moments of the gradients. Mathematically, the update rule for the parameters θ at each optimization step t is given by:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

where: - η is the learning rate, - $\hat{m}_t$ is the exponentially decaying average of past gradients, - $\hat{v}_t$ is the exponentially decaying average of past squared gradients, and - ϵ is a small constant to prevent division by zero.

The Adam optimizer is popular due to its efficiency in training deep neural networks and its ability to automatically adapt learning rates for different parameters. It typically converges faster and requires less tuning of hyperparameters compared to traditional SGD variants.

While Adam is widely used and often performs well in practice, other optimizers such as SGD with momentum, RMSprop, and AdaGrad also have their advantages and may be preferred depending on the specific characteristics of the dataset and the neural network architecture. However, Adam is commonly chosen as a default optimizer due to its robust performance across a wide range of scenarios and its ease of use.

2.5 Training

2.5.1 Training a Stable Diffusion Model

Stable Diffusion models, a subset of non-autoregressive diffusion models, learn to generate high-quality images by progressively adding and removing noise from an initial latent noise vector. The training process involves two main phases: forward diffusion and reverse diffusion.

Mathematically, the forward diffusion process initializes with an initial latent noise vector z_0 and iteratively adds noise at each timestep t according to:

$$z_{t+1} = \sqrt{\alpha_t} z_t + \epsilon_t$$

where α_t represents the noise scaling factor at timestep t, and ϵ_t is a sample from a standard normal distribution.

The reverse diffusion process aims to recover the original image from the final noisy latent representation z_T. It iteratively removes noise in reverse order, calculated as:

$$z_t = \frac{z_{t+1} - \epsilon_t}{\sqrt{\alpha_t}}$$

Efficient training of Stable Diffusion models requires careful time sampling strategies to balance computational cost and performance. Techniques like the DDPM Sampler prioritize timesteps where the model struggles to remove noise efficiently.

In summary, Stable Diffusion models leverage forward and reverse diffusion processes, along with strategic time sampling, to learn to generate high-quality images from noisy latent representations.

2.5.2 Training of created model

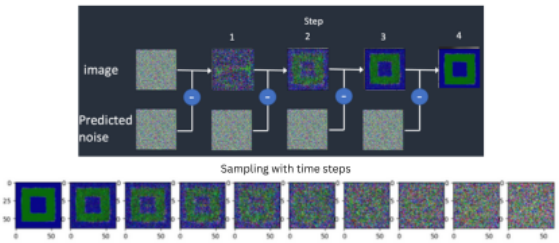

Figure 3: The image illustrates the model prediction with time sampling.

1. **Input Processing:**
    - The absorption spectrum is passed through the spectra encoder to obtain the latent spectrum vector for conditioning.

- Let spectra denote the absorption spectrum and latent_spectrum represent the output latent spectrum vector.

2. **Noise Addition:**

   - A structure corresponding to the absorption spectrum is selected.
   - Gaussian noise is added to the structure at each timestep, following the principles of stable diffusion training.
   - Let structure denote the selected structure and noise represent the added noise at each timestep.

3. **Model Input Preparation:**

   - The noisy structure along with the latent spectrum vector is fed into the model as conditional input.
   - The input to the model is represented as (noisy_structure, latent_spectrum).

4. **Model Training:**

   - The model's objective is to predict the added noise given the conditional input.
   - Let noise_pred denote the predicted noise.
   - The model output target is to minimize the Mean Squared Error (MSE) loss between the predicted noise and the actual noise.

Mathematically, the training procedure can be summarized as follows:

$$\text{noise\_pred} = \text{model}(\text{noisy\_structure}, \text{latent\_spectrum})$$

$$\text{loss} = \text{MSE}(\text{noise}, \text{noise\_pred})$$

Finally, the model parameters are updated using the Adam optimizer to minimize the loss:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \text{loss}$$

where η represents the learning rate, θ denotes the model parameters, and ∇_θloss denotes the gradient of the loss with respect to the model parameters.

3 Results

The results of the training process reveal several key insights into the performance of the model. The model was trained for 45 hours, during which it exhibited convergence in the loss curve. However, it is notable that further training iterations could potentially lead to additional convergence improvements. The loss curve, depicted as Loss versus training steps, showcases the gradual reduction in loss over the training duration, indicating the model's learning progress. Additionally, the final generated structures, corresponding to the provided absorption spectra, are presented in figures alongside their respective spectra. These visual representations provide a tangible demonstration of the model's ability to generate structures based on input spectral information. Overall, the results underscore the effectiveness of the training process in producing coherent structures aligned with the given absorption spectra.

