

PAPER • OPEN ACCESS

Investigation of inverse design of multilayer thin-films with conditional invertible neural networks

To cite this article: Alexander Luce *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 015014

View the [article online](#) for updates and enhancements.

You may also like

- [Autoencoder-extended Conditional Invertible Neural Networks for Unfolding Signal Traces](#)
M Erdmann, K Hafner, J Schulte et al.
- [Accelerating physics-informed neural network based 1D arc simulation by meta learning](#)
Linlin Zhong, Bingyu Wu and Yifan Wang
- [The interface density dependence of the electrical properties of \$0.9\text{Pb}\(\text{Sc}_{0.5}\text{Ta}_{0.5}\)\text{O}_3-0.1\text{PbTiO}_3/0.55\text{Pb}\(\text{Sc}_{0.5}\text{Ta}_{0.5}\)\text{O}_3-0.45\text{PbTiO}_3\$ multilayer thin films](#)
Xue-Dong Li, , Hong Liu et al.



PAPER


Investigation of inverse design of multilayer thin-films with conditional invertible neural networks

OPEN ACCESS

RECEIVED
25 October 2022REVISED
19 December 2022ACCEPTED FOR PUBLICATION
9 January 2023PUBLISHED
7 February 2023

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Alexander Luce^{1,2,*} , Ali Mahdavi² , Heribert Wanker^{1,2,3}  and Florian Marquardt^{1,4} ¹ University Erlangen-Nürnberg, Erlangen, Germany² ams-OSRAM, Regensburg, Germany³ University Regensburg, Regensburg, Germany⁴ Max Planck Institute for the Science of Light, Erlangen, Germany

* Author to whom any correspondence should be addressed.

E-mail: alexander.luce@ams-osram.com**Keywords:** conditional invertible neural networks, multilayer thin-films, inverse design, latent space, global optimization**Abstract**

In this work, we apply conditional invertible neural networks (cINN) to inversely design multilayer thin-films given an optical target in order to overcome limitations of state-of-the-art optimization approaches. Usually, state-of-the-art algorithms depend on a set of carefully chosen initial thin-film parameters or employ neural networks which must be retrained for every new application. We aim to overcome those limitations by training the cINN to learn the loss landscape of all thin-film configurations within a training dataset. We show that cINNs can generate a stochastic ensemble of proposals for thin-film configurations that are reasonably close to the desired target depending only on random variables. By refining the proposed configurations further by a local optimization, we show that the generated thin-films reach the target with significantly greater precision than comparable state-of-the-art approaches. Furthermore, we tested the generative capabilities on samples which are outside of the training data distribution and found that the cINN was able to predict thin-films for out-of-distribution targets, too. The results suggest that in order to improve the generative design of thin-films, it is instructive to use established and new machine learning methods in conjunction in order to obtain the most favorable results.

1. Introduction

In optics, being able to develop devices that manipulate light in a desired way is a key aspect for all applications within the field such as illumination [1] or integrated photonics [2]. Recent developments in machine learning, deep learning and inverse design offer new possibilities to engineer such optical and photonic devices [3–8]. Nanophotonics in particular benefits from the recent advancements in optimization and design algorithms [9, 10]. For example the development of meta optics or the design of scattering nano particles was greatly improved by employing gradient-based inverse design and deep learning [11–13]. Multilayer thin-films are another instance of nanophotonic devices which are employed to fulfill a variety of different functionalities. They are particularly interesting for industrial applications because the standard fabrication processes are so mature that the predicted optical properties of thin-films generally match the properties of the fabricated thin-films very well. Application examples are vertical-cavity surface-emitting lasers [14, 15], anti-reflection coatings [16] and wavelength demultiplexers [17]. Recently, they were successfully employed to enhance the directionality of a white LED while maintaining the desired color temperature [18]. Designing multilayer thin-films [19–21] has long been a task in the nanophotonics community, and sophisticated techniques have been developed to synthesize thin-films that exhibit the desired optical properties, as the problem becomes computationally intractable when using simple combinatorics, since the number of possible thin-film designs scales exponentially with the number of layers. To design multilayer thin-films, many open-source or commercial software solutions are available that employ different approaches to optimize thin-films [22–26]. Methods such as the Fourier method [27, 28] or

the needle method [21, 28–30] compute the position inside the thin-film where the introduction of a new layer is most beneficial. Then the software will continue with a refinement process, often based on a gradient-based optimization such as the Levenberg–Marquardt algorithm [28, 31], until it reaches a local minimum where it will then introduce another layer. Although the software will usually converge to a satisfying solution with respect to the given target, the presented solutions often use excessive amounts of layers and the optimization is still limited by the selected parameters in the beginning of the optimization. The problem of converging to local optima was tackled in the past by the development of numerous global optimization techniques which have been introduced and tested in the field of thin-film optimization [32–36]. Recently, the innovations of machine learning attracted much interest in the thin-film community and resulted in interesting new ways to create thin films [5, 37]. Particularly, deep reinforcement learning or Q-learning showed promising results in designing new and efficient multilayer thin-films while punishing complicated designs, which employ many layers [38, 39] and require targets that are difficult to achieve with conventional optimization.

In this work we employ so called conditional invertible neural networks (cINNs) [40] to directly infer the loss landscape of all thin-film configurations with a fixed number of layers and material choice. The cINN learns to map the thin-film configuration to a latent space, conditional on the optical properties, i.e. the reflectivity of a thin-film. During inference, due to the invertibility of the architecture, the cINN maps selected points from the latent space to their most likely thin-film configurations, conditional on a chosen target. This results in requiring only a single application of the cINN to obtain the most likely thin-film configuration given an optical target. Additionally, the log-likelihood training makes the occurrence of mode-collapse [41] almost impossible. For thin-films, many different configurations lead to similar optical properties. For conventional optimization, this leads to the convergence of the optimization to possibly unfavorable local minima. A cINN circumvents this due to the properties of the latent space—by varying the points in the latent space, a perfectly trained cINN is able to predict any possible thin-film configuration that satisfies the desired optical properties. In this work, we investigated how good the generative capabilities of a cINN are for finding suitable thin-film configurations in a real-world application. We present an optimization algorithm, which is suitable to improve the thin-film predictions of the cINN. Then, we compared the optimization results of the presented algorithm to state-of-the-art software. Finally, we discuss the limitations of the approach and give a guideline when the application of a cINN is advantageous.

2. Normalizing flows and cINNs

Invertible neural networks are closely related to normalizing flows, which were first popularized by Dinh *et al* [42]. A normalizing flow is an architecture that connects two probability distributions by a series of invertible transformations. The idea is to map a complex probability distribution to a known and simple distribution such as a Gaussian distribution. This can be used both for density estimation and therefore also for sampling. A sample can be easily drawn with a Gaussian distribution by selecting $z \sim \mathcal{N}(\mu, \sigma)$. The generated points z are then mapped to the complex distribution via the normalizing flow. The architecture of a normalizing flow is constructed from the following. Assume two probability distributions, π which is known and for which $z \sim \pi(z)$ holds and the complex, unknown distribution p . The mapping between both is given by the change-of-variables formula

$$p(x) = \pi(z) \left| \det \left[\frac{\partial z}{\partial x} \right] \right|. \quad (1)$$

Consider a transformation f which maps $f(x) = z$. Then the change-of-variables formula can be written as

$$p(x) = \pi(z) \left| \det \left[\frac{\partial f(x)}{\partial x} \right] \right|. \quad (2)$$

