Machine Learning for Starting Points Generation in Optical Lens Design

Zhang, Rui

Award date:
2021

Technische Universiteit
Eindhoven
University of Technology

Department of Computer Science and Engineering
Master Data Science in Engineering

Machine Learning for Starting Points Generation in Optical Lens Design

Master Thesis

Rui Zhang

Supervisors:
Robert Peharz (TU/e)
Pierluigi Frisco (ASML)
Teus Tukker (ASML)

Eindhoven, August 2021

Contents

Abstract

Optical lens design is known to be a formidable problem due to its complex and sensitive cost function landscape, various physical and manufacturing constraints, and its high dependency on the empirical good starting points. In conventional optical lens design, designers start with known designs with satisfying performance, which are known as the starting points, and tune the parameters with their specific goals to obtain a local optima. However, the number of available starting points is limited, and there lacks a general method to arrive at starting points with any given property parameters at first.

In this thesis, we focus on the application of machine learning algorithms to the problem of starting points generation in optical lens design. Specifically, a neural network (NN) is built to generate lens design parameters from the input lens property parameters. A strong data augmentation strategy is adopted to address the issue of limited data. In addition, a hybrid training scheme is designed to take use of the reference designs and the optical key performance indicators (KPIs). To improve the performance on input specifications with worse KPIs, the NN is then integrated with the Bayesian Optimization (BO) framework. The training data for NN are sampled in a smarter way with the acquisition function in BO, to focus on the hard training samples where the NN currently has the worst KPI.

Experimental results show that the hybrid training scheme and the data augmentation are valid to produce lens designs with satisfying KPI values at any given reasonable lens property specifications, and most generated designs outperform the reference designs in terms of the KPIs. The improved sampling with the BO method shows its superiority over the random sampling method in the baseline, which yields a better overall performance across the input space, and a better worst case performance.

Chapter 1

Introduction

In this Chapter, an introduction to optical lens and optical lens design will be given; then the challenges and related work in optical lens design will be introduced, which lead to the motivation and mission of this work. An outline of the remainder of this thesis is given at the end of this Chapter.

1.1 An Overview of Optical Lens Design

Optical lens design is the process of designing a lens system that meets a given set of performance requirements and constraints [1]. Specifically, the design of a lens system involves deciding the shape, position, and material for each element in the lens system. The performance requirements to be satisfied include two aspects. The first aspect include certain functional optical property parameters of the lens, such as the focal length, F#, Field of View (HFoV), etc., which quantify the property of the lens. Formal definitions of these quantities will be given in Chapter 2. It is only necessary for readers to know for now that these parameters characterize the property and application of the lens. For example, focal length is a measure of the optical system's ability to converge or diverge light. A system with a smaller focal length bends the ray more sharply, bringing light to a point called focus in a shorter distance. A telescope normally has a large focal length of more than 1000mm, while for microscopes, the focal length is considerably small, often ranges from 2∼40mm. The second aspect is to ensure the image formed by the lens is of a high quality. We often see distortions in photos captured by a camera, as shown in Figure 1.1. The red rectangles mark the areas with a strong distortion. Real lenses all have such distortions, which are measured by different types of aberrations in optics, such as the Seidel aberrations [2], and the chromatic aberrations [3]. They will also be formally introduced in Chapter 2. For optical designers, it is their mission to design a lens that meets a certain set of property parameters, and to reduce the aberrations to the best. Besides, in real lens systems, there are often physical constraints such as the limitations for the cost specifications, and the manufacturing abilities [1].

Figure 1.2 shows a lens system (a photographic/camera lens) which can be seen in camera shops. Such a lens in Figure 1.2 is normally consisted of multiple glass lens elements, spaced by air or are glued together. During the propagation of light, it will be refracted or reflected at the interface of two dissimilar mediums, according to some physical laws, which will be elaborated in Chapter 2. The shapes, materials and positions of these lens elements determine the way how lights will be bent when traveling through the system, and the process of lens

(a) Geometric distortion (b) Chromatic distortion

Figure 1.1: Representative aberrations exist in photographs. (a) Geometrical aberration [4]. (b) Chromatic aberration [5].

design is to determine these geometric aspects and the materials of each element in a lens system. For brevity, we will refer to a lens system simply as a lens in the remainder of the thesis.

Figure 1.2: An example of the Zeiss Otus 55mm f/1.4 camera lens (up) and the graphical illustration of its internal lens elements (down).

In this project, we will investigate applications of machine learning algorithms to assist optical lens design. The general goal is to determine the geometric and material aspects of elements in a lens to (1) meet certain property parameters, (2) optimize the system's optical performance in terms of least aberrations while satisfying other physical requirements.

1.2 Motivation and Related Work

The conventional optical lens design is a two-step process. The first step is the starting point selection, where optical designers look for an existing solution in the reference database with similar property parameters, known as the starting point. Next, they will perform a local optimization with this starting point in commercial software packages like OpticStudio [6] and CodeV [7]. A customized merit function which takes into consideration their specific goals, and minimizing the aberrations as well as conforming to other physical constraints,

is implemented in the software as the optimization objective. Then the software will use built-in optimization methods to search for new local minimums by tuning some parameters. The search can be combined with methods like evolutionary algorithm [8], particle swarm optimization [9], saddle point construction [10], etc., to converge to better local minimums and maintain the variety in output designs. These methods depend heavily on the starting points that are selected.

However, how to arrive at those good starting points in the first place, is a more difficult question to answer. Although through hundreds of years' accumulation in optical design, lens designs from patents and literature have formed an empirical collection of reference designs, however the amount of them is still very limited. Apart from this, these lens designs constitute discrete points in the property space, and are very different in terms of the number of elements and the types of elements they contained, hence there is no straightforward way to interpolate in the property parameters' space to arrive at new designs. In other words, if we are given a set of desired property parameters of the system with no nearby reference designs in the starting point collection, there is no good way to take use of the knowledge in the database. In addition, the performance is very sensitive to variations in lens parameters [11], and there are many infeasible cases like when the rays cannot reach the image plane (i.e., the total internal reflection, TIR), and overlapping surfaces, which add to the difficulties in optimization.

In this project, we aim at helping with the first step in optical lens design using machine learning techniques: **to generate lens designs that can serve as starting points for optical designers to carry on further optimization.** To be specific, the algorithm should be able to produce lens designs for any given input specifications about the property of the lens, with a satisfying optical performance while meeting other physical requirements.

As this is a rather new research area, applications of machine learning can be found mainly in recent years. Most of them are using deep learning. Deep Neural Networks (DNNs) have gained tremendous popularity during the past decades due to its powerful function approximation ability and the development of the computing power of GPUs, especially in the field of Computer Vision and Natural Language Processing. However, problems of applying DNNs to this problem are also obvious: DNNs require huge amount of data, and the complex cost function of optical performance, and its landscape with discontinuities may give rise to indifferentiable cases in training.

