Research Article | Vol. 29, No. 3 / 1 February 2021 / Optics Express 3841
Optics EXPRESS

Deep learning-enabled framework for automatic lens design starting point generation

GEOFFROI CÔTÉ,[1,2,*] JEAN-FRANÇOIS LALONDE,[2] AND SIMON THIBAULT[1]

[1]Centre d'optique, photonique et laser, Université Laval, Quebec City, G1V 0A6, Canada
[2]Laboratoire de vision et systèmes numériques, Université Laval, Quebec City, G1V 0A6, Canada
[*]geoffroi.cote.1@ulaval.ca
https://lvsn.github.io/lensnet

Abstract: We present a simple, highly modular deep neural network (DNN) framework to address the problem of automatically inferring lens design starting points tailored to the desired specifications. In contrast to previous work, our model can handle various and complex lens structures suitable for real-world problems such as Cooke Triplets or Double Gauss lenses. Our successfully trained dynamic model can infer lens designs with realistic glass materials whose optical performance compares favorably to reference designs from the literature on 80 different lens structures. Using our trained model as a backbone, we make available to the community a web application that outputs a selection of varied, high-quality starting points directly from the desired specifications, which we believe will complement any lens designer's toolbox.

1. Introduction

Learning-based approaches, in particular those that build on recent advances in deep learning (DL) [1], have shown promising benefits over traditional approaches in many optical signal processing applications such as superresolution microscopy [2,3] or computational imaging [4,5]. In contrast to standard classification or regression problems, DL does not lend itself as naturally to design problems in optics due to the scarcity of relevant designs in datasets, generally harsh cost function landscapes (presence of discontinuities, abundance of poor local minima), and the unsuitability of DL to one-to-many mappings. Nevertheless, applications using DL have emerged in the design of metasurfaces [6,7], thin films [8], freeform surfaces [9] and nanostructures [10].

Lens design in particular is an area where there is much to gain by casting the problem as a learning task, thus taking advantage of known and successful design forms. In contrast, most known lens design approaches such as evolutionary algorithms [11–13] and other global optimization methods [14], local search [15,16], and saddle point construction [17], do not make use of previous knowledge other than by the mechanism of the starting point, and therefore must start from scratch with every new design process.

Because of the importance of starting points in lens design optimization [18,19], and to alleviate the hassle of parsing through databases and reoptimizing approximate matches to new specifications, [20] used DL to infer lens design starting points directly from the desired specifications. It was shown that a deep neural network (DNN) model could be trained to automatically produce high-quality starting points by extrapolating from known designs, though the experiments were conducted on simple air-spaced and cemented doublets only. The DNN parameters were optimized by applying differentiable ray tracing to the inferred lens designs and by optimizing an optical loss based on the meridional RMS spot size. The ray-tracing module was implemented in the automatic differentiation framework PyTorch [21], which enables backpropagation, a necessary component in training deep neural networks. Moreover, reusing known designs from the literature to provide a supervision signal was key in being able to train

#401590 https://doi.org/10.1364/OE.401590
Journal © 2021 Received 27 Jul 2020; revised 7 Jan 2021; accepted 8 Jan 2021; published 25 Jan 2021

Optics EXPRESS

the DNN to optimize optical performance despite the discontinuities, non-linearity and numerous local minima of typical lens design optimization functions.

In following work [22], the approach was expanded to provide the DNN with a dynamic architecture that allows a single model to adopt a shared representation for different lens structures, in this case any sequence of glass elements or air gaps. The core idea was to use a recurrent neural network (RNN) [23] as the backbone of the model. RNNs have met considerable success on many learning-related tasks involving sequential, variable-length data as in machine translation [24–26] or speech recognition [27]. Likewise, an RNN can capture the sequential structure of lens designs and infer every piece of a design sequentially. Though the use of a dynamic model allowed a shared representation of various lens structures, the aperture stop was assumed to lie on the first surface of the system; thus, the framework was still limited to a restrained number of different lens structures unsuitable to real-world lens design problems.

In this paper, we address most limitations of this previous framework, most notably by allowing the aperture stop to be anywhere in the design, by including vignetting factors as specifications, by inferring glass variables that closely approximate the behavior from the discrete glass materials in the widely used Schott catalog [28], and by implementing a more powerful differentiable ray-tracing module that handles oblique rays and precise aperture stop ray aiming. This allows us to (1) model a significant variety of lens structures that closely reflect real-world problems, (2) learn from a large number of diversified lens designs from the literature, which allows the DNN to learn and reproduce general lens design practices and tendencies, and (3) infer high-quality starting points on a wide variety of lens structures and specifications. This framework is a promising tool for both novice and experienced lens designers, as its parameters can be saved and shared for future use once it is trained, and the only task required of the user in order to obtain high-quality starting points is to enter their desired specifications as input.

We successfully train our DNN model on 80 different lens structures by extrapolating from 150 reference designs found in the literature [29]. Our trained model acts as a backbone for a web application called LensNet that we make readily available to the community for starting points queries [30]. In the paper, we provide a qualitative demonstration of the web application to show the simplicity of the framework for users: for the desired combination of effective focal length (EFL), f-number (defined by EFL/EPD) and half field of view (HFOV), the DNN model can be queried simultaneously for all appropriate lens structures, then the lens designer can choose among the inferred selection of starting points for a suitable compromise between the number of optical elements and the performance achieved. Unlike lens design databases, our framework provides starting points from diverse lens configurations for the same first-order specifications, which means that their performance can be directly compared on an equal footing without the need to fine-tune each of them to the same specifications. Additionally, we provide a quantitative assessment of the optical performance of the designs based on RMS spot size and distortion, and show that they compare favorably to the reference designs.

2. Method

Our method is summarized as follows: the DNN model takes as input the desired specifications (EPD, HFOV, vignetting factors and lens structure), and directly infers all the variables required to fully define a lens design, namely the curvatures c, thicknesses t, refractive indices n_{d} and Abbe numbers v. We consider the object at infinity and use a solve on the last curvature to enforce a unit EFL so the model doesn't have to learn how to scale designs. Those are scaled to the desired EFL outside training.

