Semi-Supervised Deep Learning Model for Efficient Computation of Optical Properties of Suspended-Core Fibers

Gao Wang 1, Sufen Ren 2, Shuna Li 3, Shengchao Chen 2,* and Benguo Yu 4

1 State Key Laboratory of Dynamic Measurement Technology, North University of China, Taiyuan 030051, China
2 School of Information and Communication Engineering, Hainan University, Haikou 570228, China
3 School of Information and Communication Engineering, North University of China, Taiyuan 030051, China
4 School of Biomedical Information and Engineering, Hainan Medical University, Haikou 571199, China
* Correspondence: scchen@hainanu.edu.cn

Abstract: Suspended-core fibers (SCFs) are considered the best candidates for enhancing fiber non-linearity in mid-infrared applications. Accurate modeling and optimization of its structure is a key part of the SCF structure design process. Due to the drawbacks of traditional numerical simulation methods, such as low speed and large errors, the deep learning-based inverse design of SCFs has become mainstream. However, the advantage of deep learning models over traditional optimization methods relies heavily on large-scale a priori datasets to train the models, a common bottleneck of data-driven methods. This paper presents a comprehensive deep learning model for the efficient inverse design of SCFs. A semi-supervised learning strategy is introduced to alleviate the burden of data acquisition. Taking SCF's three key optical properties (effective mode area, nonlinear coefficient, and dispersion) as examples, we demonstrate that satisfactory computational results can be obtained based on small-scale training data. The proposed scheme can provide a new and effective platform for data-limited physical computing tasks.

Keywords: suspended-core fiber; inverse design; semi-supervised deep learning; optical properties

1. Introduction

Suspended-core fiber (SCF) is a microstructured fiber (MOF) that is considered an excellent alternate choice for enhancing the nonlinear properties of optical fibers [1,2]. Compared to other microstructured fibers, SCF is easier to fabricate and can achieve higher numerical apertures (NA). Therefore, it is widely used in various fiber laser applications, such as Raman fiber lasers [3], Brillouin fiber lasers [4], etc. In addition, its application is also extended to high-capacity networks [5].

Accurate modeling and optimization of SCF structures usually rely on numerical computational methods such as finite difference, finite element methods [6], block iterative frequency domain methods [7]. Plane wave expansion [8]. However, large-scale iterative analysis is often required to obtain more accurate results in modeling and optimizing fiber optic structures. In addition, the large number of structural design parameters makes each round of iterative analysis time-consuming, and complex structures require multiple simulations to optimize the design. These requirements seriously reduce the efficiency of numerical methods and pose a serious challenge to traditional modeling and optimization methods.

With the rapid development of artificial intelligence (AI), especially the gradual improvement of deep learning (DL) techniques, researchers are actively seeking to apply DL techniques to solve challenging tasks, including materials science [9], chemistry [10], laser physics [11], particle physics [12], and quantum mechanics [13]. DL model that, as a typical data-driven approach, can learn the respective complex nonlinear relationships in a given dataset and abstract their relationships to form a solution strategy through a
computational model consisting of multiple layers of data processing units. This avoids
the high cost and inefficiency caused by human intervention in solution computation and
the direct interaction between the fundamental physical laws in traditional optimization.
As DL techniques spread in various fields, research on optics or optical systems based
on DL has been explored and refined. On the one hand, a well-trained DL model can be
utilized as a fast solver to predict a physical quantity, such as in fiber-optic demodulation
systems [14,15], optical computational imaging [16], and biomedical engineering [17]. In
addition, it can be seen as an optimization tool without expert empirical intervention
for applications in the inverse design of optical materials, such as metamaterials [18–20],
integrated photonics [21], plasmonics [22–24], etc.

DL techniques have now been widely applied to the problem of solving optical
properties during the inverse design of MOFs. Da Silva Ferreira et al. combined a multilayer
perceptron (MLP) and an extreme learning machine-artificial neural network (ELM-ANN)
to calculate photonic crystal (PCs) for dispersion relations [25]. Chugh et al. introduced
an ANN model to achieve fast computation of various optical properties of PC fibers
(PCF) [26]. Yuan et al. applied a back-propagation neural network (BPNN) to accelerate
the calculation process of SCF’s optical properties [27]. Without exception, however, the
premise that these works guarantee superior performance requires a large-scale a priori
data set to train the model. However, collecting and labeling the dataset is laborious and
tedious. Therefore, exploring an effective deep learning model that can mine the nonlinear
physical relationship between SCF geometric and optical properties in minimal data, is
crucial in the reverse design process of MOFs.

In this paper, we proposed an efficient deep learning model for optimizing the optical
parameter solving process for the inverse design of SCF structures with limited available a
priori data. The proposed model consists of a Generative Adversarial Network (GAN) for
data augmentation, and a back-propagation neural network (BPNN) cascaded for optical
properties computation. The model employs a semi-supervised learning strategy that
performs online data augmentation during training based on pre-collected data input to
the computation model. The computation model calculates the optical properties of the
designed SCF structure, including the effective mode area, nonlinear system, and dispersion.
Experimental results show that the proposed model can obtain efficient and highly accurate
optical property calculations with extremely limited a priori data. In conclusion, the model
creates a new framework and platform for the efficient design of optical fiber structures
with arbitrary microstructures and also provides reliable support for DL-based physical
computational problems under minimal data sets.

This work is organized as follows. Section 2 describes the design and theoretical
analysis of the SCF model in this work. Section 3 describes the structure and configuration
of the proposed semi-supervised deep learning model for computing the optical properties
of the SCF. Section 4 compares the predicted and actual values of the optical properties.
Section 5 concludes this work.