Despite of all these, researchers have been taking first steps into this field by generating designs for specific types. Gannon and Liang [12] designed a network for producing the orthogonal coefficients for freeform lens surfaces, expressed with polynomials. This type of surface enables them to synthesize great amount of data to train a neural network. The network receives as input the system's property parameters like focal length, F#, and HFoV. However, the polynomial freeform surface is expensive in manufacturing. Other similar works can be found in designing thin films [13], metasurfaces [14]. For a sequence of lens elements, Cote et al. [15] first designed a network, to be trained with augmented data from the reference dataset. They optimized the rooted mean squared error (RMS) of the spot size of a bunch of rays on the image plane, which is easy to compute and is also a measure of the image quality. Promising results are obtained for air-spaced doublets with two glass elements. This is later developed with the adoption of a Recurrent Neural Network (RNN) [16][17], which is able to learn and generate different types of designs simultaneously. However, their data augmentation resembles a random sampling, and the network behaves comparably worse with certain input specifications. In addition, the input space scales with the increase of the dimension of the problem (i.e., for more complex lens designs), under which circumstances

training the network sufficiently will require much more training data. Unfortunately, some details are not disclosed, and their code is not public, therefore it is not able to compare to and develop upon their work.

1.3 Our Contributions

In this thesis, we first refer to Cote et al. [15] and implement a network similar to their work. The same data augmentation strategy is adopted. A more informative optical performance KPI, based on the aberration theories, and more physical constraints and infeasible cases are considered. The process of computing our KPIs, which is based on the physical model of rays' propagation, is implemented as a complex differentiable loss function with the network. Experiments show that our implementation with these modifications and improvements has a promising performance, which is able to generate lens designs with a satisfying KPI on any given input specification. Compared to the reference lenses in the dataset, our generated lenses outperform their KPIs. To improve the generated lens' performance in the hard area and have a more efficient global optimization, a Bayesian Optimization (BO) based framework is proposed and incorporated with the baseline model. It is aimed for training the network with the most valuable samples based on the current performance of the network. Compare to our baseline model, the improved BO framework is shown to have a better worst case performance and overall performance.

1.4 Outline

The remainder of this thesis is arranged as follows:

- **Chapter 2** provides a detailed introduction to the background knowledge, the involved optics notions and theories.
- **Chapter 3** introduces the deep learning-based model as the baseline method.
- **Chapter 4** describes the proposed Bayesian Optimization framework.
- **Chapter 5** describes the dataset, including the data augmentation method, pre- and post-processing of the data.
- **Chapter 6** describes the experiments and analysis for the baseline method and BO method.
- **Chapter 7** finally concludes the thesis and discusses further research directions.

Chapter 2

Background for Optics

In this chapter, the relevant physics and optics background, and the definition of crucial terminologies in optical lens design will be given. These are necessary for formalizing the problem, and to distinguish the input, output, and define the KPIs that will be used in optical lens design.

2.1 Basics of Geometrical Optics

In this section, the basic concepts and theories in geometrical optics will be introduced. It is aimed to establish the basis for the formalization of this problem, connecting readers from a high school-level knowledge of physics to only the necessary notions in optical lens design. Therefore, the derivation of some formulas which involves intensive optics knowledge but not the focus of this work, will be omitted. They will be offered in the appendix and reference for readers' interests.

2.1.1 Definition and Assumptions for Geometrical Optics

In this project, the treatment of a lens system will be based on the geometrical optics, also named ray optics. It is a model to describe how light is propagated, refracted, and reflected, and the formation of images, embodied with the concept of rays [18]. There are several basic postulates for geometrical optics [18]:

- Light propagates in straight lines in a homogeneous medium, such that can be embodied with the concept of rays.
- The angle of incidence equals the angle of reflection. This is called the law of reflection.
- The ratio of the sines of the angles of incidence and refraction, at the interface of two dissimilar mediums, is a constant which depends only on the mediums. This is known as the famous Snell's law, which will be formally introduced in section 2.1.2.

2.1.2 Important Theorems in Geometrical Optics

A basic theorem to be used in computing the direction and position of the ray after refracting at each surface is the Snell's law [19], which is stated as:

$$n \sin(i) = n' \sin(i') \tag{2.1}$$

with n and n' being the refractive index of the first and second medium, which is defined by:

$$n = \frac{c}{v_m} \tag{2.2}$$

where c is the velocity of light in vacuum, and v_m is the velocity of light in the medium. The angles i and i' are the angles of incidence and refraction.

Another important theorem, which will be involved in deriving the optical performance of the system, is the theorem of Malus and Dupin. In order to arrive at this theorem, two basic optical notions need to be firstly introduced. The first one is the **optical path length**. It is defined as the product of the distance of the physical path a ray travels in the medium and the refractive index of that medium. The second notion is the **wavefront**. For rays emitting from the same point on the object, the wavefront is a surface composed of points with the same optical path lengths from that source. Figure 2.1 shows an illustration of these two notions.

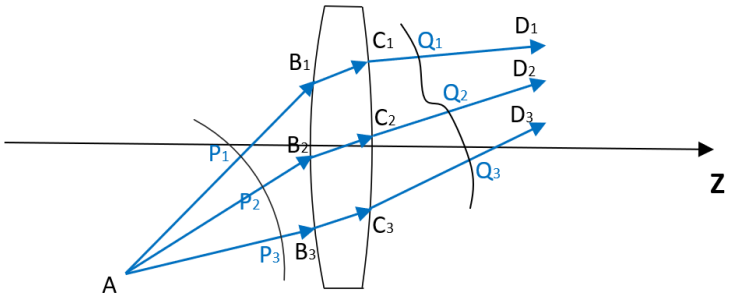


Figure 2.1: Illustration of the optical path and the wavefront.

In Figure 2.1, there are three optical paths $AB_1C_1D_1$, $AB_2C_2D_2$ and $AB_3C_3D_3$ from the same object point A. For example, for optical path $AB_1C_1D_1$, its length is computed as the sum of all segments:

$$[AB_1C_1D_1] = \sum n_i d_i \tag{2.3}$$

In the ray segment AP_1, AP_2, and AP_3, since they are in the same medium, it is clear that the sphere locus centered at A, where P_1, P_2, P_3 lie on, is a wavefront. The wavefront of the locus of Q_1, Q_2, Q_3 is generally not on a sphere because of the refraction in the glass lens leads to different lengths of optical paths B_1C_1, B_2C_2, and B_3C_3. The theorem of Malus and Dupin states that rays are the normal of the geometrical wavefronts [3].

2.1.3 Representation of a Lens System

A lens system is composed of multiple lens elements, including the glass lenses and stops. The necessary variables for describing these components will be introduced in this section.

The Coordinate System

To geometrically represent an optical lens system, firstly the coordinate system and some notations should be defined. Optical design is commonly performed in a symmetrical system, which means the system is symmetric with respect to an axis of revolution. This axis is called

the optical axis, which passes the center of all elements in the lens. The system is therefore called a symmetric, centered system. It is the type of system most frequently met in optical lens design [3]. Figure 2.2 shows an illustration of such a system. The convention is to use the right-hand Cartesian coordinate system, with Z-axis being the optical axis which passes the center of each lens, and Y-axis in the plane of the paper and is orthogonal to Z-axis, and X-axis orthogonal to the other two axes, pointing into the plane of the paper . The rays are defined as traveling from left to right in Z-direction.