We train the DNN model using a combination of supervised and unsupervised training schemes, as illustrated in Fig. 1. With supervised training, reference lens designs found in the literature are used to guide the training, by successively (1) giving the specifications that those lens designs were optimized for as inputs to the model, and (2) minimizing a supervised loss term by stochastic

Optics EXPRESS

gradient descent (SGD) so that the model replicates the reference designs. Supervised training alone cannot successfully generalize to unseen specifications because lens designs found in databases are too scarce. Thus, we combine this with unsupervised training in which we compute and optimize the optical performance of the designs using our differentiable ray-tracing module. We jointly and successively apply the following unsupervised training steps until convergence: (1) a large variety of input specifications are given to the model, which infers a lens design for every set of specifications, (2) the optical loss function is computed from the RMS spot size of the inferred designs as well as penalty terms to penalize ray failures, overlapping surfaces, poor glass element shapes, and glass variables deviating from Schott glasses [28], and (3) the model parameters are updated by SGD so that the optical performance of all designs is optimized. Intuitively, the model learns good priors by being forced to reproduce reference designs with supervised training, and this allows unsupervised training to narrow its search within known design forms instead of exploring the whole solution space and getting stuck in poor local minima.

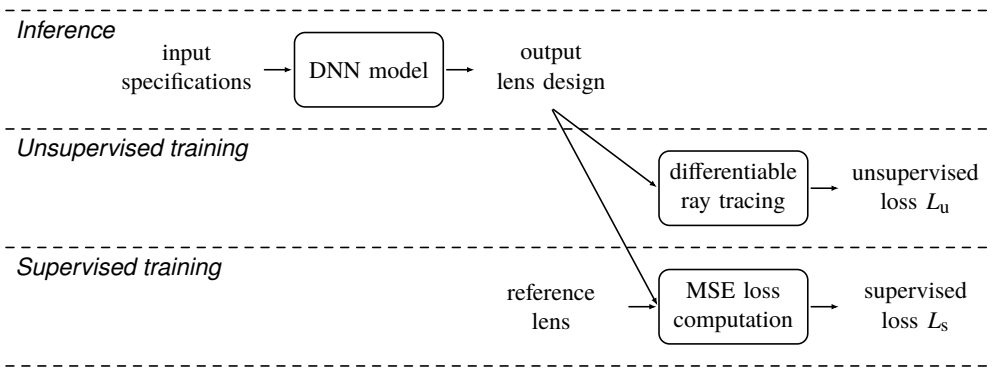

Fig. 1. Overview of the DL framework. The model learns to maximize optical performance and design viability through unsupervised learning. The process is assisted by injecting knowledge from reference designs through supervised training.

2.1. Deep neural network model

Our DNN architecture builds on the dynamic architecture introduced in [22]. In Fig. 2 is shown a schematic view of the DNN model used to infer the lens designs, configured for a single-lens design for clarity. We use a lens structure-agnostic model architecture: the same curvature ("c"), glass ("g"), thickness ("t") and aperture stop ("s") building blocks are rearranged dynamically to support any sequence of glass elements or air gaps and any aperture stop location; thus, a single model with a predefined number of parameters can be trained or queried simultaneously on all desired lens structures.

The model inputs consist of the input sequence, which is used to dynamically arrange the building blocks, as well as the input specification vector composed of the desired EPD, HFOV and vignetting factors v_h and v_l. A randomly generated vector $\mathbf{z}$ completes the specification vector, only to allow a variety of lens designs to be inferred from a given set of specifications, i.e. a one-to-many mapping instead of one-to-one. This prevents a problem that occurs in the presence of multiple reference designs for the same lens structure and specifications, in which the model erroneously tries to reproduce the average of the designs instead of reproducing each design separately.

In the DNN, four learnable linear projection layers $m_{e,s}$, $m_{e,c}$, $m_{e,g}$ and $m_{e,t}$ are used to generate high-dimensional embeddings based on the specification vector and the element type. There, a dynamic network m_d collects and processes the variable-length sequence of embeddings. We use a variant of a recurrent neural network (RNN) model to represent m_d, namely a stacked

Optics EXPRESS

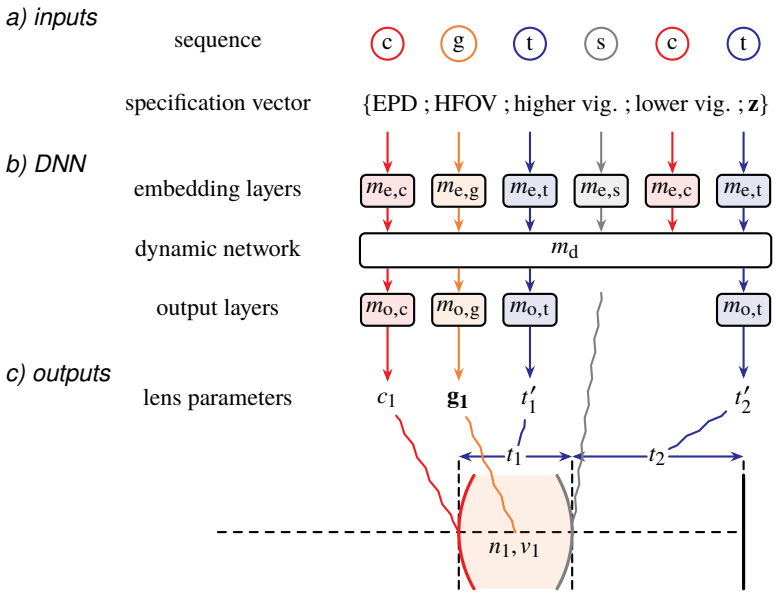

Fig. 2. Dynamic DNN architecture used to learn multiple lens structures at once. The configuration – illustrated here for single-lens designs where the second surface acts as the aperture stop – is changed dynamically to accommodate any desired lens structure. **a)** The input sequence specifies how the curvature ("c"), glass ("g"), thickness ("t") or aperture stop ("s") building blocks are to be arranged, while the specification vector is fed to the embedding layers. **b)** The embedding layers generate a variable-length sequence of high-dimensional embeddings, which are processed by the dynamic network, then the output layers map the variable-length sequence of output vectors to lens parameters. All components m of the model have learnable parameters. **c)** The intermediate lens parameters are converted to viable parameters using non-learnable functions. The last curvature is solved for to obtain a unit EFL.

bidirectionnal [23] gated recurrent unit (GRU) [24] network. For the current experiments, we instantiate m_d with 6 layers and an input dimensionality of 128. Next, three learnable linear projection layers $m_{o,c}$, $m_{o,g}$ and $m_{o,t}$ transform the high-dimensional dynamic model outputs into the curvatures c, glass variables g_1, g_2 and normalized thicknesses t'.