2. Suspended-Core Fiber Design and Theoretical Analysis

2.1. Structure Design of SCFs

The structures of the SCFs used in this work are reproduced from reference [28].
Figure 1a–c show the cross-sectional schematics of the three-, four-, and six-bridges (cantilever) SCFs, respectively, and the corresponding Figure 1c shows the fundamental optical
mode field distributions of the three-, four-, and six-bridges (cantilever) SCFs. The diameter,
width of the cantilever, and the number of the cantilever of SCFs structure are denoted by
d, W, and n, respectively.
2.2. Basic Theoretical Analysis

The real part of the RI of the chalcogenide materials can be calculated from Sellmeier equation as shown in Equation (1) [29].

\[ n^2(\lambda) = 1 + \sum_i \frac{A_i \lambda^2}{\lambda_i^2 - \lambda^2}, \]  

where \( \lambda \) denotes the wavelength in free space, \( A_i \) and \( \lambda_i^2(i = 1, 2, 3) \) are the constant of substrate materials of SCF. \( As_2S_3 \) was used as the core filling material for SCFs, and its material constants: \( A_1, A_2, A_3, \lambda_1^2, \lambda_2^2, \) and \( \lambda_3^2 \) are 1.8983, 1.9222, 0.8765, 0.0225, 0.0625 and 0.1225, respectively.

The effective index of the fundamental mode propagation of the SCF optical field distribution can be obtained by a finite element model (FEM). With the known effective index \( n_{\text{eff}} \), the dispersion of the SCF as a function of wavelength can be expressed by Equation (2).

\[ D(\lambda) = -\frac{\lambda}{c} \frac{d^2}{d\lambda^2} \text{Re}[n_{\text{eff}}(\lambda)], \]  

where \( \text{Re}[n_{\text{eff}}(\lambda)] \) is the real part of the effective index, and \( c \) is the light velocity in free space.
The nonlinear coefficient $\gamma$ can be calculated by Equation (3) [29].

$$\gamma = \left(\frac{2\pi}{\lambda}\right) \cdot \left(\frac{n_2}{A_{\text{eff}}}\right),$$  (3)

where $c$ is the light speed in free space, $n_2$ is the nonlinear refractive index and $n_2$ of $As_2S_3$ is $4.2 \times 10^{-18}$ m$^2$/W, the $A_{\text{eff}}$ denotes the effective area of the propagation mode in the SCF and is defined as Equation (4) [27].

$$A_{\text{eff}} = \left(\int_{\text{NLR}} |E(x,y)|^2 \, dx \, dy \right)^2 / \int_{\text{NLR}} |E(x,y)|^4 \, dx \, dy,$$  (4)

where $E(x,y)$ is the distribution of the optical field across the SCF cross-section, and NLR stands for nonlinear material region.

3. Semi-Supervised Deep Learning Model with Suspended-Core Fiber

3.1. Data Pre-Processing

The data pre-processing is necessary to avoid the problem of slow convergence before the training dataset is fed into the neural network model. The ‘Min-Max’ normalization was used to linearly transform the original data and map the original data to the interval [0, 1], which can eliminate the adverse effects caused by singular samples in the characteristic data and improve the convergence speed and performance of the model. The transformation function can be defined as Equation (5).

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)}$$  (5)

where $x$ represents the original eigenvalue of SCF and $x^*$ represents the normalized result.

3.2. Model Design for Optical Properties Calculation

The semi-supervised deep learning model used to compute the optical properties of the SCF is shown in Figure 1. The data processing process of the collation model is shown in Figure 2a, where the original data set is input to the GAN after the data mentioned above pre-processing, and the network generates a series of generative data with a distribution close to the original data. The structure of the GAN is shown in Figure 2b, which is an unsupervised deep learning model for generating data, consisting of a separate generative model ($G$) and discriminative model ($D$). The objective of the $G$ is to generate actual data to deceive the $D$ as much as possible, while the objective of the $D$ is to distinguish the data generated by the $G$ from the actual data as much as possible. The training process is a dynamic game process. The original data, instead of random noise, was used as the output of the $G$ to make the generated data more realistic. The original data is fed to the $D$ together with the generated data. The $G$ is a typical encoder-decoder structure. In contrast, the $D$ outputs a data truth score to encourage the $G$ to generate actual data as much as possible, and their structures are shown in the red dashed box in Figure 2b. Then the generated data were fed to the BPNN for computation with the original data, and the network structure is shown in Figure 2c.

GAN and BPNN are entirely independent in the training phase, where the training process of GAN is a “Min-Max” dynamic game process. The loss function of the network can be established as Equation (6), while the training process of BPNN is an error back propagation process, and MSE is chosen as its loss function, which can be defined as Equation (7).

$$\text{Loss}_{\text{GAN}} = \min_G \max_D E_{x \sim p_{\text{data}}} \left[\log D(x)\right] + E_{z \sim p_z} \left[\log(1 - D(G(z)))\right],$$  (6)

$$\text{Loss}_{\text{BPNN}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$  (7)
where \( Z \) is the random noise fed into the \( G \), \( n \) represents the total number of samples, and \( y_i \) is the predicted value of the neural network model, \( \hat{y}_i \) is the actual value.

To ensure that the parameters are approximate to nonlinear functions in the error propagation process, \( Sigmoid [30] \) was used as the activation function between hidden layers, and the function can be formulated as Equation (8):

\[
Sigmoid(x) = \frac{1}{1 + e^{-x}}, \quad (8)
\]

where \( x \) is the input of the hidden node.