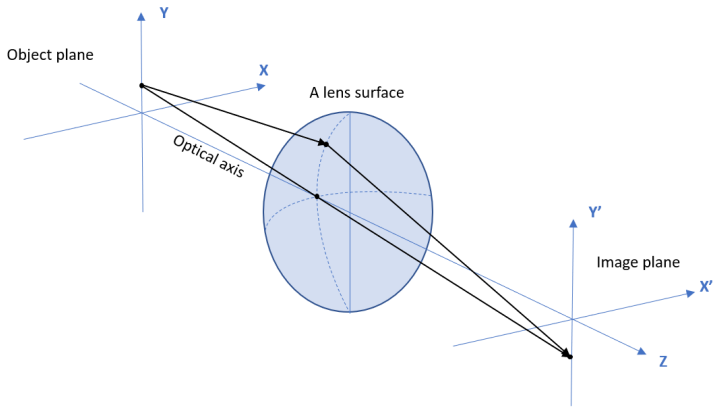


Figure 2.2: The coordinate system for a lens.

Assume the object to be imaged by the lens system is placed in front of the lens system. The object plane is the plane this object is located on, and contains object points that are emitting rays. The image plane locates at the end side of the lenses, which is the plane an image is formed on. This means the optical system is in between the object plane and the image plane. Points on this plane formed by rays from the object points are called the image points. In Figure 2.2, two rays emitting from the same object point in the object plane, travel along the optical axis through a lens, and meet at the same point in the image plane. The result that they meet at the same image point, is called ideal image formation. It in fact only exists in the paraxial zone, which will be introduced in section 2.2.1.

Lenses and Surfaces

Lenses can now be expressed in the defined coordinate system. Figure 2.3 shows an example of a lens with 2 lens elements, plotted in the Y-Z plane, which is also called the **meridian plane**.

As mentioned in Chapter 1, a lens system (or simply lens) is composed of multiple glass lenses. Each of these lenses has a thickness, therefore splits the space into three parts with its two surfaces. Since the refraction happens at the change of the medium, it is a common method in optical design to use the surface representation for the elements. The lens system in Figure 2.3 is composed of 2 lens elements with 4 surfaces. Each surface is defined with 4 parameters: the **curvature** c, **thickness** t, **refractive index** n, and **Abbe number** v.

The curvature c describes the shape of the surface, which is the reciprocal of the radius R of a reference sphere centered at an on-axis point:

$$c = \frac{1}{R} \tag{2.4}$$

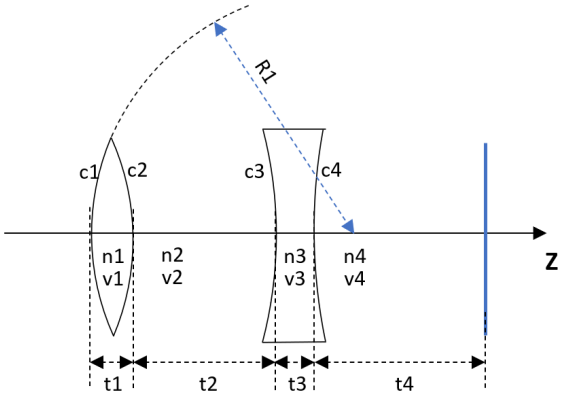


Figure 2.3: An example lens system with the surface representation.

The thickness t is the distance from the center of this surface to the center of next surface. The refractive index n is the ratio of light's speed in vacuum to its speed in the medium behind this surface, as has been defined in Equation 2.2. Since the velocity of light with different wavelengths in are different in a certain medium (i.e., the dispersion in optics), the refractive index of a medium hence depends on the wavelength of light. The Abbe number v of a medium reflects this change in refractive index with the wavelength, and is defined as:

$$v = \frac{n_D - 1}{n_F - n_C} \tag{2.5}$$

where n_C, n_D and n_F are the refractive indices of that medium at the wavelengths of the C line (Hydrogen red light), D line (Helium yellow light), and F line (Hydrogen blue light) in the Fraunhofer lines [20].

Stops

We have seen the glass lenses in an optical system, and until now there is no restriction put on the diameter of the lenses and the size of the object. However, in real lens systems, the rays that are allowed to pass through the system are limited by the diameter of the lens surfaces, and some internal diaphragms. For example, Figure 2.4 shows a case when the rays emitting from the object point Q is limited by the diameter of the lens, and only the rays within the shaded area can reach the image plane. These kinds of limitations on the rays that are allowed to pass through the system are imposed by optical elements called **stops**. They exist in the form of physical iris diaphragms, with a centered hole which allows the transmission of rays; or the rim of a lens, where the rays out of the rim of the lens will miss this lens surface and be blocked by the system.

There are different types of stops in a lens system. An **aperture stop** limits the maximum amount of light from an object point that can pass the optical system. The lens shown in Figure 2.4 is an aperture stop. For the convenience of analysis, the diameter of its center hole is not directly defined, and the convention is to define the diameter of the entrance pupil, which is the image of the aperture stop formed by the lenses in front it when the viewer look from the object plane, as is shown in Figure 2.5. This diameter is called the **Entrance Pupil Diameter (Enpd)**. The position of the entrance pupil is flexible. Sometimes it can be a physical diaphragm, as in Figure 2.5; in some cases, the surface of a lens can also function

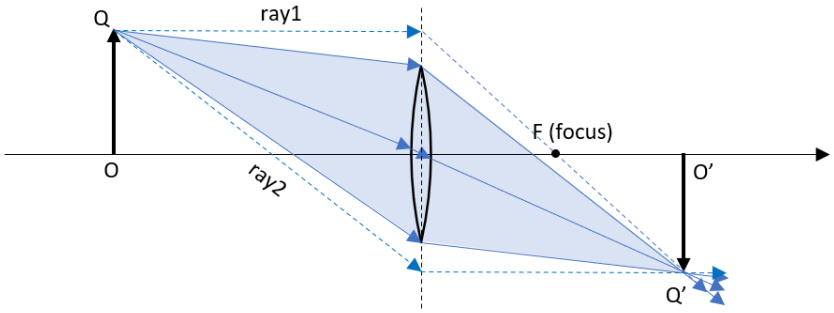


Figure 2.4: Function of an aperture stop. In this case the lens is the aperture stop.

as an aperture stop. When there is no lens in front of the aperture stop, the diameter and position of the entrance pupil are identical with the aperture stop, which is the case in Figure 2.4.

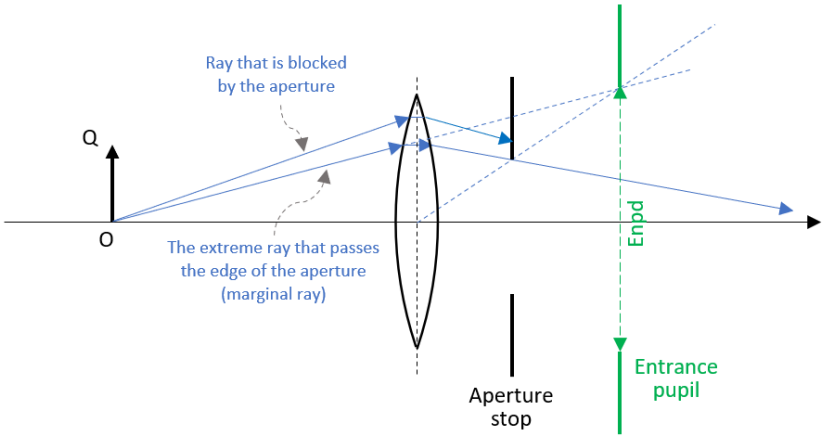


Figure 2.5: Aperture stop, entrance pupil and the entrance pupil diameter.