In a final step, the normalized thicknesses are smoothly rectified to positive values according to $t = \ln(1 + \exp(t'))$ while the glass variables are converted to viable refractive indices n_d and Abbe numbers v.

2.2. Glass model

As refractive indices n_d and Abbe numbers v of real glass are correlated, we use a disentangled representation for our glass variables $\mathbf{g}$ by computing the principal component analysis (PCA) on real glass variables from the Schott Catalog [28]. Then, we go from one representation to the other using the PCA model:

$$g_1, g_2 = \mathrm{PCA}(n_d, v)\ . \tag{1}$$

In contrast to [20], we allow the glass variables to have any value. We transfer the task of bounding the glass variables to the unsupervised loss function instead.

In this work, we consider that the partial dispersion of all inferred glasses varies linearly with the Abbe number v and use the same model as Smith [31].

Optics EXPRESS

2.3. Training domains

When training the model using unsupervised steps, for every lens structure we must decide on a training domain $\mathcal{D}_s$ from which we will uniformly and randomly sample the specifications (EPD, HFOV, v_{h} and v_{l}), except for the random vector $\mathbf{z} \sim U(-1, 1)$. Each lens structure has a specific training domain that is chosen to encompass the specifications of the reference lens designs used for supervision (see Fig. 3). For the current experiments, we follow a simple rule: for every reference design r, we establish a domain $\mathcal{D}_r$ between 80% and 120% of the HFOV and EPD of the reference design. Likewise, we define a range of vignetting factors for $\mathcal{D}_r$ by adding or subtracting 0.05 to the reference values, but avoiding negative vignetting factors. Then, to obtain a single domain $\mathcal{D}_s$ per lens structure, we combine the domains $\mathcal{D}_r$ belonging to the structure so that $\mathcal{D}_s$ encompasses all $\mathcal{D}_r$. When querying the model, the specifications entered must be within the training domains, otherwise it will generalize poorly.

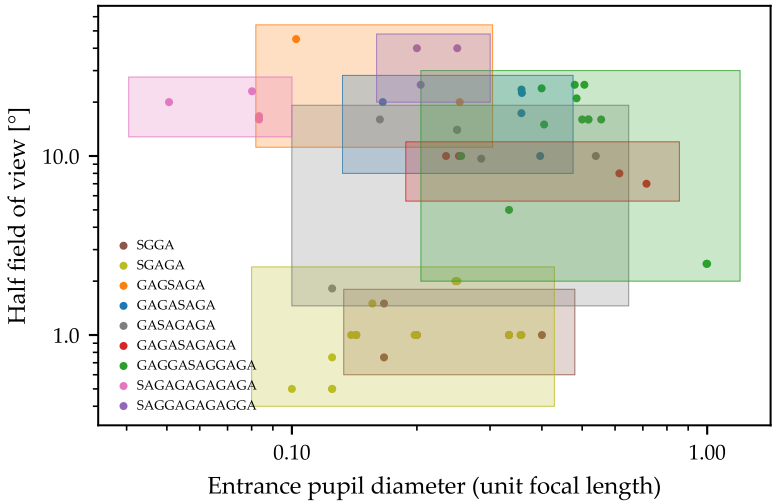

Fig. 3. Domains on which the model is trained for different lens structures, named after their sequence of **G**lass elements, **A**ir gaps and aperture **S**top. The dots represent reference designs used for supervision in our experiments. With a unit EFL, the EPD is the inverse of the f-number. Only lens structures with at least 3 reference designs are shown to avoid clutter.

2.4. Optical performance-based unsupervised loss function

Here we describe the loss function used to improve the optical performance of designs inferred on a wide variety of specifications. The high-level goal of the unsupervised loss function is to provide functional starting points that are optimized for image quality. Thus, we target four factors: (1) optical performance, estimated by the RMS spot size of the designs at the image plane, (2) ray behavior, as to avoid overlapping surfaces, total internal reflexion (TIR) and missed surfaces, (3) glass materials, so that the inferred glass materials closely approximate the behavior of real and widely available glass materials, and (4) glass element shape, which should be neither too brittle nor too thick. To this end, we use our ray-tracing module implemented in PyTorch [21], which enables us to (1) compute the derivatives of the loss with automatic differentiation instead of numerical differentiation, (2) exploit GPU computing capabilities by parallelizing the ray tracing on many designs at once, regardless of the lens structure, and (3) have complete access to intermediate ray-tracing data and use them to shape our loss function. Our ray-tracing module is more powerful than the one described in [20], as it considers vignetting, aperture stop

Optics EXPRESS

placement, precise aperture stop ray aiming, and skew ray tracing that covers the whole entrance pupil instead of the meridional plane only. Similar to previous work [20], we use RMS spot size as our performance metric instead of, e.g., modulation transfer function (MTF), wavefront error or Seidel aberrations, because it can be computed efficiently and usually provides a good estimation of optical performance for non-diffraction-limited systems.

Ray tracing and spot size. For all lens designs, we consider three field angles H that correspond to {0, 0.7, 1} times the HFOV, three wavelengths w consisting of the C, d and F Fraunhofer lines, and 64 pupil intersections p that span the pupil uniformly in a pseudo-random way. With the object at infinity, we initialize the rays at the paraxial entrance pupil and propagate them across all surfaces k up to the image plane. For simplicity, we only consider two vignetting factors v_h and v_l – also given as inputs to the model – that respectively represent the higher and lower vignetting factors on the y-axis, and assume negligible vignetting on the x-axis. When applying vignetting, the relative y_p coordinates at the entrance pupil are updated according to:

$$y'_\mathrm{p} = \left(1 - \lambda_H \frac{v_\mathrm{h} + v_\mathrm{l}}{2}\right) y_\mathrm{p} - \lambda_H \frac{v_\mathrm{h} - v_\mathrm{l}}{2} , \tag{2}$$

where λ_H is a field-dependent parameter that we set to {0, 0.6, 1} for the fields {0, 0.7, 1}.