**Figure 2.** (a) Computational flow of the proposed model. (b) Processing flow and structure diagram of a generative adversarial network for data augmentation, G is a typical encoder-decoder structure with 512, 256, 128, 256, 512 neurons per layer (full-connection layer, from left to right), and D with 512, 256, 128, 64, 32, 1 neuron per layer (full-connection layer, from left to the right). (c) The structure of the back-propagation neural network for optical properties calculation has three hidden layers with 512 neurons each, one input layer and one output layer, the input layer corresponds to the four geometric features of the SCF, and the output layer corresponds to the optical properties to be calculated.

### 4. Experiments

The number of cantilevers of the SCFs used in the experiments was 3, 4, and 6, and each SCF contained 14 different core radii (0.4 \( \mu m \), 0.45 \( \mu m \), 0.5 \( \mu m \), 0.55 \( \mu m \), 0.6 \( \mu m \), 0.7 \( \mu m \), 0.75 \( \mu m \), 0.8 \( \mu m \), 0.85 \( \mu m \), 0.9 \( \mu m \), 0.95 \( \mu m \), 1 \( \mu m \), 1.05 \( \mu m \), 1.1 \( \mu m \)) and 6 cantilever widths (0.6 \( \mu m \), 0.7 \( \mu m \), 0.8 \( \mu m \), 0.9 \( \mu m \), 1.0 \( \mu m \), 1.1 \( \mu m \)). To build the dataset, the relevant optical features were calculated using a permutation form (e.g., \( n = 3, r = 0.4 \mu m, W = 0.07 \mu m \)). The dataset is divided into training, validation, and test sets in the ratio of 6:2:2 with no overlap between the three. During the training process, the learning rates were adjusted to 0.01, 0.005, and 0.001, respectively, and SGD, Adam, and Adagrad were selected as optimizers to explore the appropriate training configuration (e.g., Optimizer: Adam, Learning Rate: 0.001) to ensure the performance of the model. MSE was used as the loss function during model training, and MSE, RMSE, and MAE were chosen as error evaluation metrics to comprehensively assess the model’s effectiveness, which can be formulated as Equation (9), Equation (10) and Equation (11), respectively.

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \quad (9)
\]

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \quad (10)
\]
$$MAE = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|,$$
where $n$ represents the total number of samples, $y_i$ is the predicted value, and $\hat{y}_i$ denotes the actual value.

4.1. Model Training Process

In the training process, MSE was used as the loss function. We calculated the MSE of the training and verification process of different learning rates and optimizers (SGD, Adagrad, Adam), as shown in Table 1. The similar MSEs on the validation dataset indicate that the models trained by all three optimizers can achieve satisfactory results. Detailed prediction results and analysis of the effective mode area, nonlinear coefficients, and dispersion of the SCF will be presented in the following.

Table 1. Average mean square deviation of a neural network model of training process and verification process under different learning rates and optimizers.

| Optimizer | LR   | MSE (Training, a.u.) | MSE (Validation, a.u.) |
|-----------|------|----------------------|------------------------|
| SGD       | 0.01 | 0.0061               | 0.0043                 |
| SGD       | 0.005| 0.0058               | 0.0043                 |
| SGD       | 0.001| 0.0064               | 0.0043                 |
| Adagrad   | 0.01 | 0.0054               | 0.0043                 |
| Adagrad   | 0.005| 0.0069               | 0.0046                 |
| Adagrad   | 0.001| 0.0008               | 0.0044                 |
| Adam      | 0.01 | 0.0108               | 0.0044                 |
| Adam      | 0.005| 0.0057               | 0.0046                 |
| Adam      | 0.001| 0.0052               | 0.0043                 |

4.2. Effective Mode Area

The waveguide characteristics played by SCF in mid-infrared applications depend mainly on its effective mode area. This relationship can be seen in Equation (3). The smaller its mode area, the greater the advantage of the fiber in nonlinear applications. Conversely, a larger effective mode area can also enhance the effectiveness of SCF in optical power transmission applications.

This section shows the proposed model’s results in predicting the SCF’s effective mode area ($A_{\text{eff}}$). The predicted and actual values for each sample in the test dataset are compared, and the performance of the obtained model with different training configurations is analyzed, as shown in Table 2. Among them, the SGD optimizer and the Adam optimizer have better performance.

Based on the above conclusions, the models trained by Adam and SGD were selected for testing to compare the error between the actual and predicted values of $A_{\text{eff}}$, as shown in Figure 3. As these points get closer to $y$ equal to 0, the residual between the predicted and actual values is more diminutive. Their proximity to the origin is positively correlated with the model’s performance. The model with Adam as the optimizer works best in the effective pattern area prediction for SCF when the learning rate is 0.001. This means that the best-performing model can be obtained with this set of training configurations (Optimizer: Adam, LR: 0.001).
Table 2. Performance evaluation results for predictions of $A_{\text{eff}}$ on the test dataset with models trained with different training configuration.

| Optimizer | LR  | MSE ($\times 10^{-6}$, a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-----------|-----|-----------------------------|-------------|------------|
| SGD       | 0.01| 1.63                        | 0.0004      | 0.0003     |
|           | 0.005| 1.91                        | 0.0014      | 0.0014     |
|           | 0.001| 3.78                        | 0.0002      | 0.0002     |
| Adagrad   | 0.01| 2.38                        | 0.0049      | 0.0049     |
|           | 0.005| 4.92                        | 0.0022      | 0.0022     |
|           | 0.001| 8.01                        | 0.0028      | 0.0028     |
| Adam      | 0.01| 2.17                        | 0.0005      | 0.0004     |
|           | 0.005| 1.13                        | 0.0011      | 0.0009     |
|           | 0.001| 0.51                        | 0.0002      | 0.0002     |

Figure 3. Comparison of the residual between the predicted value and the actual value of $A_{\text{eff}}$ in the test data set of the model trained by each algorithm under different learning rates in SGD optimizer and Adam optimizer. (a) Adam optimizer, LR: 0.01, 0.001, 0.005 (b) SGD optimizer, LR: 0.01, 0.001, 0.005.