The transformation f can be given by a series of invertible transformations $f = f_K \circ f_{K-1} \dots \circ f_0$ with $x = z_K = f(z_0) = (f_K \circ \dots \circ f_0)(z_0)$. Then, the probability density at any intermediate point is given by $p_i(x_i) = z_i = f_i(z_{i-1})$. By rewriting the change-of-variables formula and taking the logarithm one obtains

$$\log(p(x)) = \log \left(\pi(z_0) \prod_{i=1}^K \left| \det \left[\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}} \right] \right|^{-1} \right) = \log(\pi(z_0)) - \sum_{i=1}^K \log \left| \det \left[\frac{\partial f_i(z_{i-1})}{\partial z_{i-1}} \right] \right|. \quad (3)$$

To be practical, a key component of any transformation of a normalizing flow is that the Jacobian determinant of the individual transformations must be easy to compute. A suitable invertible transformation, which is sufficiently expressive is the so called real-valued non-volume preserving

transformation (RNVP block) [43]. The input z is split into two separate vectors u_1 and u_2 and is processed with the help of two transformation functions, s_2 and t_2 , while the other input u_2 is kept fixed

$$f_i(u_1 \otimes u_2) = \{u_1 \odot \exp(s_2(u_2)) \oplus t_2(u_2) \otimes u_2\} = \{v_1 \otimes u_2\}. \quad (4)$$

\odot, \oplus, \ominus and \otimes denote element wise computation while \otimes denotes a concatenation. The inverse of this transformation is given by

$$f_i^{-1}(v_1 \otimes u_2) = \{(v_1 \ominus t_2(u_2)) \odot \exp(s_2(u_2)) \otimes u_2\}. \quad (5)$$

To invert the entire transformation, no inversion of the transformations s_2 and t_2 is required. Therefore, a neural network can be used as a transformation to make the normalizing flow expressive. The Jacobian of the transformation is given by an upper triangular matrix. Therefore, the Jacobian determinant is easy to compute since only the diagonal elements contribute.

$$\det \left[\frac{\partial f_i}{\partial z} \right] = \det \begin{bmatrix} \mathbb{1}^d & 0 \\ \frac{\partial v_1}{\partial u_2} & \text{diag}(\exp(s_2(u_2) + t_2(u_2))) \end{bmatrix}. \quad (6)$$

Since only one part of the input is transformed by a neural network, the RNVP block is repeated and applied to the yet untransformed part of the input u_2 with two additional transformation functions s_1 and t_1 . Other transformations, which can be advantageous depending on the specific application are the GLOW transformation [44], which utilizes invertible 1x1 convolutions or the masked autoregressive flow with a generalized implementation of the RNVP block [45].

Utilizing normalizing flows for the task of a generative neural network for the proposition of thin-films requires some modification of the normalizing flow. Ardizzone *et al* [40] propose a so called cINN which extends the change-of-variables formula to conditional probability densities with condition c

$$p(x|c) = \pi(z) \left| \det \left[\frac{\partial f(x; c)}{\partial x} \right] \right|. \quad (7)$$

By assuming a Gaussian probability distribution for π and taking the logarithm, the conditional maximum likelihood loss function is derived by Ardizzone *et al* for training of a cINN as

$$\mathcal{L}_{\text{cML}} = E \left[\frac{\|f(x; c)\|_2^2}{2} - \log \left| \det \left[\frac{\partial f(x; c)}{\partial x} \right] \right| \right]. \quad (8)$$

The condition c can be the result of another neural network $\varphi(c) = s$, which extracts features from the given condition c . The features are then passed to the RNVP transformations by appending the features to the input vectors u_1 and u_2 . By jointly training the cINN via the conditional maximum likelihood loss, Ardizzone *et al* showed that the feature extraction network learns to extract useful information for the invertible transformation. The condition c can be thought of as the target of the cINN. During inference, the cINN transforms Gaussian samples z to the learned distribution, conditional on the target c . The inferred result returned by the cINN is then given by $x = f^{-1}(z, \varphi(c))$ while f denotes the invertible neural network. Ardizzone *et al* also provide a Python implementation FrEIA⁵ for conditional and regular invertible neural network based on the Pytorch⁶ Deep Learning library.

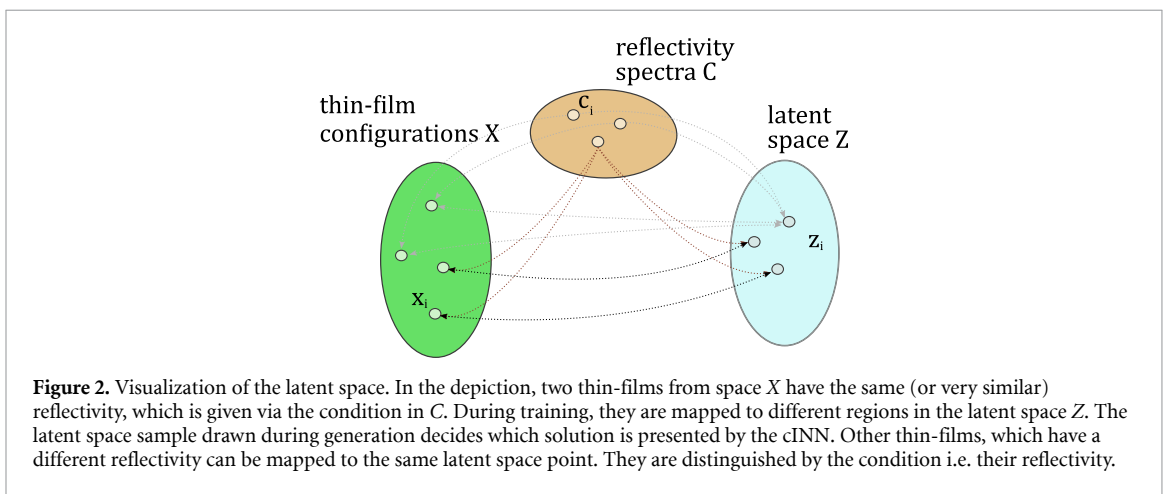
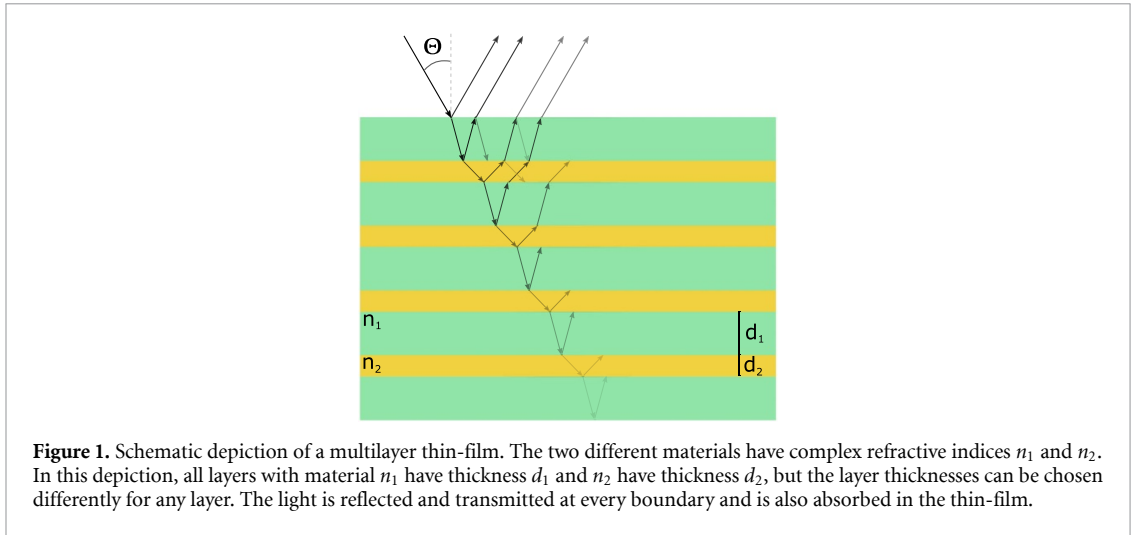
3. Application of cINNs to generating multilayer thin-films

Multilayer thin-films are an optical component that consists of a sequence of planar layers with different materials stacked on top of each other with a varying layer thickness. Light, which irradiates the thin-film, can be transmitted and reflected at the layer interfaces and absorbed within the layer, as depicted in figure 1. This interaction of reflection, transmission and absorption is different for different wavelengths of light λ and angles of incidence Θ . A fast and convenient way to compute the optical response of a thin-film is to employ the transfer matrix method (TMM) [46]. In a previous work, we developed the Python package TMM-Fast⁷ [47] which allows to compute the optical response of a thin-film. The package also implements convenience functionality for thin-film-dataset generation, which is especially important for machine

⁵ <https://github.com/VLL-HD/FrEIA#papers>.