We apply a ray-aiming correction to off-axis rays to prevent them from deviating from the real aperture stop in the presence of pupil aberrations. We consider a linear relationship between the entrance pupil coordinates y_p and the aperture stop coordinates y_s:

$$\Delta y_\mathrm{s} \approx \Delta y_\mathrm{p} \frac{\mathrm{d}y_\mathrm{s}}{\mathrm{d}y_\mathrm{p}} . \tag{3}$$

We can compute the derivative term nearly exactly by tracing real marginal rays from the paraxial entrance pupil to the aperture stop, then applying automatic differentiation. We compute the real aperture stop diameter by tracing an on-axis marginal ray. We then apply a single correction step to all computed x_p and y_p coordinates for off-axis rays so that the marginal meridional and sagittal rays hit the border of the real aperture stop. In practice, we found that this single correction step guarantees negligible ray-aiming error for most considered lens designs. When choosing reference designs, we screen out those that have non-negligible ray-aiming error even after this correction step (e.g., wide-angle lenses).

We obtain the mean RMS spot size s by computing the field-wise RMS spot size over all wavelengths and pupil intersections, and by averaging the results over the fields:

$$s = \operatorname*{avg}_H \left[\sqrt{\operatorname*{avg}_{w,p} \left[(x_{w,p} - \bar{x})^2 + (y_{w,p} - \bar{y})^2 \right]} \right] . \tag{4}$$

Penalty terms. Ray-tracing failures, namely missed surfaces and TIR, happen respectively when the angle of incidence I or the angle of refraction I' are invalid. These situations can be avoided by enforcing $|\sin(I)|<1$ and $|\sin(I')|<1$. Rays that encounter ray failures are ignored from further computations. Another form of undesired ray behavior is backward traveling, which is used to detect and penalize situations where optical surfaces overlap. This is avoided by enforcing that no ray at a given surface has a negative longitudinal displacement: $\Delta z>0$. We design a ray behavior penalty term q_bvr for a given lens design by combining those constraints in a piecewise-differentiable way to provide a well-behaved training signal:

$$q_\mathrm{bvr} = \operatorname*{avg}_{H,w,p,k} \left[\max(|\sin I_{Hwpk}| - 1, 0) + \max(|\sin I'_{Hwpk}| - 1, 0) + \max(-\Delta z_{Hwpk}, 0) \right] . \tag{5}$$

In [20], glass variables were bounded to prevent them from deviating too far from the average glass in the Schott catalog [28]. Here, as glass variables are not bounded, we need to include

Optics EXPRESS

a glass penalty term q_{glass} to keep them close to real glass variables. For every inferred glass variables $\mathbf{g}_m$ of a lens design, where m iterates over all glass elements, we compute its L2 distance with respect to the nearest neighbor $\mathbf{g}_{m,\text{NN}}$ from the Schott catalog, and we average:

$$q_{\text{glass}} = \underset{m}{\text{avg}} \left[\|\mathbf{g}_m - \mathbf{g}_{m,\text{NN}}\|_2 \right] . \tag{6}$$

To output good starting points, we also want to avoid glass elements that are too thick or too brittle. We design a lens element shape penalty term q_{shape} that targets each glass element's diameter-to-thickness ratio R_m. We define lower and upper thresholds R_{l} and R_{u} that we set to 3 and 12, respectively, and penalize values outside this range linearly:

$$q_{\text{shape}} = \underset{m}{\text{avg}} \left[\max \left(R_m - R_{\text{u}}, -(R_m - R_{\text{l}}), 0 \right) \right] . \tag{7}$$

Unsupervised loss. The unsupervised loss for a single design is the product of the spot size and the shaped penalty terms:

$$L_{\text{u}} = s(1 + \lambda_{\text{bvr}} q_{\text{bvr}})(1 + \lambda_{\text{glass}} q_{\text{glass}})(1 + \lambda_{\text{shape}} q_{\text{shape}}) , \tag{8}$$

where λ_{bvr}, λ_{glass} and λ_{shape} are parameters used to scale the behavior, glass and shape penalty terms that we set to 10000, 10 and 0.01, respectively. The choice of using multiplicative instead of additive loss terms, although uncommon in the machine learning literature, has done well on this learning task as it allows us to weigh the penalty terms easily – despite the mean spot size decreasing by more than a hundredfold during training.

When we average the mean unsupervised loss for the entire batch before updating the DNN parameters, we use a geometric mean instead of an arithmetic mean to scale all designs equally, independently of the magnitude of the RMS spot size:

$$\overline{L_{\text{u}}} = \exp \left(\underset{b}{\text{avg}} \left[\log L_{\text{u},b} \right] \right) . \tag{9}$$

Since optimizing for RMS spot size alone can lead to designs whose performance has a high sensitivity to manufacturing precision, we add tolerancing noise from a normal distribution ($\sigma = 10^{-3}$) to all curvatures before ray tracing during training. It is an uncomplete but efficient way of addressing the issue of tolerancing in lens design optimization.

2.5. Supervised and joint loss

A supervised training sample is composed of a single reference lens design and the specifications that this design was optimized for. Before training, every reference design is given a random vector $\mathbf{z}$ that is held fixed throughout training. We set the size of $\mathbf{z}$ to 2, which is sufficient to allow the model to reproduce different reference designs with the same specifications. The supervised loss function for a given training sample is simply the MSE between all lens variables θ_k inferred by the model for the given specifications and their reference values θ'_k:

$$L_{\text{s}} = \underset{k}{\text{avg}} \left[(\theta_k - \theta'_k)^2 \right] , \tag{10}$$

where $\boldsymbol{\theta} = \{c_1 \dots c_{N-1}, t_1 \dots t_N, g_{1,1}, g_{1,2} \dots g_{M,1}, g_{M,2}\}$, N is the number of optical surfaces excluding the image plane, and M is the number of glass elements.

In contrast to the unsupervised loss, we average a batch of samples using the arithmetic mean:

$$\overline{L_{\text{s}}} = \underset{b}{\text{avg}} \left[L_{\text{s},b} \right] . \tag{11}$$

In practice, we train the model by successively applying joint training steps until convergence: (1) we randomly sample an unsupervised batch and a supervised batch of 512 training samples

Optics EXPRESS

each, (2) we compute the respective mean losses according to Eq. (9) and Eq. (11), (3) we combine them according to $L_\mathrm{j} = \overline{L_\mathrm{u}} + \lambda_\mathrm{s}\overline{L_\mathrm{s}}$, where λ_s is set empirically to 0.2, (4) we compute the derivative of L_j with respect to every DNN parameter using automatic differentiation, and (5) we update the weights using the Adam optimizer [32], which is a variant of SGD.