In addition, when the object is very large, it will not be completely imaged by the lens. A **field stop** is another type of stop which limits the portion of the object to be imaged by the lens system. Figure 2.6 shows a simple model of a single-lens box camera, and we can see the height of the object point affects the amount of light that will be obstructed by the system. The proportion of obstructed light is termed as vignetting. For Q_1 in the object plane, all the light can pass the lens and hit the image plane, therefore it has a vignetting of 0%; at Q_2, vignetting increases to 50%; Q_3 is the extreme case with vignetting=100%, and light from points higher than Q_3 can never hit the image plane. For the field stop, the parameter we care about is the angle of the ray which passes its rim when vignetting=50% (the lower green ray at Q_2). Its angle with the optical axis is called the **Half Field of View (HFoV)** of the system, which is also marked in Figure 2.6.

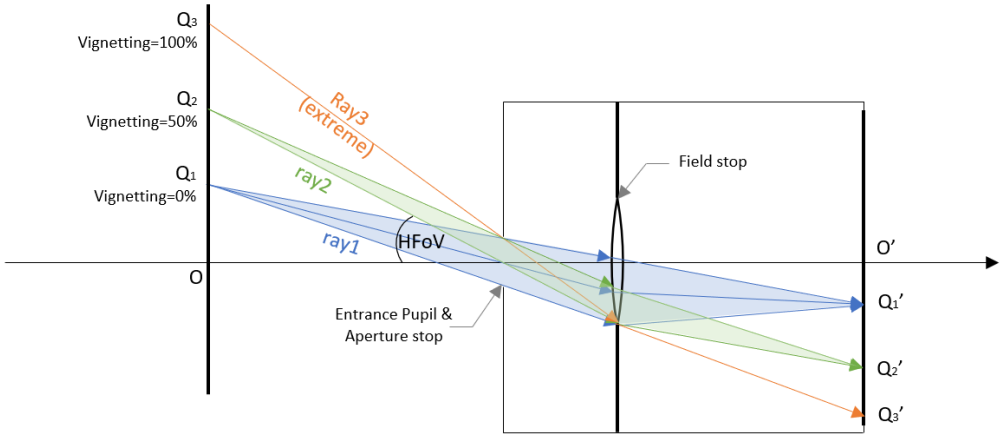


Figure 2.6: Definition and function of the field stop.

2.2 KPIs for Optical Performance

We have seen how a lens system can be represented with geometrical optics. In this section, the performance evaluation of lens systems will be introduced.

Basically, the requirements for a lens system is that it forms an image with a good quality, with no defects in sharpness, shape, and colors, etc. This can be quantified with the aberrations. Section 2.2.1 and 2.2.2 will give an introduction to related notions in the aberration theory. Then the monochromatic and chromatic aberrations that are used in this project will be introduced in section 2.2.3 and section 2.2.4, respectively. In addition, constraints from manufacturing limits, and physically infeasible cases should also be considered. This corresponds to section 2.2.5.

2.2.1 Paraxial Optics

As mentioned in section 2.1.3, the ideal image formation, i.e., the property that rays starting from the same object point will meet at the same image point, is only valid for the so-called **paraxial region**. The paraxial region is a region close enough to the optical axis where only small ray angles and heights are involved, and some approximations can be made, e.g., ray angles are sufficiently small such that $\sin(\theta) \approx \tan(\theta) \approx \theta$.

In most cases, rays from the same object point do not converge at the same point in the image plane, and this attributes to the existence of defects in an image, such as the reduction in sharpness, shape, and colors (recall Figure 1.1). In optics, the formal terminology for this phenomenon is called aberration.

2.2.2 Preliminaries for Aberration Theories

A general concept of aberration can be interpreted as the failure of an optical system to conform to the mode of ideal image formation; in other words, aberrations are the deviation of the system from the ideal imaging in the paraxial region.

Aberrations are related to both the lens' geometry (the parameters c, t, n, v of each surface), and two critical rays that pass through the system: the **marginal ray** and the

chief ray [18]. They are determined by the aperture stop and the field stop, or to say, the Enpd and HFoV of the system. Figure 2.7 shows a simple lens system in the meridian plane (the Y-Z plane) with its elements and the two critical rays. The **marginal ray** is the ray in blue, which passes through the center of the object and the edge of the aperture stop; the **chief ray** in red is the ray which passes through the center of the aperture stop and the edge of the field stop.

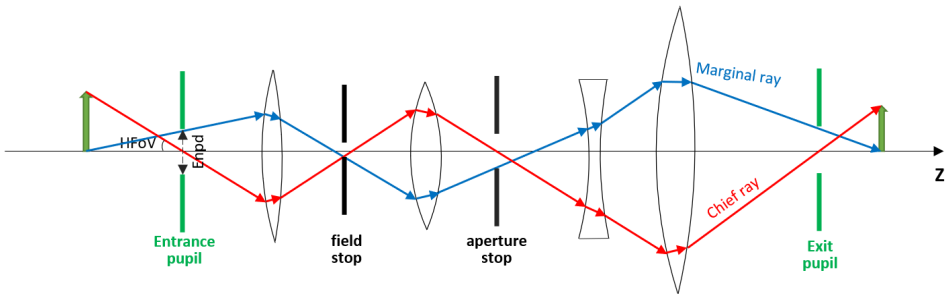


Figure 2.7: Important rays and components in a lens system.

A third property parameter that relates to the computation of aberrations is the **Effective Focal Length (EFL)** of the system. It is a measure of the ability of a lens to converge or diverge light. In paraxial optics, parallel rays passing through a single convex lens will be brought to the same point; and for concave lens, a point source should be placed in front of the lens to form parallel rays after the lens. This point is called the focal point (or focus), and the distance from the center of the lens to the focal point is called the focal length. The shorter this length, the stronger the ability of the lens to diverge or converge light. For a lens system with multiple surfaces, the EFL is measured from the center of a hypothetical plane where the refraction of incident rays happens, to the focal point. This plane can be regarded as a virtual single lens equivalent of the system, as illustrated in Figure 2.8. The EFL is therefore the distance from the center of the virtual single lens equivalent to the focal point of the system.

In optical lens design, Enpd, HFoV and EFL are often given as the desired property parameters of the system, and the mission can be concluded as to determine the lens parameters which produce an image with good quality under the given property parameters. In this project, the image quality is measured by the aberrations.

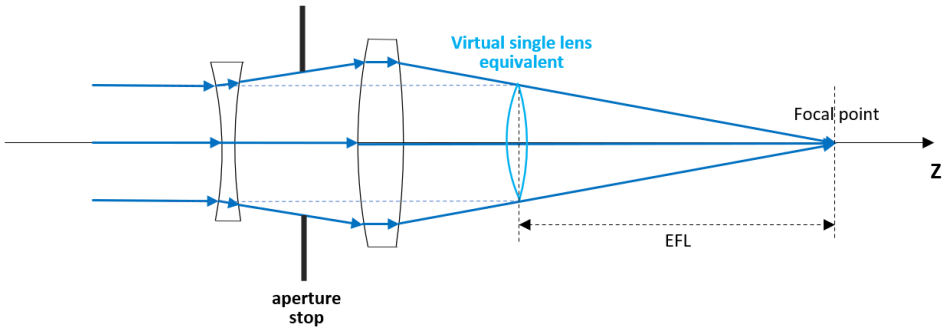


Figure 2.8: Illustration of the EFL and the virtual single lens equivalent

2.2.3 Monochromatic Aberrations

In this section, the monochromatic aberrations will be introduced. They are the type of aberration that are computed only at a certain wavelength.