⁶ <https://pytorch.org/>.

⁷ https://github.com/MLResearchAtOSRAM/tmm_fast.



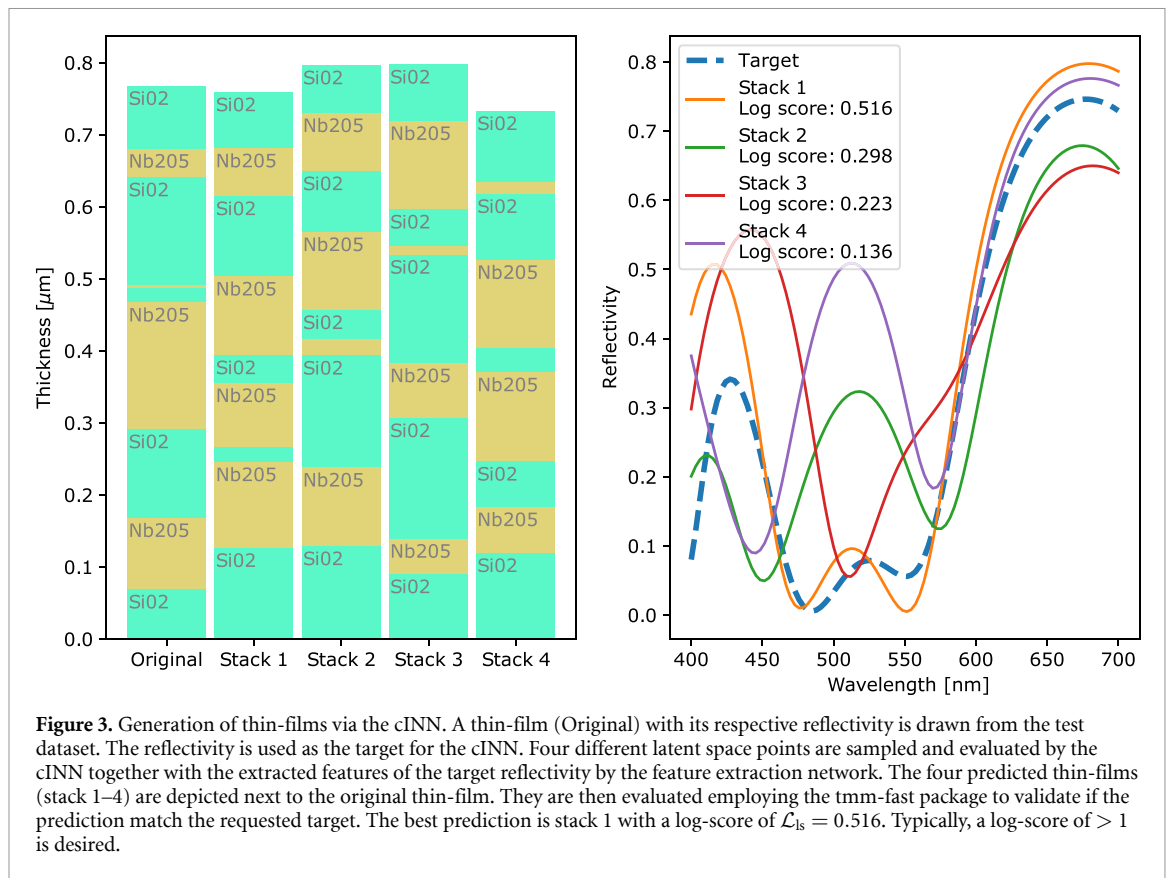
learning. By changing the layer thicknesses, the behavior of the thin-film can be modified, which are the parameters p that are up to optimization.

Finding a thin-film design, which fulfills the optical criteria while employing only a very limited number of layers is a challenging task since the loss landscape is highly non-convex and high-dimensional. The loss landscape is given by a mapping of the optical characteristics \mathcal{M} of the thin-films by a loss function $\mathcal{L}(\mathcal{M}(\Theta, \lambda), \mathcal{M}_{\text{target}}(\Theta, \lambda)) = E$ with $E \in \mathbb{R}$. The results of any local optimization, which is performed on the thin-film is highly dependent on the initial choice of the thin-film parameters as in any non-convex optimization. The idea of this paper is to interpret the loss landscape of the thin-films parameters as a probability distribution and train a cINN to find a mapping between the Gaussian latent space and the true loss landscape. The condition is then given by the optical characteristic of the thin-film \mathcal{M} .

An important aspect of thin-films is also the possibility of ambiguity, which is a byproduct of the non-convex loss landscape. Two distinct thin-films $p_1 \neq p_2$ can have a very similar optical characteristic $\mathcal{M}_1(p_1) \approx \mathcal{M}_2(p_2)$. Many generative models suffer under mode collapse [41], which slows the training of the neural network and often leads the neural network to ignore possible solutions or give wrong solutions altogether if ambiguity is present within the problem. The log-likelihood training of the cINN makes mode collapse virtually impossible to occur [48] and turns ambiguous solutions distinguishable via the latent space, see figure 2. Therefore, it is possible to scan the latent space systematically for favorable design configurations.

3.1. Multilayer thin-film dataset

As noted before, the TMM-Fast package is used to create a dataset for training. In the presented case, we consider nine-layer thin-films and fix the material choice to an alternating sequence of niobium pentoxide (Nb₂O₅) and silicon dioxide (SiO₂), while the light injection layer consists of gallium nitride and the ambient outcoupling layer consists of air. For the dataset, one million thin-films are created for which the layer thickness of the nine layers of Nb₂O₅ and SiO₂ are uniformly sampled from a thickness range of



[10 nm, 200 nm]. 10% of the created thin-films are separated as test dataset. As optical response, we compute the reflectivity of the thin-film for normal incidence at 100 equidistant points in a range of [400 nm, 700 nm]. The computations were performed with our self-developed open-source transfer matrix solver [47] and took about 40min on a Intel(R) Xeon(R) E5-2697v2 CPU. To test the generative capabilities, 600 so called target reflectivities are generated, which implement a selection of potentially interesting targets for which *a priori* no ground-truth thin-film with such an optical response is known. The dataset is normalized to mean $\mu = 0$ and standard deviation $\sigma = 1$ before it is processed by the neural networks.

3.2. Layout of the cINN

As explained in section 2, a cINN consists of two neural networks, the invertible part and the feature extraction network, which processes the condition. The used feature extraction network is based on a ResNet architecture adapted to 1D convolutions. It features two residual shortcuts with 20-channel 1D convolutions, batch normalization, ReLU activation and a final dense layer with ReLU activation function. The feature extraction network receives the appropriate reflectivity spectrum (training or target), depending on whether the network is in training or inference mode. The output of the feature network is passed to the invertible neural network. The invertible part consists of eight ‘all_in_one_blocks’ from the FrEIA package. These blocks implement the RNVP transformation of the normalizing flow with additional functionality, such as active normalization [44]. Here, a dense neural network with three layers, 512 neurons each and ReLU activations are used as secondary transformations for every RNVP block. Additionally, batch normalization is performed in the secondary transformations.