Figures 4–6 show the predicted values of $A_{\text{eff}}$ during wavelength variation compared to the actual values for models trained by different optimizers for a different number of cantilevers, core radii, and cantilever widths. The $A_{\text{eff}}$ corresponding to these parameters has never been recorded or provided. Figures 4–6 show the $A_{\text{eff}}$ predictions of the model when the number of cantilevers is 3, 4, and 6, respectively. As can be seen in the $A_{\text{eff}}$ prediction results for the SCF with several cantilevers of 3, the network trained by the Adam optimizer predicts best when the wavelength is between 3 $\mu$m and 4 $\mu$m. When the number of cantilever beams is 4, the predicted value of $A_{\text{eff}}$ by the network trained by Adam is closer to the actual value in the wavelength range of 1 $\mu$m–2 $\mu$m. When the number of cantilever beams is 6, in the wavelength range of 2 $\mu$m to 3 $\mu$m, the best network prediction trained by Adam is still obtained.

Combined with Figures 4–6, the predicted value of $A_{\text{eff}}$ in the test dataset of Adam trained model is closer to the real value. This can be proved by the quantitative analysis in Table 3 that our proposed model can work well in predicting $A_{\text{eff}}$ of SCF.
Figure 4. Comparison of the predicted and actual values of $A_{\text{eff}}$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 3). (a) $r = 0.4 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (b) $r = 0.4 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (c) $r = 0.4 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (d) $r = 0.4 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$. (e) $r = 0.5 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (f) $r = 0.5 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (g) $r = 0.5 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (h) $r = 0.5 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$. (i) $r = 0.6 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (j) $r = 0.6 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (k) $r = 0.6 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (l) $r = 0.6 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$.

Figure 5. Comparison of the predicted and actual values of $A_{\text{eff}}$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 4). (a) $r = 0.7 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (b) $r = 0.7 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (c) $r = 0.7 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (d) $r = 0.7 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$. (e) $r = 0.8 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (f) $r = 0.8 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (g) $r = 0.8 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (h) $r = 0.8 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$. (i) $r = 0.9 \ \mu\text{m}, W = 0.06 \ \mu\text{m}$. (j) $r = 0.9 \ \mu\text{m}, W = 0.07 \ \mu\text{m}$. (k) $r = 0.9 \ \mu\text{m}, W = 0.08 \ \mu\text{m}$. (l) $r = 0.9 \ \mu\text{m}, W = 0.09 \ \mu\text{m}$. 
Figure 6. Comparison of the predicted and actual values of $A_{eff}$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 6). (a) $r = 0.9 \mu m$, $W = 0.08 \mu m$. (b) $r = 0.9 \mu m$, $W = 0.09 \mu m$. (c) $r = 0.9 \mu m$, $W = 0.1 \mu m$. (d) $r = 0.9 \mu m$, $W = 0.11 \mu m$. (e) $r = 1 \mu m$, $W = 0.08 \mu m$. (f) $r = 1 \mu m$, $W = 0.09 \mu m$. (g) $r = 1 \mu m$, $W = 0.1 \mu m$. (h) $r = 1 \mu m$, $W = 0.11 \mu m$. (i) $r = 1.1 \mu m$, $W = 0.08 \mu m$. (j) $r = 1.1 \mu m$, $W = 0.09 \mu m$. (k) $r = 1.1 \mu m$, $W = 0.1 \mu m$. (l) $r = 1.1 \mu m$, $W = 0.11 \mu m$.

Table 3. Average of each model on evaluation metrics when predicting $A_{eff}$ in the test dataset under different combinations of parameters.

| Optimizer   | MSE ($\times 10^{-6}$, a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-------------|-------------------------------|-------------|------------|
| SGD         | 0.0704                        | 0.00066     | 0.00062    |
| Adagrad     | 5.10                          | 0.0033      | 0.0033     |
| Adam        | 0.467                         | 0.00058     | 0.0005     |

4.3. Nonlinear Coefficient

This section uses the well-trained model to predict the nonlinear coefficient ($\gamma$) directly related to the performance presented by the SCFs in nonlinear applications. Table 4 shows the quantitative analysis of the prediction results for the $\gamma$ value. The model trained with the SGD optimizer and Adagrad optimizer could achieve better results. Figure 7 shows the magnitude of the residuals between the predicted and actual $\gamma$ values of the test samples for the models trained with different LRs by the SGD optimizer and the Adagrad optimizer. The $\gamma$ predicted values of the SGD-trained model are closest to the actual values. Combined with Table 4 and Figure 7, the model trained by the SGD optimizer with 0.001 LR shows the best results in terms of evaluation metrics, which means that the trained model configured by this parameter achieves superior performance in terms of $\gamma$ prediction.
Table 4. Performance evaluation results for predictions of $\gamma$ on test datasets with models trained with different optimizers and different LRs.

| Optimizer | LR  | MSE ($\times 10^{-6}$, a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-----------|-----|------------------------------|-------------|------------|
| SGD       | 0.01| 2.87                         | 0.0017      | 0.0012     |
|           | 0.005| 1.85                         | 0.0014      | 0.0010     |
|           | 0.001| 1.2                          | 0.0011      | 0.0007     |
| Adagrad   | 0.01| 2.39                         | 0.0015      | 0.0011     |
|           | 0.005| 7.82                         | 0.0028      | 0.0024     |
|           | 0.001| 1.66                         | 0.0013      | 0.0010     |
| Adam      | 0.01| 6.92                         | 0.0026      | 0.0025     |
|           | 0.005| 8.06                         | 0.0028      | 0.0025     |
|           | 0.001| 4.37                         | 0.0021      | 0.0016     |

Figure 7. Comparison of the residual between the predicted value and the actual value of $\gamma$ in the test data set of the model trained by each algorithm under different learning rates in SGD optimizer and Adam optimizer. (a) SGD optimizer, LR: 0.01, 0.001, 0.005 (b) Adagrad optimizer, LR: 0.01, 0.001, 0.005.