In real lenses, outside the paraxial region, the ideal imaging is not satisfied. This means rays starting from the same object point will not converge at the same point in the image space. Figure 2.9 shows an illustration of this situation.

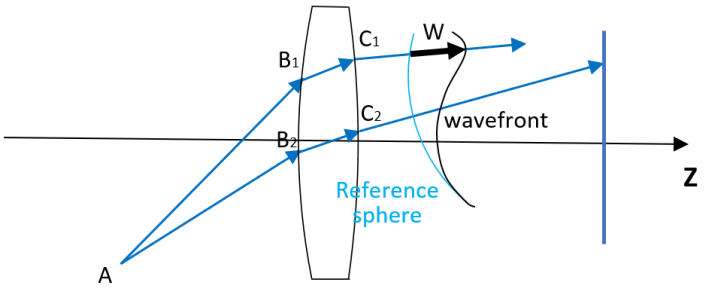


Figure 2.9: Definition of the wavefront aberration.

This deviation from the ideal imaging is termed as the wavefront aberration, and can be measured in the **optical path difference (OPD)**. The aberrated wavefront is expressed with respect to a reference sphere, with its center O' as the paraxial image point.

The wavefront aberration can be expanded into multiple terms with the series expansion [2], with each of them corresponding to a different type of aberration. Among them, the most widely used aberrations are the five Seidel aberrations. They are widely used not only because the computation of them only requires to trace two rays through the system, but also the correction of the Seidel aberrations is a necessity for minimizing the effect of other higher-order aberrations [2] in the total wavefront aberration. In addition, all of them can be expressed in a form of the contributions of each surface, which provides optical designers insights to locate the weak points in the system.

The derivation of the Seidel aberrations can be found in many classical optics books, e.g. [2] and [3]. Here a simple version of the derivation of the first term, S_1, is included in Appendix A for readers' interests. However, the derivation for other terms requires computation for off-axis points (points that do not locate on the optical axis), and is significantly more tedious and involves more optics preliminaries. Since the deviation is not the focus of this project, we will directly borrow as conclusion their forms [3]:

Spherical aberration:

$$S_1 = \sum -A^2 h \delta(\frac{u}{n}) \tag{2.6}$$

Coma:

$$S_2 = \sum -A\overline{A} h \delta(\frac{u}{n}) \tag{2.7}$$

Astigmatism:

$$S_3 = \sum -\overline{A}^2 h \delta(\frac{u}{n}) \tag{2.8}$$

Field curvature:

$$S_4 = \sum -H^2 c \delta(\frac{1}{n}) \tag{2.9}$$

Distortion:

$$S_5 = \sum \frac{\overline{A}}{A}(b_3 + b_4) \tag{2.10}$$

with $A = n \cdot i$ is the product of the refractive index n with the incidence angle i of the marginal ray, which is also known as the refraction invariant. The notations with a bar over them denote the same quantities of the chief ray. The incidence angles of the marginal ray and chief ray can be computed as $i = hc + u$ and $\overline{i} = \overline{h}c + \overline{u}$ (derivation can be found in [2]), where h is the marginal ray's height to the optical axis at a surface (the distance from the intersection of the ray at the surface to the optical axis); u is the angle between the incidence marginal ray and the optical axis. The other two notations with a bar $\overline{u}$ and $\overline{h}$ denote the same quantity for the chief ray. $H = n(\overline{u}h - u\overline{h})$, which is called the Lagrange invariant. The δ means the difference of the quantity after and in front of the surface. The Σ in these coefficients means to sum over all the surfaces. Among these variables involved, it is clear that the only unknown variables are the ray angles u and $\overline{u}$, and the ray heights h and $\overline{h}$ of the marginal ray and the chief ray, after refracted by each surface. These variables are computed through the process called ray tracing, which will be introduced later in 2.3.

The total wavefront aberration is computed as [2]:

$$W = \frac{1}{8}S_1 + \frac{1}{2}S_2 + \frac{1}{2}S_3 + \frac{1}{4}(S_3 + S_4) + \frac{1}{2}S_5 \tag{2.11}$$

The weights of the terms are derived with each term, and are fixed.

There is another equivalent expression of the total wavefront aberration, in which the derivation is using a different method and another set of notions, but has a close final form which can be linked with the Seidel aberrations. This is useful for our further analysis, so it is also presented here[3]:

$$\begin{aligned} W &= W\left(r^2, \eta^2, r\eta\cos\varphi\right) \\ &= b_1 r^4 \quad \text{Spherical aberration} \\ &+ b_2 \eta r^3 \cos\varphi \quad \text{Coma} \\ &+ b_3 \eta^2 r^2 \cos^2\varphi \quad \text{Astigmatism} \\ &+ b_4 \eta^2 r^2 \quad \text{Field curvature} \\ &+ b_5 \eta^3 r \cos\varphi \quad \text{Distortion} \\ &+ \text{etc.} \end{aligned} \tag{2.12}$$

with r being the segment of the ray's intersection at the image plane to the center of the image plane (please refer to Figure 2.10 for an illustration); η is the segment of the ray's intersection at the entrance pupil's plane to the center of this plane; φ is the included angle of these two segments; b_1 to b_5 are five coefficients.

2.2.4 Chromatic Aberrations

The Seidel aberrations, are by definition monochromatic aberrations which are only corrected for light with a certain wavelength. However, these aberrations depend on the refractive index of the material of the medium, and the refractive index is a function of the wavelength of the light. This chromatic variation, called dispersion, is reflected in the Abbe number of the material. The chromatic aberrations are the type of aberration to quantify this variation.

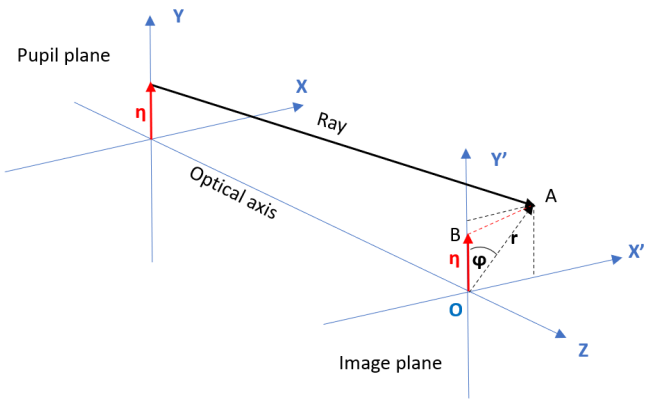


Figure 2.10: Geometrical illustration of a ray traveling through the pupil plane and hit the image plane.

In our project, two chromatic aberrations, the axial color aberration and the lateral color aberration, are also included in the optical performance KPIs:
Axial color:

$$C_1 = \sum Ah\delta(\frac{\delta n}{n}) \tag{2.13}$$

Lateral color:

$$C_2 = \sum \overline{A}h\delta(\frac{\delta n}{n}) \tag{2.14}$$

When computing the total chromatic aberration, the contributions of the axial and lateral color are both 0.5 [3]:

$$C = \frac{1}{2}C_1 + \frac{1}{2}C_2 \tag{2.15}$$

A geometrical illustration of these two chromatic aberrations is shown in Figure 2.11. The axial color aberration is essentially the change of the on-axis intersection length with wavelength. And the lateral color aberration is the transverse magnification with wavelength.