The training consists of 40 epochs of training, with an initial learning rate of 0.001 and a step decay of 0.08 of the learning rate after 20 and 35 epochs. We used Adam as optimizer with standard hyper-parameters. Training was performed on a GPU server with a Nvidia Tesla P100 graphics card and took less than one hour.

3.3. Evaluation of the generative capabilities of the cINN

To test the generative capabilities of the trained cINN qualitatively, the reflectivity of a test dataset sample is taken and four random samples from the latent space are drawn. First, the reflectivity is processed by the feature extraction network. Then, the four drawn latent space samples and the extracted features are passed to the invertible network and processed by it. The generated thin-films (stack 1–4) are shown in figure 3 together with the thin-film that generated the target during the dataset generation (labeled Original). To validate the performance of the generated thin-films, the TMM is used to compute the reflectivity. The

Algorithm 1. Pseudocode for the optimization of a thin-film with initialization by the cINN. The user needs to specify the *loss function* $\mathcal{L} : \vec{x} \mapsto l$, where \vec{x} denotes the optical properties of interest of the thin-film. \mathcal{L} measures the performance of the thin-film such that the requirements are fulfilled if $l = 0$.

```

1: Sample latent space values via  $z = \mathcal{N}(0, \sigma)$ 
2: Choose target  $t$ 
3: features = conditional_network( $t$ )
4: proposals = invertible_network( $z$ , features)
5: parameters = argmax( $\mathcal{L}_{ls}$ (proposals))
6: while (iter < maxiter) and (error > tolerance) do
7:   parameters, error = downhill_simplex_step(parameters)
8: end while
9: return parameters, error

```

performance metric displayed is the *log-score* (\mathcal{L}_{ls}), which is the transformed mean squared error between the target and the validated reflectivity. The log-score is given by $\mathcal{L}_{ls} = -0.43 \log \left(\frac{1}{N} \sqrt{\sum_i (x_i - x_{\text{target},i})^2} \right) - 0.52$. By performing a logarithmic transformation of the mean squared error the performance metric becomes more human-readable to quickly judge whether a thin-film is sufficiently approximating the target. The prefactors of the log-score are chosen arbitrarily in such a way that $\mathcal{L}_{ls} = 1$ is a threshold below which a generated thin-film is not acceptable although log-scores > 1 are still desired. The validated reflectivities of the propositions of the cINN capture the general trend of the target, they fail to match the target sufficiently and have a log-score of < 1 . Repeating the experiment with different targets yields similar results.

3.4. Influence of the standard deviation on cINN predictions

The latent space samples for the inverse application of the cINN are drawn from a Gaussian distribution. During training, the distribution of thin-films is mapped to a normal distribution (zero mean and standard deviation of one). Since the log-likelihood training preserves the probability density of the mapping between the thin-film distribution and the latent space, the log-probability of a latent space sample dictates the likelihood of the predicted thin-film to be the appropriate thin-film for the given condition. It is possible to think of the standard deviation as a ‘creativity parameter’ for the cINN. Sampling with $\sigma \approx 0$ results in only the most likely thin-films while $\sigma > 1$ results in explorative behavior of the cINN. Notably, the cINN will also predict nonphysical thin-films for latent space samples with low likelihood, e.g. layers with negative thickness.

3.5. Local optimization by Nelder-Mead downhill-simplex algorithm

To improve upon the proposals from the cINN, a subsequent local optimization can be performed by using the suggested thin-film as initial starting values. The optimization is performed by using the downhill-simplex algorithm [49] via the implementation of SciPy⁸. First, latent space values are drawn from a Gaussian distribution with zero mean and a standard deviation of one. The samples are passed to the invertible network together with the features extracted from the target. Then the thin-film with the best approximation is optimized via the downhill-simplex algorithm. The entire algorithm is also depicted as pseudo-code in algorithm 1.

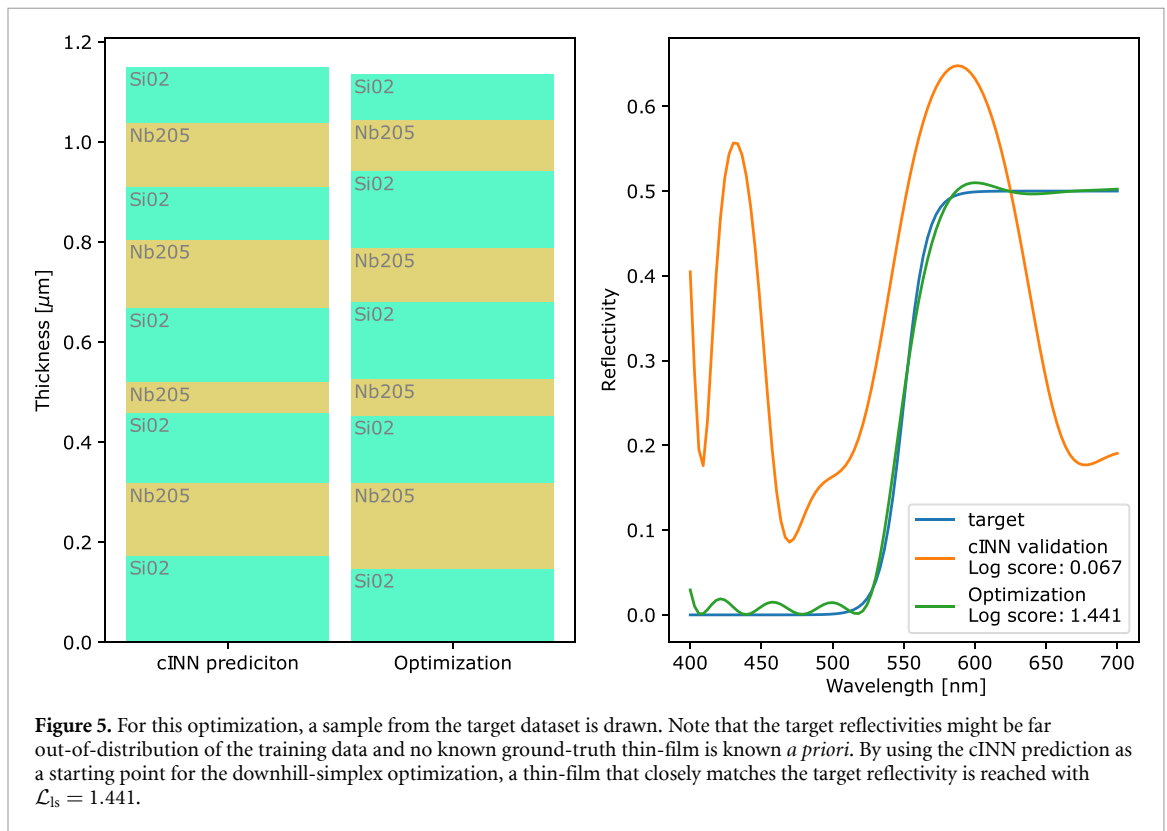
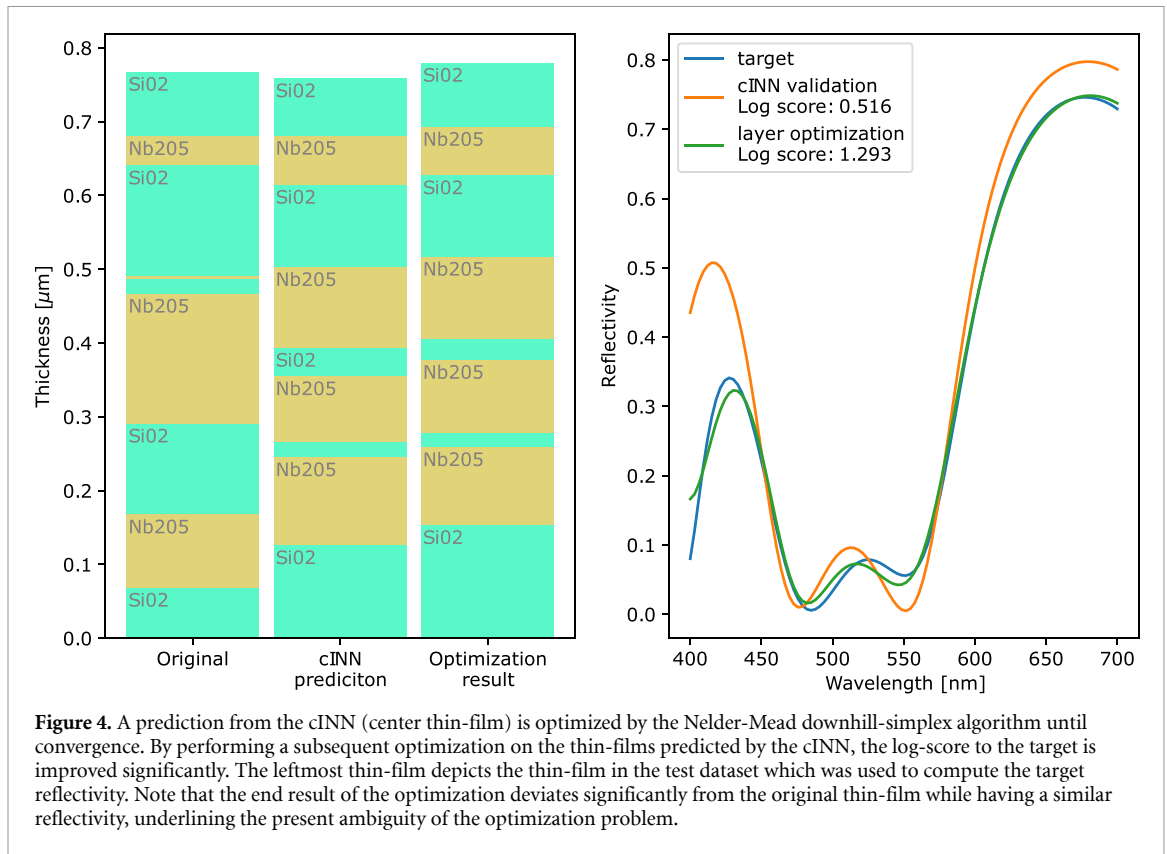
Applying the sketched out algorithm on the target shown in figure 3 results in selecting *Stack 1* from the four proposed thin-films for further refinement. The optimization yields thin-films with a much better approximation of the target with $\mathcal{L}_{ls} = 1.293$. The results of one such optimization is shown in figure 4. Repeating the algorithm with different targets yields similar results.