Figures 8–10 compare the predicted and actual values of the $\gamma$ value for different combinations of the number of cantilevers, core radius, and cantilever width for the models trained by different optimizers during the wavelength variation. Figures 8–10 show the comparison results when the number of cantilever beams is 3, 4, and 6, respectively. The $\gamma$ values gradually decrease as the wavelength increases, and the predicted values of $\gamma$ of the network model trained by SGD are closer to the actual values. In addition, the network trained by SGD shows higher prediction performance than other networks in the wavelength range of 2 $\mu$m to 3 $\mu$m.

The proposed model can predict $\gamma$ well during the linear variation of wavelength. Combining Figures 8–10, the predicted values of $\gamma$ in the SGD training model test dataset (compared with other optimizers) are closer to the actual values. Moreover, the quantitative analysis in Table 5 proves that our proposed model can better predict the $\gamma$ value of SCF.
Figure 8. Comparison the predicted and actual values of $\gamma$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 3). (a) $r = 0.4\ \mu m$, $W = 0.06\ \mu m$. (b) $r = 0.4\ \mu m$, $W = 0.07\ \mu m$. (c) $r = 0.4\ \mu m$, $W = 0.08\ \mu m$. (d) $r = 0.4\ \mu m$, $W = 0.09\ \mu m$. (e) $r = 0.5\ \mu m$, $W = 0.06\ \mu m$. (f) $r = 0.5\ \mu m$, $W = 0.07\ \mu m$. (g) $r = 0.5\ \mu m$, $W = 0.08\ \mu m$. (h) $r = 0.5\ \mu m$, $W = 0.09\ \mu m$. (i) $r = 0.6\ \mu m$, $W = 0.06\ \mu m$. (j) $r = 0.6\ \mu m$, $W = 0.07\ \mu m$. (k) $r = 0.6\ \mu m$, $W = 0.08\ \mu m$. (l) $r = 0.6\ \mu m$, $W = 0.09\ \mu m$.

Figure 9. Comparison the predicted and actual values of $\gamma$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 4). (a) $r = 0.7\ \mu m$, $W = 0.06\ \mu m$. (b) $r = 0.7\ \mu m$, $W = 0.07\ \mu m$. (c) $r = 0.7\ \mu m$, $W = 0.08\ \mu m$. (d) $r = 0.7\ \mu m$, $W = 0.09\ \mu m$. (e) $r = 0.8\ \mu m$, $W = 0.06\ \mu m$. (f) $r = 0.8\ \mu m$, $W = 0.07\ \mu m$. (g) $r = 0.8\ \mu m$, $W = 0.08\ \mu m$. (h) $r = 0.8\ \mu m$, $W = 0.09\ \mu m$. (i) $r = 0.9\ \mu m$, $W = 0.06\ \mu m$. (j) $r = 0.9\ \mu m$, $W = 0.07\ \mu m$. (k) $r = 0.9\ \mu m$, $W = 0.08\ \mu m$. (l) $r = 0.9\ \mu m$, $W = 0.09\ \mu m$. 
Figure 10. Comparison the predicted and actual values of $\gamma$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 6). (a) $r = 0.9 \, \mu m$, $W = 0.08 \, \mu m$. (b) $r = 0.9 \, \mu m$, $W = 0.09 \, \mu m$. (c) $r = 0.9 \, \mu m$, $W = 0.1 \, \mu m$. (d) $r = 0.9 \, \mu m$, $W = 0.11 \, \mu m$. (e) $r = 1 \, \mu m$, $W = 0.08 \, \mu m$. (f) $r = 1 \, \mu m$, $W = 0.09 \, \mu m$. (g) $r = 1 \, \mu m$, $W = 0.1 \, \mu m$. (h) $r = 1 \, \mu m$, $W = 0.11 \, \mu m$. (i) $r = 1.1 \, \mu m$, $W = 0.08 \, \mu m$. (j) $r = 1.1 \, \mu m$, $W = 0.09 \, \mu m$. (k) $r = 1.1 \, \mu m$, $W = 0.1 \, \mu m$. (l) $r = 1.1 \, \mu m$, $W = 0.11 \, \mu m$.

Table 5. Average of each model on evaluation metrics when predicting $\gamma$ in the test dataset under different combinations of parameters.

| Optimizer | MSE ($\times10^{-6}$, a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-----------|-----------------------------|------------|------------|
| SGD       | 1.97                        | 0.0014     | 0.001      |
| Adagrad   | 3.96                        | 0.0019     | 0.0015     |
| Adam      | 6.45                        | 0.0025     | 0.0022     |

4.4. Dispersion

Dispersion ($D$) is essential in generating SCF-based supercontinuum spectra. Table 6 shows the quantitative analysis of the models in terms of $D$-value prediction, where the models trained by SGD and Adam show good performance. Figure 11 shows the magnitude of the residuals between the predicted and actual values of $D$ in the test set of the model trained by the SGD optimizer and Adam optimizer at different LRs. The model trained by Adam with an LR of 0.001 achieves the best performance.
Table 6. Performance evaluation results for predictions of $D$ on the test dataset with models trained with different optimizers and different LRs.