2.6. Training details

When generating training samples, we sample between all possible lens structures evenly for unsupervised training, and between all reference designs evenly for supervised training. We set the values $\beta = (0.9, 0.99)$ for the Adam optimizer and use gradient clipping with a threshold of 0.1 to prevent blow-ups due to the strong non-linearity of ray tracing. We train over 500000 steps with a warm-up phase where the learning rate is linearly increased from 4×10^{-6} to 4×10^{-4} over 100000 steps, then progressively decayed to the original value over the remaining steps following a cosine half-cycle. Training takes about 50 hours using a single Nvidia Tesla P100 GPU. In contrast, once the model is trained, inference time is very short as about 20000 designs can be inferred per second, assuming a large batch size (4096) to fully load the GPU.

3. Results

We train a single instance of the model using the previously described method to produce the following results, with supervision from 150 reference designs from the Zebase 6 collection [29] shared among 80 different lens structures. We use the model to infer various lens designs and assess their performance both qualitatively and quantitatively.

3.1. Demonstration and web interface

To provide a more convenient way of using our framework, we created an interface that automatically generates a selection of starting points and ensures that the designs are inferred within the training domains. An example of results obtained when using this interface is provided in Fig. 4. Here, the user supplies the desired EFL, f-number and HFOV, and the model generates one lens design for each lens structure relevant to these specifications, i.e. whose training domain encompasses those specifications. The vignetting factors are selected automatically as the minimum values in the respective training domains, as to minimize vignetting. The inferred designs are scaled to satisfy the required EFL. Then, the user may compare the designs on different performance metrics, e.g. with distortion, field curvature, lateral color or ray fan plots, which should provide a rough idea of the limitations of each lens structure. In contrast to starting points found in traditional lens design databases, here all the suggested designs are preoptimized for the same first-order specifications, and as such can be compared on equal grounds. In practice, the user could select a given design, e.g. one with low manufacturing complexity that approximately meets the requirements of the problem at hand, and further optimize it using available optical optimization software. This interface, called LensNet, is hosted online [30] (see Visualization 1 for a demonstration video).

3.2. Design viability

In Table 1, we gather statistics to show the viability of the inferred lens designs. The statistics are gathered from 100 000 lens designs inferred in equal proportion from 80 lens structures, with specifications drawn randomly and uniformly in the respective training domains $\mathcal{D}_\mathrm{s}$. Experiments indicate that only a slight proportion of rays experience failure (2.7×10^{-6}) or backward traveling (9.3×10^{-5}), and that the vast majority of the inferred glass elements have suitable diameter-to-thickness ratios (6.3 ± 3.9).

To assess the feasibility of the inferred glass variables, for each inferred pair of variables (g_1, g_2) we find the closest Schott catalog glass in the normalized $\mathbf{g}$-space, then compute the

Optics EXPRESS

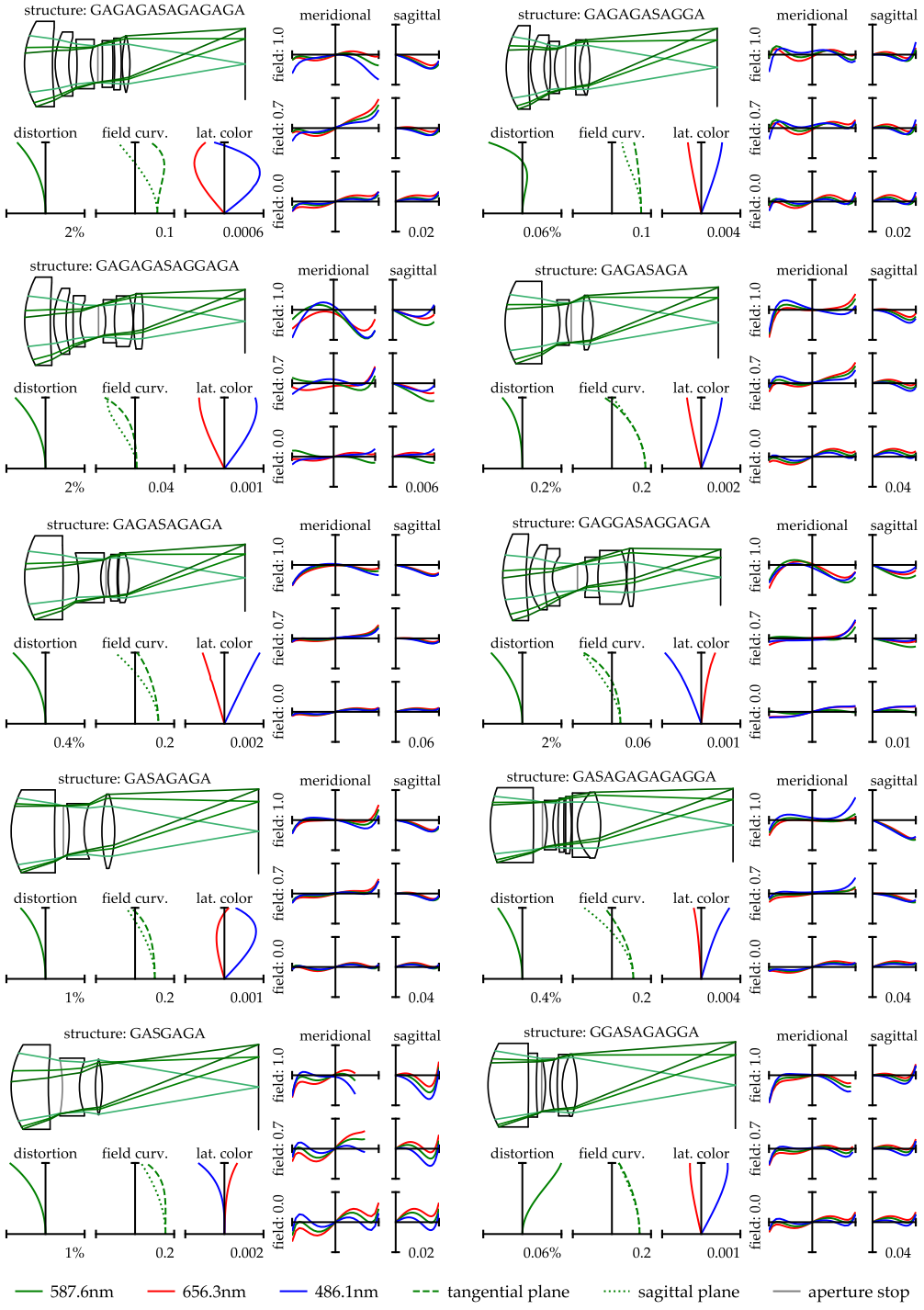

Fig. 4. Designs obtained automatically when querying the framework for a 25 mm EFL, a f-number of 3 and a HFOV of 12°, without manual selection, editing or additional optimization. Some common aberration plots are shown (units are in mm). The framework is used to produce a design for every lens structure whose domain includes these specifications.