We can explore if the cINN can also propose thin-films for previously unseen targets by selecting a reflectivity from the target dataset. Algorithm 1 is employed and the results are shown in figure 5. An interesting observation is that even though the log-score of the reflectivity of the predicted thin-film is very poor ($\mathcal{L}_{ls} = 0.067$), the thin-film parameterizations of the prediction and after the optimization are very similar. This suggests that the cINN approximated the global loss landscape sufficiently to identify promising regions but not sufficiently to single out local minima.

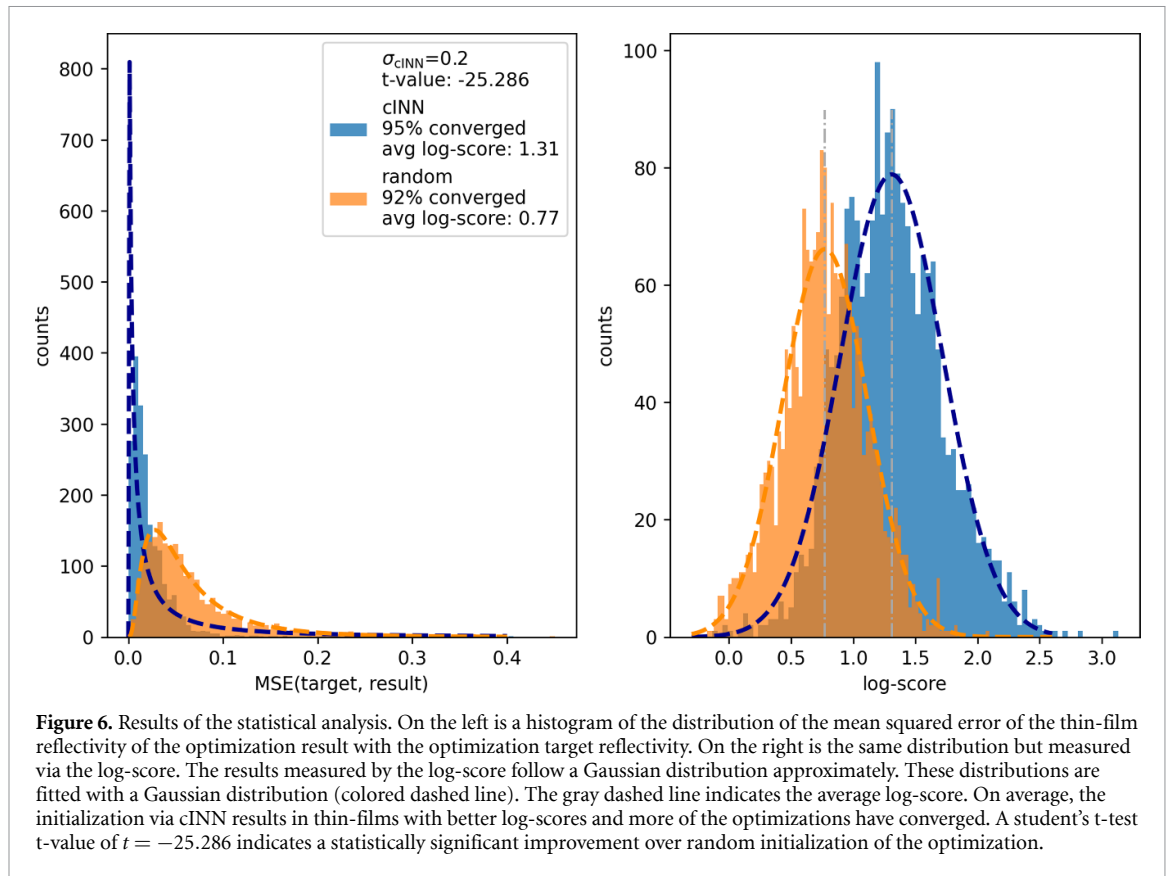
3.6. Statistical investigation of the optimization initialization via cINN

Since the cINN cannot achieve the desired accuracy on its own and requires a subsequent optimization, it is important to investigate if and how much of an advantage is gained by using a cINN as initialization of the

⁸ <https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html>.



optimization. To test this, 2000 reflectivities are sampled from the test dataset. These reflectivities are used as targets for two optimizations, once with initialization via the cINN and once by sampling thin-films from the same distribution with which the training dataset was generated, see section 3.1. We chose to only evaluate a sampled reflectivity once in order to prevent the selection bias of the cINN. The results of the optimization with a sampling standard deviation of $\sigma = 0.2$ for the latent space is shown in figure 6. On average, by