| Optimizer | LR   | MSE (a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-----------|------|------------|-------------|------------|
| SGD       | 0.01 | 0.01366    | 0.11686     | 0.06026    |
|           | 0.005| 0.01421    | 0.11920     | 0.08287    |
|           | 0.001| 0.01349    | 0.11614     | 0.06438    |
| Adagrad   | 0.01 | 0.01455    | 0.12061     | 0.05501    |
|           | 0.005| 0.01350    | 0.11617     | 0.06380    |
|           | 0.001| 0.01367    | 0.11692     | 0.07498    |
| Adam      | 0.01 | 0.01354    | 0.11638     | 0.06264    |
|           | 0.005| 0.01347    | 0.11604     | 0.06417    |
|           | 0.001| 0.01345    | 0.11602     | 0.06414    |

Figure 11. Comparison of the residual between the predicted value and the actual value of $D$ in the test data set of the model trained by each algorithm under different learning rates in SGD optimizer and Adam optimizer. (a) SGD optimizer, LR: 0.01, 0.001, 0.005 (b) Adam optimizer, LR: 0.01, 0.001, 0.005.

Figures 12–14 compare the predicted and actual values of the $D$ value for different combinations of the number of cantilevers, core radius, and cantilever width for the models trained by different optimizers during the wavelength variation. Figures 12–14 show the comparison results when the number of cantilevers is 3, 4, and 6, respectively. The network trained by the Adam optimizer makes a better prediction of the $D$ value of the SCF as the wavelength varies linearly, implying that its prediction is closer to the actual value than the other two optimizers. However, during the model’s prediction of the $D$ value, its predicted value is smoother with the wavelength, probably because the $D$ value shows severe irregularities in the range of wavelength variation. The model is less sensitive to such nonlinear irregularities. Even though the proposed model can present superior prediction results, the problem still poses an obstacle to its excellent prediction results. To address this problem, it is straightforward to consider increasing the number of iterations and adjusting the nonlinear activation function to solve it.

Combining Figures 12–14, we can see that the predicted values of $D$ in the test dataset for the Adam-trained model (compared to the other optimizers) are much closer to the actual values. Combined with the quantitative analysis in Table 7, we can prove that the proposed model can predict $D$ effectively.
Figure 12. Comparison the predicted and actual values of $D$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 3).  

(a) $r = 0.4 \, \mu m$, $W = 0.06 \, \mu m$.  
(b) $r = 0.4 \, \mu m$, $W = 0.07 \, \mu m$.  
(c) $r = 0.4 \, \mu m$, $W = 0.08 \, \mu m$.  
(d) $r = 0.4 \, \mu m$, $W = 0.09 \, \mu m$.  
(e) $r = 0.5 \, \mu m$, $W = 0.06 \, \mu m$.  
(f) $r = 0.5 \, \mu m$, $W = 0.07 \, \mu m$.  
(g) $r = 0.5 \, \mu m$, $W = 0.08 \, \mu m$.  
(h) $r = 0.5 \, \mu m$, $W = 0.09 \, \mu m$.  
(i) $r = 0.6 \, \mu m$, $W = 0.06 \, \mu m$.  
(j) $r = 0.6 \, \mu m$, $W = 0.07 \, \mu m$.  
(k) $r = 0.6 \, \mu m$, $W = 0.08 \, \mu m$.  
(l) $r = 0.6 \, \mu m$, $W = 0.09 \, \mu m$.

Figure 13. Comparison the predicted and actual values of $D$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 4).  

(a) $r = 0.7 \, \mu m$, $W = 0.06 \, \mu m$.  
(b) $r = 0.7 \, \mu m$, $W = 0.07 \, \mu m$.  
(c) $r = 0.7 \, \mu m$, $W = 0.08 \, \mu m$.  
(d) $r = 0.7 \, \mu m$, $W = 0.09 \, \mu m$.  
(e) $r = 0.8 \, \mu m$, $W = 0.06 \, \mu m$.  
(f) $r = 0.8 \, \mu m$, $W = 0.07 \, \mu m$.  
(g) $r = 0.8 \, \mu m$, $W = 0.08 \, \mu m$.  
(h) $r = 0.8 \, \mu m$, $W = 0.09 \, \mu m$.  
(i) $r = 0.9 \, \mu m$, $W = 0.06 \, \mu m$.  
(j) $r = 0.9 \, \mu m$, $W = 0.07 \, \mu m$.  
(k) $r = 0.9 \, \mu m$, $W = 0.08 \, \mu m$.  
(l) $r = 0.9 \, \mu m$, $W = 0.09 \, \mu m$.  


Figure 14. Comparison of the predicted and actual values of $D$ in the test data set of the network model trained by different optimizers for different combinations of core radius and cantilever width (number of cantilevers: 6). (a) $r = 0.9 \, \mu m$, $W = 0.08 \, \mu m$. (b) $r = 0.9 \, \mu m$, $W = 0.09 \, \mu m$. (c) $r = 0.9 \, \mu m$, $W = 0.1 \, \mu m$. (d) $r = 0.9 \, \mu m$, $W = 0.11 \, \mu m$. (e) $r = 1 \, \mu m$, $W = 0.08 \, \mu m$. (f) $r = 1 \, \mu m$, $W = 0.09 \, \mu m$. (g) $r = 1 \, \mu m$, $W = 0.1 \, \mu m$. (h) $r = 1 \, \mu m$, $W = 0.11 \, \mu m$. (i) $r = 1.1 \, \mu m$, $W = 0.08 \, \mu m$. (j) $r = 1.1 \, \mu m$, $W = 0.09 \, \mu m$. (k) $r = 1.1 \, \mu m$, $W = 0.1 \, \mu m$. (l) $r = 1.1 \, \mu m$, $W = 0.11 \, \mu m$.