Optics EXPRESS

Table 1. Statistics on the viability of the inferred designs regarding the fulfillment of the first-order specifications, the diameter-to-thickness ratio of the glass elements, and the correspondence between the inferred glass variables and real glass materials.

Proportion of ray failures	2.7×10^{-6}
Proportion of backward-traveling rays	9.3×10^{-5}
Diameter-to-thickness ratio of glass elements	6.3 ± 3.9
Refractive index deviation from closest Schott glass	$(1.2 \pm 2.8) \times 10^{-4}$
Abbe number deviation from closest Schott glass	$(1.5 \pm 4.0) \times 10^{-2}$

difference between the two refractive indices n_d and Abbe numbers v. Based on the result, we see that the average deviation between inferred glass materials and their closest neighbor in the Schott catalog is minor. For reference, our average deviation in refractive indices and Abbe numbers is below standard manufacturing tolerances (called Step 3), of $\pm 5 \times 10^{-4}$ for refractive indices and $\pm 0.5\%$ for Abbe numbers (about ± 0.1 in a best-case scenario) [33].

3.3. Optical performance

The fact that the performance of a design is ultimately defined by its suitability to the problem at hand, for which there exists no universal metric, combined with the scarcity of expertly-designed lenses to compare with, precludes us from making a satisfying global assessment of the performance of the inferred designs. Instead, in Fig. 5 we settle on showing that the performance of the inferred designs does not deviate very far from that of the reference designs, while recognizing that optical performance, e.g. RMS spot size, cannot be compared directly for different EPD, HFOV or vignetting factors. For every reference design r, we infer 1000 designs from sets of specifications that are randomly and uniformly sampled from the domain $\mathcal{D}_r$. We average their performance on two different metrics, RMS spot size and distortion at full field of view, using the geometric instead of the arithmetic mean since the performance can sometimes span multiple orders of magnitude. We compare their performance to the reference designs. When a given structure is represented by more than one reference design, we average both the mean performance of the inferred designs and the performance of the reference designs by applying the geometric mean in order to compute the final score.

From Fig. 5, we see that the optical performance of the inferred designs, as measured by the RMS spot size, compares favorably to the reference designs for most lens structures. Distortion, on the other hand, more often than not exceeds the reference value. This can be attributed to our unsupervised training scheme since it only takes RMS spot size into account and is thus blind to distortion. However, in most cases where the distortion strongly exceeds the reference value, the RMS spot size will be significantly lower than the reference, meaning that the inferred starting points trade distortion correction for sharper image quality.

In some cases, the RMS spot size of our designs is well below the reference value. This may be because not all reference designs were optimized for performance alone; some may have additional performance-limiting constraints such as targeting a specific back focal length (BFL) or using specific glass materials.

3.4. Ablation study

In Table 2, we show the impact of dropping some of our methodology choices, which is commonly referred to as an ablation study in the machine learning literature. We compare the performance of different model instances by computing the geometric average of the spot size from 100 000 lens designs inferred in equal proportion from the 80 lens structures, with specifications drawn randomly and uniformly in the respective training domains $\mathcal{D}_s$.