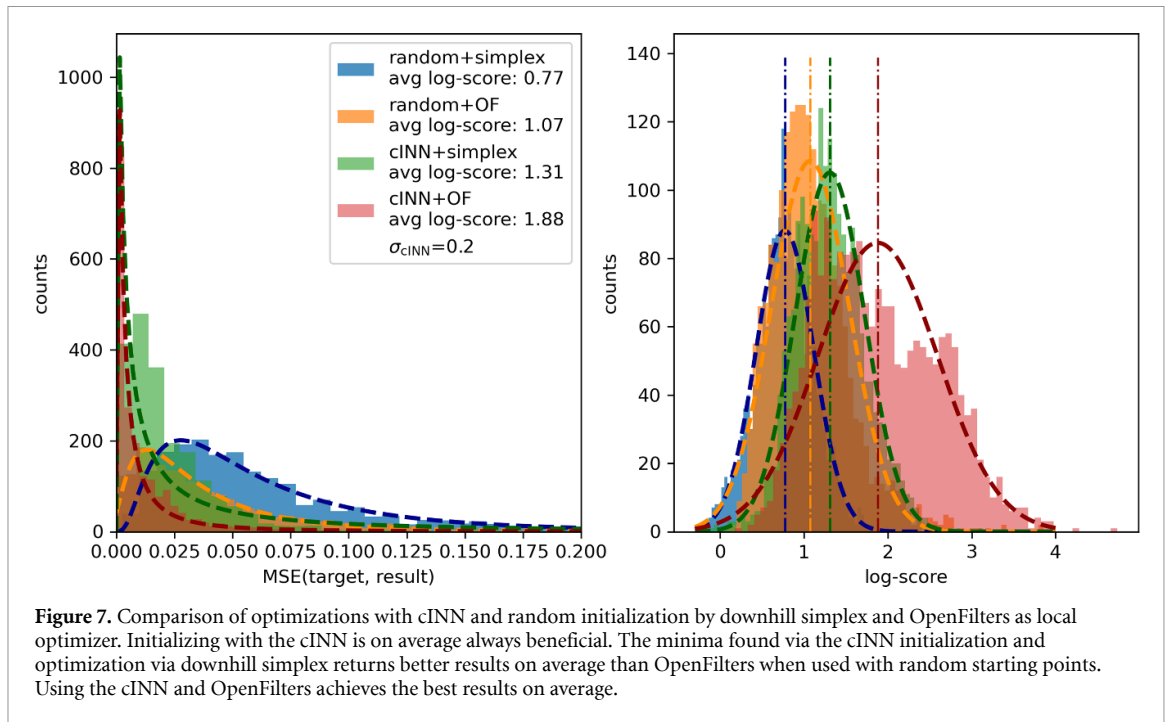
initialization via the cINN the optimization converges more often, with less iterations and to better local minima than an initialization with random initial values. The same experiment was repeated with standard deviations $\sigma_i = [0.05, 0.2, 0.5, 1.0, 1.5, 2.0]$. For all of the evaluated standard deviations, the cINN initialization was superior than random initialization. The results with $\sigma = 0.2$ showed the biggest improvement over random initialization while $\sigma = 2.0$ showed the least improvement with an average log-score of $\mathcal{L}_{ls} = 1.00$. Additionally, Student's t-test is performed which yields a t-value of -25.286 , which suggests that the results for the presented analysis are significant beyond a reasonable doubt.

Next, the same statistical investigation was performed on all 600 target dataset reflectivities. Here, initialization of the optimization via cINN resulted in better thin-films as well. However, since many of the target reflectivities lie out-of-distribution of the training dataset and might not even be achievable given the particular material choice, the average log-score is slightly closer to the random initialization compared to the investigation of the test dataset reflectivities shown in figure 6. Again, setting the standard deviation to $\sigma = 0.2$ yields the best results while any initialization with the cINN is superior than using random initialization. The results are shown in figure 8 in the [appendix](#).

Finally, a single reflectivity from the test dataset was chosen randomly and 1000 initial thin-films are sampled via the cINN and from the same distribution with which the training dataset was generated. Then, the initial thin-films were optimized and the found local minima are evaluated via log-score and mean squared error. The results are shown in figure 9. Note, that one would expect the optimizer to converge to the same local minimum for a low sampling standard deviation of the cINN. Due to the gradient approximation of the downhill-simplex algorithm, the optimization might not always converge to exactly the same local minimum if the initializations differ slightly.

3.7. Comparison to state-of-the-art and discussion

There are many existing software solutions which offer an automated design process to find thin-films according to some specifications. For a comparison with state-of-the-art, we chose the Software OpenFilters (OFs) [28], which designs thin-films via a two-step process. First, an initial thin-film must be given by the user. Then a refinement based on the Levenberg–Marquardt (LM) algorithm is applied to optimize the thin-film layers. If the LM algorithm has converged, the needle-point method is used and a position to



introduce a new layer is chosen [29], from which the LM algorithm continues to refine the thin-film. Here, the needle-point method is used to circumvent the problem of local minima at the expense of introducing additional layers and increasing the overall thickness of a thin-film. We chose to turn off the needle-point method in the following analysis since thin-films with more than nine layers would be created. Then we repeated the statistical analysis presented in section 3.6. The results are virtually the same as with the downhill-simplex algorithm, but with a higher baseline due to the fine-tuned refinement algorithm of OFs. Again, using the cINN with a sampling standard deviation of $\sigma = 0.2$ yielded the best results overall while any initialization via the cINN resulted in better optimization than random initialization on average. All results for the statistical analysis of 2000 test dataset reflectivities with standard deviation $\sigma = 0.2$ are shown in figure 7. Crucially, the initialization via the cINN improves the optimization via OFs as well.

The analysis shows that initialization via a cINN is always advantageous compared to randomly starting a local optimization. The cINN learns to identify promising regions of thin-film parameterizations from which a local optimization converges to a local minimum. The local minima reached from the cINN propositions are very often better than starting an optimization with random parameters. This advantage could be alleviated with expert knowledge. If an expert has extensive understanding of the problem and is able to identify a promising initialization the advantage of employing a cINN may be diminished. However, it might still be interesting to query a cINN to discover promising new parameterizations. Importantly, the cINN is able to extrapolate to out-of-distribution problems as demonstrated by the target dataset reflectivities. Generally, the advantage of employing a cINN is greatest if the target of the problem changes frequently. Since the cINN learns the entire probability distribution of the input parameters, one obtains the biggest advantage if the entire loss landscape is also of interest. If the problem consists of finding the best possible parameters for a single target, there might by other means to obtain local minima closer to the global minimum. Wankerl *et al* showed this, by employing Reinforcement learning on a similar problem [39]. This approach lacks the option to change targets quickly though. A neural network is also restricted in the amount of training data it obtains. Especially for high dimensional problems, it is impossible to cover the parameterizations sufficiently to cover the global loss landscape which is a problem for cINNs. The likelihood will be estimated based on a very small sample size compared to the dimensionality of the problem. Combining a cINN for initialization and evolutionary algorithm such as in this work from Hedge [5] could increase the efficiency of the evolutionary algorithm and therefore results in minima that are much closer to the global minimum.

Finally, we realized that the best results can be achieved if Machine Learning is used in tandem with conventional optimization techniques. This leads to much better local minima concerning the task of finding suitable thin-films with respect to some target reflectivity. It is therefore instructive to view Machine Learning as an extension of generative capabilities and to exploit the best of both worlds.

4. Conclusion

In this work, we presented the application of cINNs on the task of generating multilayer thin-films by providing a target reflectivity. By sampling points in a Gaussian latent space, and providing a target as a condition, a cINN yields a thin-film prediction according to the probability density of the sampled latent space points. Since the learned probability distribution of the thin-film configurations is only an approximation of the true distribution, a cINN does not suffice to give satisfactory estimations out-of-the-box. A subsequent optimization via the downhill-simplex algorithm or by OFs improved the proposed thin-films notably. By comparing the optimization results of randomly selected thin-film initialization with initialization by the cINN, we showed a significant improvement of the optimization accuracy on average by employing a cINN.

More generally, the promising results obtained for this specific application indicate that cINN can be a very effective tool for speeding up inverse design and optimization problems, in optics and beyond. Depending on the type of problem researchers are challenged with, training a cINN to propose starting points for a subsequent optimization might be worth the initial overhead of generating a dataset and neural network training, especially if the task involves changing the optimization target frequently. This would become particularly beneficial if the loss landscape of the problem is highly non-convex and a local optimization would converge to an unfavorable local optimum very quickly.