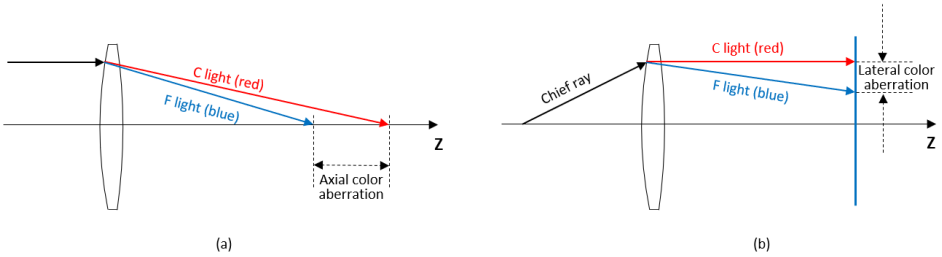


Figure 2.11: Definition of the chromatic aberrations. (a) Axial color aberration. (b) Lateral color aberration.

In this project, the designing of achromatic aplanats are investigated. They are lenses that has no Spherical aberration and no Coma aberration, which are the S_1 and S_2 in Equation 2.11, and is corrected for the wavelength between the C line (red light) and F line (blue light) in the Fraunhofer lines [20]. Therefore, the sum of the first and second terms in Equation 2.11, and the two color aberrations, will be used as our optical performance KPI for image quality.

In the ray tracing for computing the Seidel aberrations, the yellow helium D line in between the C and F lines with a wavelength of 587.6nm is used. Additionally, by pushing these two color aberrations to zero, the system is ensured to have equal axial and lateral magnifications for lights between the C line and F line, in other words, no chromatic aberrations. Such simple lens modules can be the building blocks for more complex systems, therefore it is very important to have these cornerstones with a satisfying optical performance.

To give readers a direct view of the five Seidel aberrations and the two chromatic aberrations, an illustration of them are shown in Figure 2.12 and Figure 2.13.

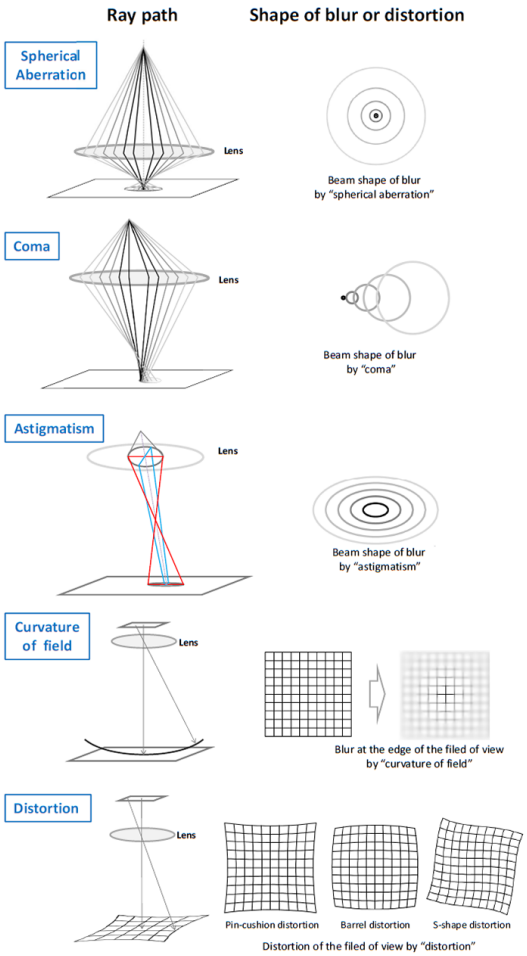


Figure 2.12: An illustration of the five Seidel aberrations from [21].

2.2.5 Physical Constraints and Infeasible Cases

Apart from minimizing the aberrations of the lens system, there are cases when not all the rays can successfully reach the final imaging plane, and when the designed elements are unrealistic (far away from real lenses), which will cause other problems in manufacturing or costs. Therefore, these cases should be considered: two adjacent surfaces overlapping with each other, total internal reflection (TIR), and lenses with undesired aspect ratios in their shapes.

Figure 2.13: An illustration of the two chromatic aberrations from [22].

- **Overlapping of surfaces** When the distance or the thickness between two adjacent surfaces is too small or even negative, two surfaces might overlap with each other, which is impossible in design. Therefore a penalty term for the range of thickness is devised to impose this constraint on the distance between two consecutive surfaces.

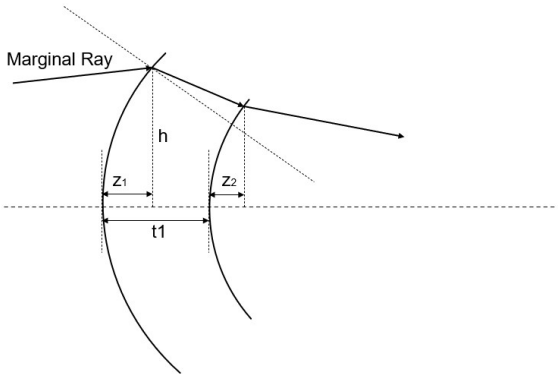


Figure 2.14: The Sagitta values and the space of two consecutive surfaces.

For two consecutive spherical surfaces, as is shown in figure 2.14, the value of the axial displacement of their edge z can be computed as [23]:

$$z = \frac{c \cdot h^2}{1 + \sqrt{1 - h^2 c^2}} \tag{2.16}$$

where c is the curvature of the surface, and h is the height of the marginal ray at the surface, i.e., the distance of its intersection with the surface to the optical axis. This quantity z is called Saggita, which gives a measure of the axial displacement of a curved surface, as is plotted in figure 2.14. Additionally, we would like the space between two surfaces to be greater than a threshold z_{range} for the ease of manufacturing, so the KPI can be set as the accumulated negative value of:

$$\delta z = z_{i+1} + t_i - z_i - z_{range} \tag{2.17}$$

over all the surfaces:

$$\mathcal{L}_{OP} = \sum_{all\ surfs} \delta z^2 \tag{2.18}$$

- **Total Internal Reflection (TIR)** When a ray travels from a medium with a higher refractive index to a medium with a lower refractive index, the ray could be completely reflected backwards and therefore never reaches the next surface, as r_2' in Figure 2.15. This is known as the total internal reflection (TIR):

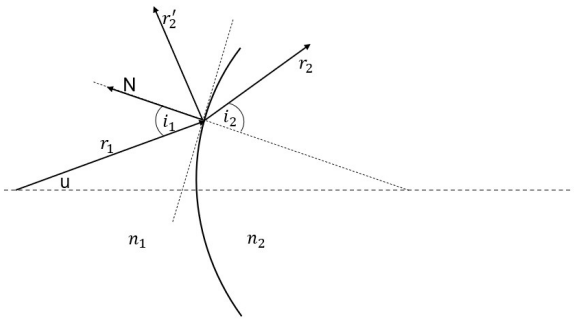


Figure 2.15: An example of ray failure, when total internal reflection occurs.