Optics EXPRESS

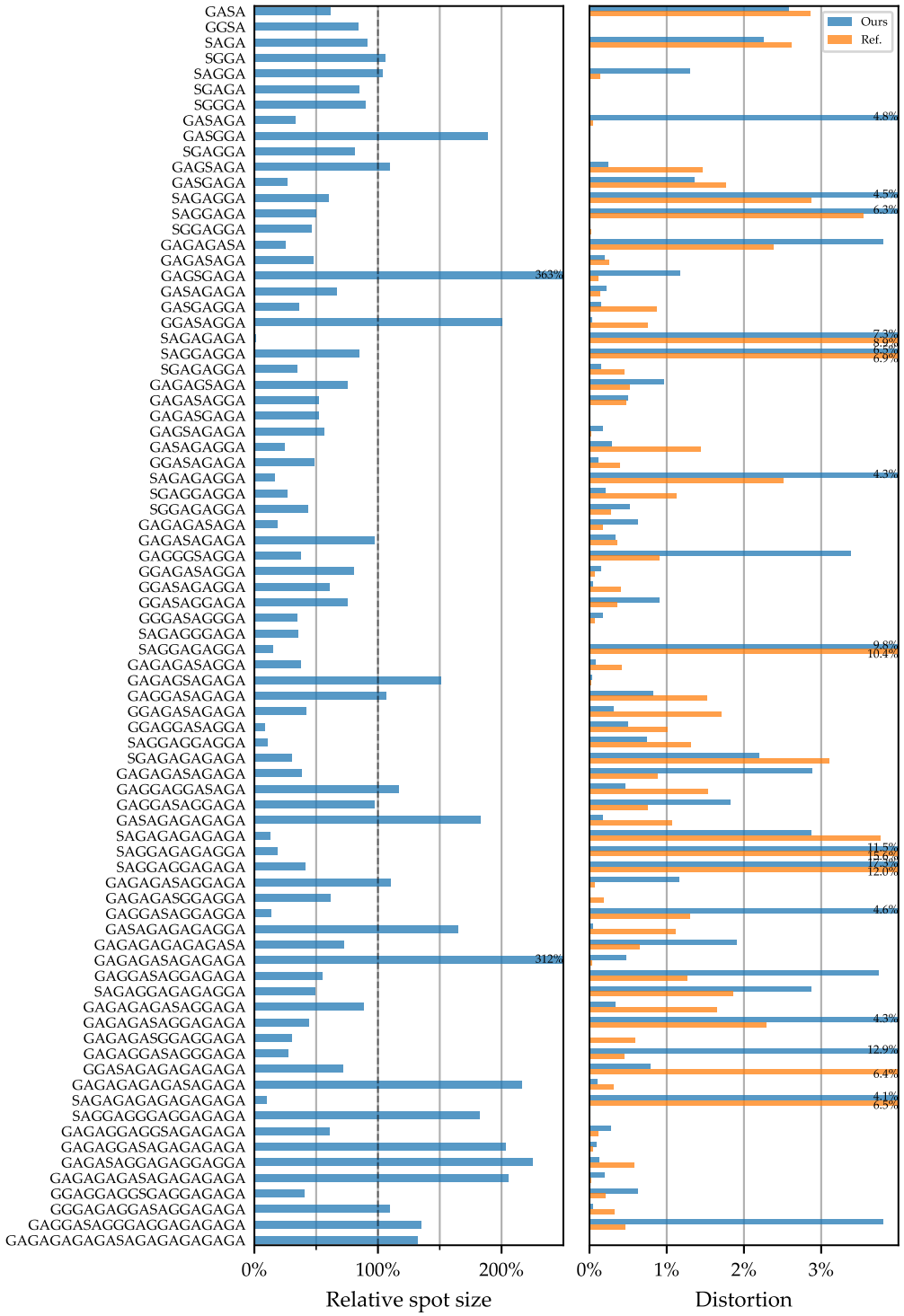

Fig. 5. Mean performance of designs inferred from specifications randomly and uniformly sampled around the specifications of every reference design r (lower is better). The relative spot size for a given structure is the mean spot size of the inferred designs normalized with the reference design spot size.

Optics EXPRESS

Table 2. Statistics on the inferred designs after training with different methodology choices. We show the geometric mean of the RMS spot size of designs inferred from the model (with unit EFL) with and without inclusion of tolerancing noise during evaluation. Also shown when relevant is the distribution of the diameter-to-thickness ratio of the inferred glass elements.

Training methodology	Noise in eval.	Ratio distribution	Spot size
Baseline	No	6.3 ± 3.9	0.000117
	Yes	–	0.000170
No tolerancing noise	No	–	0.000111
	Yes	–	0.000176
No glass element shape penalty	No	14 ± 24	0.000119
No ray behavior penalty	No	–	Diverges
No supervision signal	No	–	0.000394

We train the model with and without tolerancing noise or supervision signal, and evaluate it with and without tolerancing noise. From the results, we see that omitting tolerancing noise during training leads to a larger decrease in performance when we include the tolerancing noise during the evaluation. This suggests that adding tolerancing noise on the curvatures during training makes the designs less sensitive to fabrication tolerances.

We can also see that the inclusion of the glass element shape penalty is necessary in order to obtain viable glass element diameter-to-thickness ratios. Otherwise, the model tends to output glass elements that are overly brittle or thick. Fortunately, there is no performance cost associated with this penalty term as is seen by comparing the average RMS spot size. This may be attributed to the known lens design practice that interchanging glass and air thicknesses is oftentimes unimportant compared to other variables.

Omitting the ray behavior penalty term makes the training diverge because the optimization process can no longer keep track of the rays that experienced ray-tracing failure. On this note, when the model is randomly initialized at the beginning of the training process, an overwhelming proportion (>40%) of the traced rays experience ray failure. This shows just how important it is to target them in the optical loss function. In contrast, traditional lens design often neglects this aspect since good starting points should already be devoid of ray failures.

Finally, omitting the supervision signal during training (or equivalently, setting λ_s to 0) results in a very poor mean RMS spot size. This is not surprising: the model generates random lens designs at first because of its initial random parameterization; thus, omitting the supervision signal amounts to solving millions of lens design problems simultaneously with deficient starting points, i.e. a process bound to fail. Incidentally, when the size of the random vector $\mathbf{z}$ is too large, the model seems to ignore the other specifications and this results in a behavior similar to the omission of the supervision signal.

4. Conclusion

Here, we have introduced a framework that enables users to automatically obtain high-quality lens design starting points that are tailored to the desired specifications for a wide variety of lens structures. The framework is hosted online and readily available for starting point queries, which we believe will be useful both to experienced and novice lens designers. We have shown that the optical performance of the inferred designs compares favorably to expertly-designed lenses from the literature, as the mean RMS spot size and distortion of the inferred designs are lower than the reference values in 75% and 49% of the lens structures considered, respectively. The inferred glass materials are realistic and their mean deviation from Schott glasses is below the standard manufacturing tolerances.

Optics EXPRESS

Our DNN model has a highly modular architecture that could be expanded in the future. We are excited about the possibilities of scaling this framework to more applications and more lens components, e.g. by adding more types of building blocks for aspherical and freeform surfaces, GRIN components, etc., or by refining the vignetting model.

In this work, we used RMS spot size as our only optical performance target. It may be desirable for the lens design community to extend our framework to custom merit functions with other performance targets (distortion, MTF, encircled energy), constraints (BFL, TTL), or wavebands (SWIR, MWIR, LWIR). Adding inputs for every possible requirement would increase the number of reference designs needed as well as the training complexity, whereas adding arbitrarily weighted specific constraints and performance targets to the optical loss function would introduce a loss of generality. Instead, one possibility would be to use a form of meta-learning in which the model could quickly adapt to custom optical loss functions. This could be an important area of improvement for this framework in the future.

Another area of improvement would be to force the model to output more diverse lens designs for the same structure and specifications. Injecting noise with the random vector $\mathbf{z}$ is a step in this direction, but the model can ignore it quite easily during learning as its only goal is to optimize the optical loss function. Research on generative models suggests more sophisticated ways of generating diverse outputs, which is something that could be explored in the future.

We believe our tools could be useful in the emerging field of end-to-end optics design, which consists in modeling the image acquisition and processing pipeline in an end-to-end differentiable way, then simultaneously optimizing every component involved with respect to the downstream task. Doing this, instead of optimizing the optics for traditional image quality metrics, one can design optical components that outperform traditional optics at a specific task, e.g. achromatic extended depth of field and super-resolution imaging [34], high dynamic range estimation [35,36] and depth estimation [37,38]. In this regard, the use of our differentiable ray-tracing module to compute the optical point spread function (PSF) could enable the use of multi-element lens designs, while our lens design generating model could be used upstream to infer lens designs specialized to a given computer vision task.