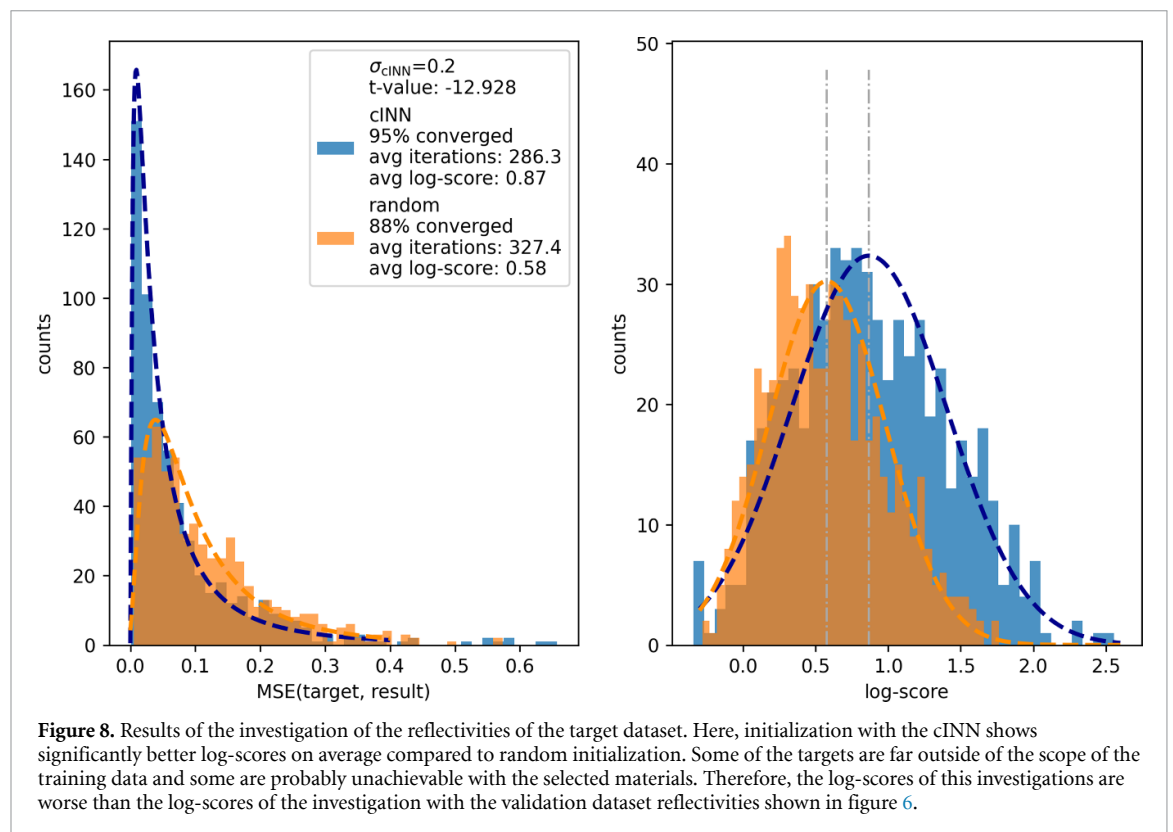
The network architecture of the cINN is available at <https://github.com/MLResearchAtOSRAM/OsCINN>

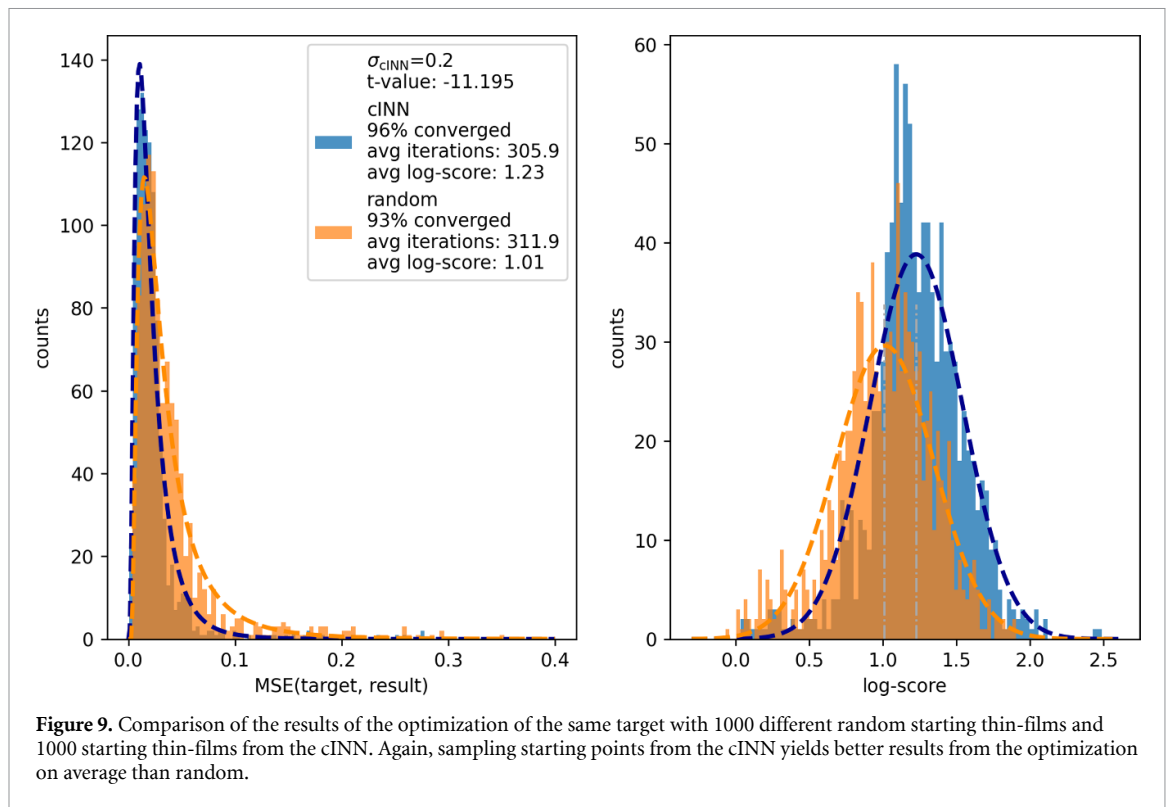
Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: [10.1364/josaa.450928](https://doi.org/10.1364/josaa.450928); <https://github.com/MLResearchAtOSRAM/OsCINN>; https://github.com/MLResearchAtOSRAM/tmm_fast.