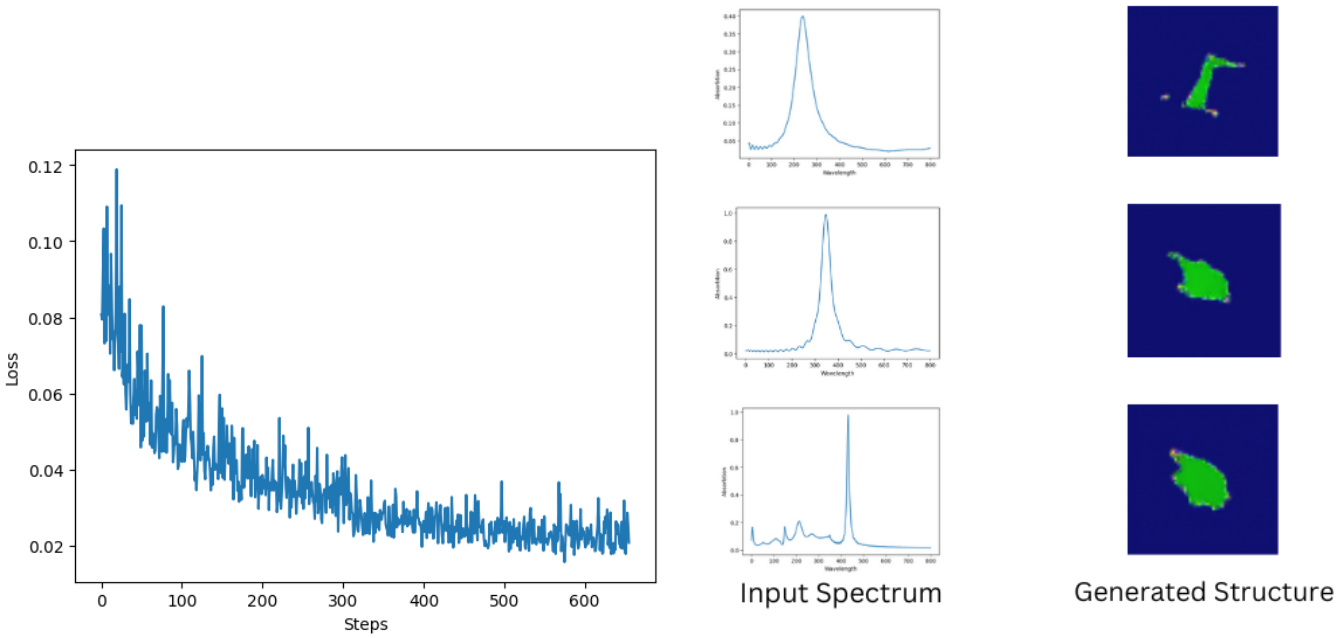

Figure 4: a) Loss vs steps curve b)Input spectrum vs Generated Structure

4 Conclusion

The results demonstrate that employing Stable Diffusion leads to faster convergence compared to the Conditional GAN method utilized in the paper "Global Inverse Design across Multiple Photonic Structure Classes Using Generative Deep Learning." However, it's noteworthy that the generation time increases significantly due to the sampling method employed, although this aspect hasn't been a significant part of the discussion thus far. Utilizing Stable Diffusion proves advantageous in learning the complex structural properties efficiently.

5 Discussion on improvements

Several avenues for enhancing the model's performance and capabilities have been identified. Firstly, refining the architecture of the spectral encoder presents an opportunity for improvement. Instead of a simple ANN model, exploring more sophisticated architectures could potentially enhance the model's ability to extract and encode spectral information effectively. Additionally, increasing the training time beyond the current duration could lead to further improvements in convergence and overall model performance. Moreover, augmenting the input images by adding more channels, such as incorporating material properties as additional channels, could provide the model with richer information for better inference and decision-making. These proposed enhancements offer promising directions for advancing the model's effectiveness and expanding its capabilities in photonic structure design.

References

[1] Christopher Yeung, Ryan Tsai, Benjamin Pham, Brian King, Yusaku Kawagoe, David Ho, Julia Liang, Mark W. Knight, Aaswath P. Raman. *Global Inverse Design across Multiple Photonic Structure Classes Using Generative Deep Learning*. Available online: https://onlinelibrary.wiley.com/doi/10.1002/adom.202100548

[2] Jonathan Ho, Ajay Jain, Pieter Abbeel *Denoising Diffusion Probabilistic Models*. `arXiv preprint arXiv:2006.11239`, 2020. https://onlinelibrary.wiley.com/doi/10.1002/adom.202100548

[3] Abdourahman Khaireh-Walieh, Denis Langevin, Pauline Bennet, Olivier Teytaud, Antoine Moreau, Peter R. Wiecha *A newcomer's guide to deep learning for inverse design in nano-photonics*. `arXiv preprint arXiv:2307.08618`, 2023. Available online: https://arxiv.org/abs/2307.08618

[4] Dataset and Code *Github link:*. https://github.com/Raman-Lab-UCLA/Multiclass$_M etasurface_I nverseDesign$