Funding. Natural Sciences and Engineering Research Council of Canada; Canada First Research Excellence Fund.

Acknowledgments. This research was supported by the Natural Sciences and Engineering Research Council of Canada and by the Sentinel North program of Universit Laval, made possible, in part, thanks to funding from the Canada First Research Excellence Fund.

Disclosures. The authors declare no conflicts of interest.

References

1. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature **521**(7553), 436–444 (2015).
2. Y. Rivenson, Z. Göröcs, H. Günaydin, Y. Zhang, H. Wang, and A. Ozcan, "Deep learning microscopy," Optica **4**(11), 1437–1443 (2017).
3. E. Nehme, L. E. Weiss, T. Michaeli, and Y. Shechtman, "Deep-STORM: super-resolution single-molecule microscopy by deep learning," Optica **5**(4), 458–464 (2018).
4. A. Sinha, J. Lee, S. Li, and G. Barbastathis, "Lensless computational imaging through deep learning," Optica **4**(9), 1117–1125 (2017).
5. G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," Optica **6**(8), 921–943 (2019).
6. Z. Liu, D. Zhu, S. P. Rodrigues, K.-T. Lee, and W. Cai, "Generative Model for the Inverse Design of Metasurfaces," Nano Lett. **18**(10), 6570–6576 (2018).
7. C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla, "Deep learning for accelerated all-dielectric metasurface design," Opt. Express **27**(20), 27523–27535 (2019).
8. R. S. Hegde, "Accelerating optics design optimizations with deep learning," Opt. Eng. **58**(6), 065103 (2019).
9. T. Yang, D. Cheng, and Y. Wang, "Direct generation of starting points for freeform off-axis three-mirror imaging system design using neural network based deep-learning," Opt. Express **27**(12), 17228–17238 (2019).
10. P. R. Wiecha, A. Lecestre, N. Mallet, and G. Larrieu, "Pushing the limits of optical information storage using deep learning," Nat. Nanotechnol. **14**(3), 237–244 (2019).
11. C. Gagné, J. Beaulieu, M. Parizeau, and S. Thibault, "Human-competitive lens system design with evolution strategies," Appl. Soft Comput. **8**(4), 1439–1452 (2008).

Optics EXPRESS

12. B. F. Carneiro de Albuquerque, F. Luis de Sousa, and A. S. Montes, "Multi-objective approach for the automatic design of optical systems," Opt. Express **24**(6), 6619 (2016).
13. C. Menke, "Application of particle swarm optimization to the automatic design of optical systems," Proc. SPIE **10690**, 106901A (2018).
14. M. Isshiki, H. Ono, K. Hiraga, J. Ishikawa, and S. Nakadate, "Lens Design: Global Optimization with Escape Function," Opt. Rev. **2**(6), 463–470 (1995).
15. J. Meiron, "Damped Least-Squares Method for Automatic Lens Design," J. Opt. Soc. Am. **55**(9), 1105–1109 (1965).
16. M. van Turnhout and F. Bociort, "Chaotic behavior in an algorithm to escape from poor local minima in lens design," Opt. Express **17**(8), 6436–6450 (2009).
17. M. van Turnhout, P. van Grol, F. Bociort, and H. P. Urbach, "Obtaining new local minima in lens design by constructing saddle points," Opt. Express **23**(5), 6679 (2015).
18. M. van Turnhout and F. Bociort, "Instabilities and fractal basins of attraction in optical system optimization," Opt. Express **17**(1), 314–328 (2009).
19. Z. Hou, M. Nikolic, P. Benitez, and F. Bociort, "SMS2D designs as starting points for lens optimization," Opt. Express **26**(25), 32463–32474 (2018).
20. G. Côté, J.-F. Lalonde, and S. Thibault, "Extrapolating from lens design databases using deep learning," Opt. Express **27**(20), 28279–28292 (2019).
21. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proceedings of the 31st Conference on Neural Information Processing Systems*, (2017).
22. G. Côté, J.-F. Lalonde, and S. Thibault, "Introducing a dynamic deep neural network to infer lens design starting points," Proc. SPIE **11104**, 1110403 (2019).
23. M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," in *Proceedings of IEEE Transactions on Signal Processing*, vol. 45 (1997), pp. 2673–2681.
24. K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, (2014), pp. 1724–1734.
25. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems (NIPS)*, (2014), pp. 3104–3112.
26. D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the 3rd International Conference on Learning Representations*, (2015).
27. A. Graves, A.-R. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, (2013), pp. 6645–6649.
28. Schott Corporation, "Optical Glass Catalog," (2019).
29. Zemax Development Corp., "Zebase 6 Optical Design Collection," (2007).
30. G. Côté, "LensNet: lens design starting point generator," https://lvsn.github.io/lensnet.
31. W. J. Smith, *Modern Lens Design* (McGraw Hill Professional, 2004).
32. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proccedings of the 3rd International Conference on Learning Representations*, (2015).
33. R. Jedamzik, S. Reichel, and P. Hartmann, "Optical glass with tightest refractive index and dispersion tolerances for high-end optical designs," Proc. SPIE **8982**, 89821F (2014).
34. V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, "End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging," ACM Trans. Graph. **37**(4), 1–13 (2018).
35. Q. Sun, E. Tseng, Q. Fu, W. Heidrich, and F. Heide, "Learning Rank-1 Diffractive Optics for Single-Shot High Dynamic Range Imaging," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (2020), pp. 1386–1396.
36. C. A. Metzler, H. Ikoma, Y. Peng, and G. Wetzstein, "Deep Optics for Single-Shot High-Dynamic-Range Imaging," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (2020), pp. 1375–1385.
37. H. Haim, S. Elmalem, R. Giryes, A. M. Bronstein, and E. Marom, "Depth Estimation From a Single Image Using Deep Learned Phase Coded Mask," IEEE Trans. Comput. Imaging **4**(3), 298–310 (2018).
38. J. Chang and G. Wetzstein, "Deep Optics for Monocular Depth Estimation and 3D Object Detection," in *Proceedings of IEEE International Conference on Computer Vision*, (2019).