Appendix

Additional results





ORCID iDs

Alexander Luce  <https://orcid.org/0000-0002-9659-6579>

Ali Mahdavi  <https://orcid.org/0000-0001-9548-5773>

Heribert Wankerl  <https://orcid.org/0000-0001-5634-4038>

Florian Marquardt  <https://orcid.org/0000-0003-4566-1753>

References

- [1] Taki T and Strassburg M 2019 Review—visible LEDs: more than efficient light *ECS J. Solid State Sci. Technol.* **9** 015017
- [2] Pelucchi E *et al* 2022 The potential and global outlook of integrated photonics for quantum technologies *Nat. Rev. Phys.* **4** 194–208
- [3] Hammond A M and Camacho R M 2019 Designing integrated photonic devices using artificial neural networks *Opt. Express* **27** 29620–38
- [4] Tahersima M H, Kojima K, Koike-Akino T, Jha D, Wang B, Lin C and Parsons K 2019 Deep neural network inverse design of integrated photonic power splitters *Sci. Rep.* **9** 1368
- [5] Hedge R S 2019 Accelerating optics design optimizations with deep learning *Opt. Eng., Bellingham* **58** 065103
- [6] Molesky S, Lin Z, Piggott A Y, Jin W, Vucković J and Rodriguez A W 2018 Inverse design in nanophotonics *Nat. Photon.* **12** 659–70
- [7] Ma W, Liu Z, Kudyshev Z A, Boltasseva A, Cai W and Liu Y 2021 Deep learning for the design of photonic structures *Nat. Photon.* **15** 77–90
- [8] Peano V, Sapper F and Marquardt F 2021 Rapid exploration of topological band structures using deep learning *Phys. Rev. X* **11** 021052
- [9] Hughes T W, Minkov M, Williamson I A D and Fan S 2018 Adjoint method and inverse design for nonlinear nanophotonic devices *ACS Photonics* **5** 4781–7
- [10] Su L, Vercauteren D, Skarda J, Sapra N V, Petykiewicz J A and Vuckovic J 2019 Nanophotonic inverse design with spins: software architecture and practical considerations
- [11] Li Z, Pestourie Rel, Park J-S, Huang Y-W, Johnson S G and Capasso F 2022 Inverse design enables large-scale high-performance meta-optics reshaping virtual reality *Nat. Commun.* **13** 2409
- [12] An S *et al* 2020 Deep learning modeling approach for metasurfaces with high degrees of freedom *Opt. Express* **28** 31932–42
- [13] Peurifoy J, Shen Y, Jing Li, Yang Y, Cano-Renteria F, DeLacy B G, Joannopoulos J D, Tegmark M and Soljacic M 2018 Nanophotonic particle simulation and inverse design using artificial neural networks *Sci. Adv.* **4** eaar4206
- [14] Iga K 2018 Forty years of vertical-cavity surface-emitting laser: Invention and innovation *Jpn. J. Appl. Phys.* **57** 08A01
- [15] Gebski M, Lott J A and Czyzanski T 2019 Electrically injected vcsel with a composite dbr and mhcg reflector *Opt. Express* **27** 7139–46
- [16] Kumar Raut H, Anand Ganesh V, Sreekumaran Nair A and Ramakrishna S 2011 Anti-reflective coatings: a critical, in-depth review *Energy Environ. Sci.* **4** 3779–804
- [17] Gerken M and Miller D A B 2003 Multilayer thin-film structures with high spatial dispersion *Appl. Opt.* **42** 1330–45
- [18] Wankerl H, Wiesmann C, Kreiner L, Butendeich R, Luce A, Sobczyk S, Stern M L and Lang E W 2022 Directional emission of white light via selective amplification of photon recycling and bayesian optimization of multi-layer thin films *Sci. Rep.* **12** 5226
- [19] Tikhonravov A V 1993 Some theoretical aspects of thin-film optics and their applications *Appl. Opt.* **32** 5417–26

- [20] Becker H, Tonova D, Sundermann M, Ehlers H, Günster S and Ristau D 2014 Design and realization of advanced multi-index systems *Appl. Opt.* **53** A88–A95
- [21] Tikhonravov A V and Trubetskov M K 2012 Modern design tools and a new paradigm in optical coating design *Appl. Opt.* **51** 7319–32
- [22] OptiLayer GmbH OptiLayer (available at: www.optilayer.com)
- [23] Thin Film Center Inc. Thin Film Center (available at: www.thinfilmcenter.com)
- [24] RP Photonics AG RP Coating (available at: www.rp-photonics.com/coating.html)
- [25] Software Spectra, Inc. TFCalc (available at: www.spectra.com)
- [26] Scientific Computing International Film Wizard (available at: <https://sci-soft.com/product/film-wizard/>)
- [27] Dobrowolski J A and Lowe D 1978 Optical thin film synthesis program based on the use of fourier transforms *Appl. Opt.* **17** 3039–50
- [28] Larouche S and Martinu L 2008 Openfilters: open-source software for the design, optimization and synthesis of optical filters *Appl. Opt.* **47** C219–30
- [29] Sullivan B T and Dobrowolski J A 1996 Implementation of a numerical needle method for thin-film design *Appl. Opt.* **35** 5484–92
- [30] Tikhonravov A V, Trubetskov M K and DeBell G W 2007 Optical coating design approaches based on the needle optimization technique *Appl. Opt.* **46** 704–10
- [31] Marquardt D W 1963 An algorithm for least-squares estimation of nonlinear parameters *SIAM J. Appl. Math.* **11** 431–41
- [32] Chang C P, Lee Y H and Wu S Y 1990 Optimization of a thin-film multilayer design by use of the generalized simulated-annealing method *Opt. Lett.* **15** 595
- [33] Paszkowicz W 2013 Genetic algorithms, a nature-inspired tool: a survey of applications in materials science and related fields: Part II *Mater. Manuf. Process.* **28** 708–25
- [34] Yang C, Hong L, Shen W, Zhang Y, Liu X and Zhen H 2013 Design of reflective color filters with high angular tolerance by particle swarm optimization method *Opt. Express* **21** 9315–23
- [35] Guo X, Zhou H Y, Guo S, Luan X X, Cui W K, Ma, Y F and Shi L 2014 Design of broadband omnidirectional antireflection coatings using ant colony algorithm *Opt. Express* **22** A1137
- [36] Martin S, Rivory J and Schoenauer M 1995 Synthesis of optical multilayer systems using genetic algorithms *Appl. Opt.* **34** 2247
- [37] Roberts J and Wang E W 2018 Modeling and optimization of thin-film optical devices using a variational autoencoder *Technical Report* Stanford University
- [38] Jiang A, Osamu Y and Chen L 2020 Multilayer optical thin film design with deep Q learning *Sci. Rep.* **10** 12780
- [39] Wankel H, Stern M L, Mahdavi A, Eichler C and Lang E W 2021 Parameterized reinforcement learning for optical system optimization *J. Phys. D: Appl. Phys.* **54** 305104
- [40] Ardizzone L, Kruse J, Lüth C, Bracher N, Rother C and Köthe U 2021 Conditional invertible neural networks for diverse image-to-image translation (<https://doi.org/10.48550/arXiv.2105.02104>)
- [41] Belyaeva A, Lala S, Shady M and Liu M 2018 Evaluation of mode collapse in generative adversarial networks *IEEE HPEC 2018 proc.* (available at: https://iee-hpec.org/2018/2018program/index_html_files/124.pdf)
- [42] Dinh L, Krueger D and Bengio Y 2015 Nice: non-linear independent components estimation *ICLR (Workshop Board 27)* (arXiv:1410.8516)
- [43] Dinh L, Sohl-Dickstein J and Bengio S 2016 Density estimation using real NVP *CoRR* (arXiv:1605.08803)
- [44] Kingma D P and Dhariwal P 2018 Glow: generative flow with invertible 1×1 convolutions *Neurips 2018 proc.* vol 31 (available at: <https://proceedings.neurips.cc/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf>)
- [45] Papamakarios G, Pavlakou T and Murray I 2017 Masked autoregressive flow for density estimation *Neurips 2017 proc.* vol 30 (available at: <https://proceedings.neurips.cc/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf>)
- [46] Byrnes S J 2019 Multilayer optical calculations (arXiv:1603.02720)
- [47] Luce A, Mahdavi A, Marquardt F and Wankel H 2022 Tmm-fast, a transfer matrix computation package for multilayer thin-film optimization: tutorial *J. Opt. Soc. Am. A* **39** 1007–13
- [48] Ardizzone L, Kruse J, Wirkert S, Rahner D, Pellegrini E W, Klessen R S, Maier-Hein L, Rother C and Köthe U 2019 Analyzing inverse problems with invertible neural networks (arXiv:1808.04730)
- [49] Nelder J and Mead R 1965 A simplex method for function minimization *Comput. J.* **7** 308–13