Table 7. Average of each model on evaluation metrics when predicting $D$ in the test dataset under different combinations of parameters.

| Optimizer | MSE (a.u.) | RMSE (a.u.) | MAE (a.u.) |
|-----------|------------|-------------|------------|
| SGD       | 0.01378    | 0.11740     | 0.06917    |
| Adagrad   | 0.01390    | 0.11790     | 0.06460    |
| Adam      | 0.01350    | 0.11617     | 0.06445    |

4.5. Discussion

During the experiments, the LRs were adjusted to 0.01, 0.005, and 0.001. SGD, Adam, and Adagrad were selected as the optimizers to explore the model’s performance. Combined with the above experimental results, the Adam-based model has a good prediction effect on the $A_{eff}$ and $D$. The model with SGD optimizer is the best in the prediction process of $\gamma$. The LR impacted the model’s performance when well-trained models were used to predict optical properties. Table 8 shows the combined prediction of the model for the three optical properties (evaluated in terms of MSE) under different training configurations (optimizer and LR). The overall prediction performance of the model is best when the LR is 0.001. The proposed model can predict the important optical properties of SCF well. Furthermore, it is straightforward to consider increasing the feature dimensionality of the input, increasing the number of iterations, and using more significant augmented data to improve the model’s predictive performance further. However, these strategies will also inevitably increase resource utilization and iteration time.
Table 8. MSE averages for models trained using different training configurations (optimizer and LR) predict different optical properties, the bold denotes the optimal in this prediction.

| Optimizer | LR   | $A_{eff}$ ($\times 10^{-6}$) | $\gamma$ ($\times 10^{-6}$) | $D$ ($\times 10^{-2}$) |
|-----------|------|------------------------------|-----------------------------|-------------------------|
| SGD       | 0.01 | 1.63                         | 2.87                        | 1.366                   |
|           | 0.005| 1.91                         | 1.85                        | 1.421                   |
|           | 0.001| 3.78                         | **1.2**                     | 1.349                   |
| Adagrad   | 0.01 | 2.38                         | 2.39                        | 1.455                   |
|           | 0.005| 4.92                         | 7.82                        | 1.350                   |
|           | 0.001| 8.01                         | 1.66                        | 1.367                   |
| Adam      | 0.01 | 2.17                         | 6.92                        | 1.354                   |
|           | 0.005| 1.13                         | 8.06                        | 1.347                   |
|           | 0.001| **0.51**                     | 4.37                        | **1.345**               |

5. Conclusions

In summary, this paper presents a comprehensive semi-supervised deep learning model that can efficiently solve important optical properties (effective mode area, nonlinear coefficients, and dispersion) in the inverse design of an SCF with an extremely limited a priori dataset. The model consists of a GAN, which is used to augment the limited a priori data set efficiently, and a BPNN cascade, which receives both the original and augmented data sets for calculating and optimizing the optical properties of the SCF. It is experimentally demonstrated that the model accurately predicts the optical properties. In addition, different training configurations (optimizer and learning rate) were used to train the model during the experiments to demonstrate the reliability of the model. The strategy can provide practical support for the inverse design of microstructured optical fibers that can be used for other materials and a new and reliable platform for machine learning-based physical property calculations with extremely limited a priori data sets.

Author Contributions: Conceptualization, S.C. and B.Y.; methodology, S.C.; software, G.W.; validation, G.W., S.R. and S.L.; data curation, G.W., S.R. and S.L.; writing—original draft preparation, G.W.; writing—review and editing, S.C. and B.Y.; visualization, S.R.; supervision, S.C. and B.Y.; funding acquisition, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (61762033, 61865005, 62175054); Natural Science Foundation of Hainan Province (2019CXTD400, 620RC554, 617079); Major Science and Technology Program of Haikou City (2021-002); National Key Technology Support Program (2015BAH55F04, 2015BAH55F01); Scientific Research Starting Foundation of Hainan University (KYQD(ZR)1882).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References
1. Ebendorff-Heidepriem, H.; Warren-Smith, S.C.; Monro, T.M. Suspended nanowires: Fabrication, design and characterization of fibers with nanoscale cores. Opt. Express 2009, 17, 2646–2657. [CrossRef] [PubMed]
2. Afshar, S.; Zhang, W.Q.; Ebendorff-Heidepriem, H.; Monro, T.M. Small core optical waveguides are more nonlinear than expected: Experimental confirmation. Opt. Lett. 2009, 34, 3577–3579. [CrossRef]
3. Dianov, E.M.; Prokhorov, A.M. Medium-power CW Raman fiber lasers. IEEE J. Sel. Top. Quantum Electron. 2000, 6, 1022–1028. [CrossRef]
4. Smith, S.; Zarinetchi, F.; Ezekiel, S. Narrow-linewidth stimulated Brillouin fiber laser and applications. Opt. Lett. 1991, 16, 393–395. [CrossRef] [PubMed]
5. Hirano, M.; Nakanishi, T.; Okuno, T.; Onishi, M. Silica-based highly nonlinear fibers and their application. IEEE J. Sel. Top. Quantum Electron. 2009, 15, 103–113. [CrossRef]
6. Murakami, H.; Sonnad, V.; Clementi, E. A three-dimensional finite element approach towards molecular scf computations. *Int. J. Quantum Chem.* **1992**, *42*, 785–817. [CrossRef]

7. Johnson, S.G.; Ioannopoulos, J.D. Block-iterative frequency-domain methods for Maxwell’s equations in a planewave basis. *Opt. Express* **2001**, *8*, 173–190. [CrossRef]

8. Shi, S.; Chen, C.; Prather, D.W. Plane-wave expansion method for calculating band structure of photonic crystal slabs with perfectly matched layers. *J. Opt. Soc. Am. A* **2004**, *21*, 1769–1775. [CrossRef] [PubMed]

9. Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365. [CrossRef]

10. Goh, G.B.; Hodas, N.O.; Vishnu, A. Deep learning for computational chemistry. *J. Comput. Chem.* **2017**, *38*, 1291–1307. [CrossRef] [PubMed]

11. Zahavy, T.; Dikopoltsev, A.; Moss, D.; Haham, G.I.; Cohen, O.; Mannor, S.; Segev, M. Deep learning reconstruction of ultrashort pulses. *Optica* **2018**, *5*, 666–673. [CrossRef]

12. Baldi, P.; Sadowski, P.; Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nat. Commun.* **2014**, *5*, 4308. [CrossRef]

13. Carrasquilla, J.; Melko, R.G. Machine learning phases of matter. *Nat. Phys.* **2017**, *13*, 431–434. [CrossRef]

14. Chen, S.; Yao, F.; Ren, S.; Wang, G.; Huang, M. Cost-effective improvement of the performance of AWG-based FBG wavelength interrogation via a cascaded neural network. *Opt. Express* **2022**, *30*, 7647–7663. [CrossRef]

15. Chen, S.; Yao, F.; Ren, S.; Yang, J.; Yang, Q.; Yuan, S.; Wang, G.; Huang, M. Fabry-Perot interferometric sensor demodulation system utilizing multi-peak wavelength tracking and neural network algorithm. *Opt. Express* **2022**, *30*, 24461–24480. [CrossRef]

16. Sinha, A.; Lee, J.; Li, S.; Barbastathis, G. Lensless computational imaging through deep learning. *Optica* **2017**, *4*, 1117–1125. [CrossRef]

17. Marvdashti, T.; Duan, L.; Aasi, S.Z.; Tang, J.Y.; Bowden, A.K.E. Classification of basal cell carcinoma in human skin using machine learning and quantitative features captured by polarization sensitive optical coherence tomography. *Biomed. Opt. Express* **2016**, *7*, 3721–3733. [CrossRef]

18. Jiang, J.; Sell, D.; Hoyer, S.; Hickey, J.; Yang, J.; Fan, J.A. Free-form diffractive metagrating design based on generative adversarial networks. *ACS Nano* **2019**, *13*, 8872–8878. [CrossRef] [PubMed]

19. Liu, Z.; Zhu, D.; Lee, K.T.; Kim, A.S.; Raju, L.; Cai, W. Compounding meta-atoms into metamolecules with hybrid artificial intelligence techniques. *Adv. Mater.* **2020**, *32*, 1904790. [CrossRef] [PubMed]

20. Ma, W.; Cheng, F.; Liu, Y. Deep-learning-enabled on-demand design of chiral metamaterials. *ACS Nano* **2018**, *12*, 6326–6334. [CrossRef]

21. Melati, D.; Grinberg, Y.; Kamandar Dezfooli, M.; Janz, S.; Cheben, P.; Schmid, J.H.; Sánchez-Postigo, A.; Xu, D.X. Mapping the global design space of nanophotonic components using machine learning pattern recognition. *Nat. Commun.* **2019**, *10*, 4775. [CrossRef] [PubMed]

22. So, S.; Mun, J.; Roh, J. Simultaneous inverse design of materials and structures via deep learning: Demonstration of dipole resonance engineering using core–shell nanoparticles. *ACS Appl. Mater. Interfaces* **2019**, *11*, 24264–24268. [CrossRef] [PubMed]

23. Sajedian, I.; Kim, J.; Roh, J. Finding the optical properties of plasmonic structures by image processing using a combination of convolutional neural networks and recurrent neural networks. *Microsyst. Nanoeng.* **2019**, *5*, 27. [CrossRef] [PubMed]

24. Liu, Z.; Zhu, D.; Rodrigues, S.P.; Lee, K.T.; Cai, W. Generative model for the inverse design of metasurfaces. *Nano Lett.* **2018**, *18*, 6570–6576. [CrossRef]

25. da Silva Ferreira, A.; Malheiro-Silveira, G.N.; Hernández-Figueroa, H.E. Computing optical properties of photonic crystals by using multilayer perceptron and extreme learning machine. *J. Light. Technol.* **2018**, *36*, 4066–4073. [CrossRef]

26. Chugh, S.; Gulistan, A.; Ghosh, S.; Rahman, B. Machine learning approach for computing optical properties of a photonic crystal fiber. *Opt. Express* **2019**, *27*, 36414–36425. [CrossRef] [PubMed]

27. Yuan, S.; Chen, S.; Yang, J.; Yang, Q.; Ren, S.; Wang, G.; Yu, B. Efficient calculation of optical properties of suspended-core fiber via a machine learning algorithm. *Opt. Lett.* **2022**, *46*, 5714–5721. [CrossRef]

28. Mi, N.; Wu, B.; Jiang, L.; Sun, L.; Zhao, Z.; Wang, X.; Zhang, P.; Pan, Z.; Liu, Z.; Dai, S.; et al. Structure design and numerical evaluation of highly nonlinear suspended-core chalcogenide fibers. *J. Non-Cryst. Solids* **2017**, *464*, 44–50. [CrossRef]

29. Mági, E.C.; Fu, L.; Nguyen, H.C.; Lamont, M.; Yeom, D.; Eggleton, B. Enhanced Kerr nonlinearity in sub-wavelength diameter As2Se3 chalcogenide fiber tapers. *Opt. Express* **2017**, *25*, 10324–10329. [CrossRef]

30. Yin, X.; Goudriaan, J.; Lantinga, E.A.; Vos, J.; Spiertz, H.J. A flexible sigmoid function of determine growth. *Ann. Bot.* **2003**, *91*, 361–371. [CrossRef] [PubMed]