When this happens in a lens system, the ray can never hit the image plane. To distinguish this kind of ray tracing failure, the product $n_1 \sin i_1$ in Snell's law of refraction is computed over all the surfaces. Since the refractive index of air is 1, $n_2 \sin i_2$ can never be greater than 1, therefore the product $n_1 \sin i_1$ should not be greater than 1. A KPI to be minimized can be computed as the sum of this quantity's difference with 1, when it is equal to or greater than 1. Such a value will lead to no solution to the refraction angle i_2, which is the case of TIR.

$$\begin{aligned} f &= n_1 * \sin(\arccos(N \cdot u)) \\ \mathcal{L}_{TIR} &= \sum_{all\ surfs} max(f - 1, 0)^2 \end{aligned} \tag{2.19}$$

- **Aspect Ratio of Glass Surfaces** As a matter of fact, glass lenses that are either too thin or too thick are not desired. A lens that is too thin will be brittle and adds to the difficulty and cost in manufacturing; on the other hand, light will attenuate in traveling through very thick lenses. Therefore, a penalty is imposed on the aspect ratio of the lens' shape, which is computed as the ratio of its diameter d to its thickness t:

$$a_r = \frac{d}{t} = \frac{2h}{t} \tag{2.20}$$

where d takes the maximum of that at the front and back side of the element. An illustration of its computation is shown in Figure 2.16. The half of d, h_3, can be computed as the sum of the absolute values of the Y coordinates of the marginal ray and chief ray [24]:

$$h_3 = h_1 + h_2 \tag{2.21}$$

In manufacturing, the recommended limit for the aspect ratio is between 1∼30 [25]. Therefore, the penalty for an undesired out-of-bound aspect ratio for glass lenses is computed as:

$$\mathcal{L}_{AP} = \sum_{all\ surfs} (-\min(|\frac{d}{t}| - 1, 0) + \max(|\frac{d}{t}| - 30, 0)) \tag{2.22}$$

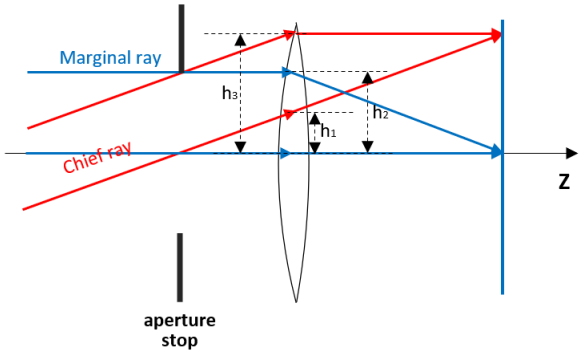


Figure 2.16: The aspect ratio of a glass lens.

2.3 Ray Tracing

Implementation-wise, for computing the Seidel aberrations and the chromatic aberrations, four variables are needed to be computed at all surfaces: the angle u and ray height h of the marginal ray, and the angle $\overline{u}$ and ray height $\overline{h}$ of the chief ray. This can be observed from Equation 2.6 to 2.10, and Equation 2.13 to 2.14. Besides, they are also the necessities to compute the penalty terms for our devised three infeasible cases. This process of iteratively computing the angles and positions of a ray through all the surfaces, is called ray tracing.

Ray tracing can be done with commercial optical designing software such as OpticStudio [6] and CodeV [7], which are rather convenient tools for optical designers equipped with their knowledge in optics. But it is not a good option for data scientists to carry on the model-oriented optimization. Therefore, in this work, a Python ray tracer is implemented to perform the ray tracing process and compute the aberrations and other KPIs.

The Python ray tracer implemented in this work performs the paraxial ray tracing process and compute the aberrations basing on the ray matrix formalism, referred to Noel et al. [26]. In the ray matrix formalism, lens elements are modeled using 2x2 matrices, named the transfer matrix:

$$M = \begin{bmatrix} ma & mb \\ mc & md \end{bmatrix} \tag{2.23}$$

in which ma, mb, mc and md are the elements to be determined with geometrical computation. The ray traveling through the lens system is defined as a vector:

$$\mathbf{r} \equiv \begin{bmatrix} h \\ u \end{bmatrix} \tag{2.24}$$

with h being its distance to the optical axis and u the angle between the ray and the axis.

With Snell's law and geometrical calculations, the transfer matrices of the two elements that are used in modeling our lens designs can be expressed. Here we direct borrow the derived forms of our interests from [27]. For a spherical surface with n_1 and n_2 being the refraction indexes of the mediums before and after it, and c being its curvature, the transfer matrix can be written as:

$$M = \begin{bmatrix} 1 & 0 \\ -\frac{(n_2-n_1)\cdot c}{n_2} & \frac{n_1}{n_2} \end{bmatrix} \tag{2.25}$$

For the propagation of rays in a free space of a distance d, the transfer matrix is:

$$M = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \tag{2.26}$$

With the transfer matrices and vectorized representation of rays, the effects of the lens elements on a ray can be modeled using the left matrix multiplication of the series of surface matrices:

$$\begin{aligned} \mathbf{r}' &= M_i \cdots M_2 M_1 \mathbf{r} \\ &= \mathbf{T_f}(\mathbf{r}, 1, i) \end{aligned} \tag{2.27}$$

which completes the tracing of a ray from its initial state $\mathbf{r}$ at the first surface to its final state $\mathbf{r}'$ after refracted by i curved surfaces. We denote this as a function $\mathbf{T_f}(\mathbf{r}, m, n)$, which stands for the forward tracing of ray $\mathbf{r}$ from surface m to surface n.

In addition, the backward tracing can be also derived to obtain the former states of a ray, at surfaces which are ahead of its current position, denoted as a function $\mathbf{T_b}$:

$$\begin{aligned} \mathbf{r} &= M_1^{-1} M_2^{-1} \cdots M_i^{-1} \mathbf{r}' \\ &= \mathbf{T_b}(\mathbf{r}', i, 1) \end{aligned} \tag{2.28}$$

In order to perform forward or backward ray tracing, we need to firstly setup the initial position and angle of the chief ray and marginal ray. This can be computed from the given Enpd and HFoV. In the design phase, it is a convention to set the object distance to be infinity. In other words, we will work with a parallel beam, i.e., parallel rays with a certain height.

Since the chief ray is defined with the field stop, its angle and position at the first surface can be computed as [3]:

$$\begin{aligned} \overline{u}_1 &= \tan(HFoV) \\ \overline{h}_1 &= -\overline{u} * z_{EP} \end{aligned} \tag{2.29}$$

with z_{EP} the position of the entrance pupil on Z-axis with respect to the center of the first surface. For computing this, we need to trace a ray $\mathbf{r_s}$ with a small angle (which is empirically set as $0.001 rad$) at the center of the aperture stop backwards, and obtain $\mathbf{r'_s}$ at the first surface:

$$\mathbf{r'_s} = \mathbf{T_b}(\mathbf{r_s}, s, 1) \tag{2.30}$$

Then z_{EP} can be computed as:

$$z_{EP} = \frac{h'_s}{\tan(u'_s)} \approx \frac{h'_s}{u'_s} \tag{2.31}$$

with h'_s and u'_s being the position and angle in $\mathbf{r'_s}$.

The setup of the marginal ray at the first surface is much easier. Since the object is placed at infinity, the initial angle of the ray is 0, and the height of the beam is the half of Enpd:

$$\begin{aligned} u_1 &= 0 \\ h_1 &= \frac{Enpd}{2} \end{aligned} \tag{2.32}$$

With their initial positions and angles, and the transfer matrix representation of the refraction of surfaces, the ray heights and angles at each surface can be computed. The Seidel aberrations and chromatic aberrations at each surface, and the KPIs for the other three infeasible cases are therefore obtained.

2.4 Summary

In this Chapter, preliminaries for the involved optics notions and theories are introduced. An optical lens system, composed of a sequence of optical surfaces and stops, can be placed in a right-handed Cartesian system. Each optical surface is defined with its curvature, thickness, refractive index, and the Abbe number. With these parameters for each surface, the propagation paths of rays through the system are determined. Two stops, the aperture stop and field stop, are additional elements to limit the size of rays and the object. And the property of the optical system thus can be described using a triplet (Enpd, HFoV, EFL). The design of such an optical system can be concluded as determining the four parameters for each surface in order to satisfy the desired property triplet.

The KPIs in this problem are built for the image quality and penalty for infeasible cases. In the ideal image formation, which is in the paraxial region, rays starting from the same object point will meet at the same point in the image plane, therefore there is no aberration in the system. But for real lenses, leaving the paraxial region, aberrations appear and should be reduced as an important requirement in optical lens design. The monochromatic Seidel aberrations and the chromatic aberrations are included as the image quality measure. Besides, physically infeasible cases including overlapping of two consecutive surfaces, total internal reflection and an undesired shape of lens are also considered. The computation of these KPIs is done by the process called ray tracing. It is to iteratively compute the angles and positions of a ray at each surface through the system. With these two values, our KPIs can be computed.

Chapter 3

The Baseline Neural Network for Lens Design

In this chapter, the baseline model, which is a deep neural network based framework is introduced. A hybrid training strategy which combines supervised and unsupervised training is adopted, and the loss for each part will be formalized.

3.1 Model Structure

In [15] and [16], a stacked deep neural network of multiple dense layers with small dimensions is introduced to generate a specific type of lens design from the given propery parameters, which is empirically proved to have results with better KPIs than larger and deeper networks. Hence, we decided to use the same structure as Côté et al. in [15] and [16]. As their source code is not given, the code is implemented from scratch, but sticking to the authors' design and what they have illustrated as the best structure in their work. This project is fully implemented in PyTorch [28], a popular Python-based deep learning framework.

Specifically, a stack of multiple DNNs work jointly in parallel, and their prediction results are ensembled by computing the average. Each of them has the same structure (but with different initialization) known as the self-normalizing neural network (SNN). It adopts the scaled exponential linear units (SELUs) proposd by Klambauer et al. [29] as the non-linear activation unit, which is a variant of the Exponential Linear Unit (ELU). Klambauer et al. derived the hyper-parameters in SELU, and mathematically proved for their recommended hyper-parameters and initialization method, the activation will follow a normal distribution $\mathcal{N} \sim (0, 1)$. This is called the self-normalization property, contrary to the external normalization techniques like Batch Norm. Results show that this makes the network converge faster, and the learning more robust and free from gradient vanishing and exploding problems. The structure of each SNN used by Cote et al. [15] is a consequence of 7 dense layer + SELU non-linear activation blocks, as is shown in Figure 3.1.

The outputs of all networks are averaged to yield the final prediction of output variables. Essentially, it resembles training a single SNN with different initialization for multiple times and ensemble their results. The authors found increasing the number of stacks, i.e. the times of training the small network, yields better results than increasing the dimensions and number of layers of the network itself. In our project, the number of stacks is 5 for all the experiments.

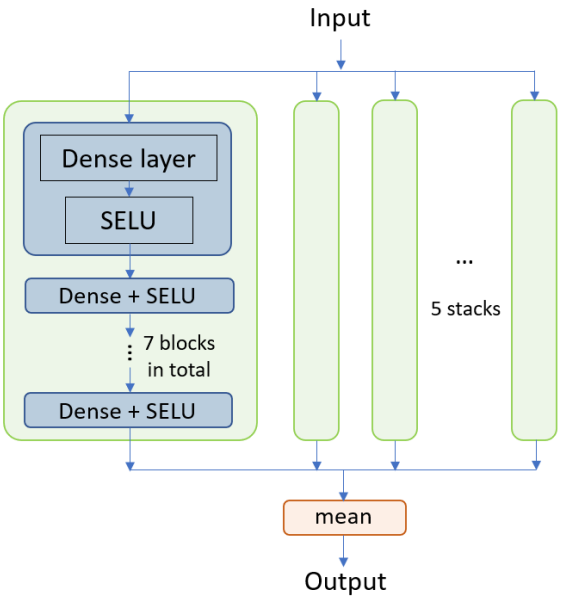


Figure 3.1: Illustration of the network structure.

3.2 Training Process

In order to train the model, one has to consider the data and proper loss functions. The data will be discussed in detail in Chapter 5. For lens design, we have two main goals: the first goal is to learn from known designs, for which we have the design parameters of the input data, i.e. the curvatures, thickness for each surface. This is a regression problem, and a supervised problem. The other goal is to optimize the KPIs for optical performance which have been introduced in Chapter 2, i.e., minimizing the aberrations, and minimizing the penalty on infeasible cases. For this goal, we do not have any labels, other than the ray tracer as a function for evaluating the KPIs. This forms an unsupervised, constrained optimization problem. Therefore, the overall objective contains a supervised term and an unsupervised term. The ray tracer to evaluate the aberrations is also implemented as a differentiable module in PyTorch, and enables us to use the gradient descent method and the auto-differentiation ability of PyTorch to conveniently do the combined optimization.

3.2.1 Supervised Term

In the supervised training, the input data, consisting of our desired property parameters (EFL, Enpd, HFoV) of the lens system, together with the allowed thickness ranges for each surface, are fed into the network. The network generates the parameters that describe the surfaces, i.e., the c, t, n, and v for each surface, which are then used to compute the mean squared error with the true value (the true c, t, n, v of the corresponding surfaces of the reference lens in the training dataset) as the supervised loss:

$$\mathcal{L}_s = \sum_{i=1}^{D_{out}} \frac{1}{D_{out}} (y_i - \hat{y}_i)^2 \tag{3.1}$$

where y_i stands for the ith dimension in the output vector, which can be c, t, n, or v in the

lens' design parameters, depending on its index. For example, the first 4 dimensions in the vector correspond to the c, t, n, and v of the first surface, respectively, and then the 5th dimension is the c for the second surface, etc. A complete description of the data will be given in Chapter 5. D_{out} stands for the dimensionality of the output vector.

3.2.2 Unsupervised Term

In the unsupervised training, we aim at two goals for optimization: the first goal is to ensure the optical performance of the generated lens system. As introduced in Chapter 2, the Seidel aberrations and the chromatic aberrations in the image plane are used as the measure for the image quality. We aim to optimize the total aberrations of certain types, which is a weighted sum of these terms. The second goal is that we expect the generated lens design is feasible. This requires there to be no ray failure in the ray tracing process, which means the marginal ray and the chief ray are traced successfully by reaching the image plane, without missing an optical surface or encountering total internal reflection. In addition, the thickness of lenses should be neither too thick or too thin, since a very thin lens will be brittle thus difficult for manufacturing, and rays will encounter severe attenuation due to very thick lenses. Therefore the unsupervised loss is composed of these two parts: the image quality measure, and penalty for all these infeasible cases.

- **Image Quality Measure** As has been introduced in Chapter 2, in this thesis the Seidel aberrations are used as the measurement for the image quality. For the achromatic aplanats, they should be free from spherical aberration, off-axial coma, and the two chromatic aberrations. Therefore, a weighted sum of these 4 corresponding terms is computed as the image quality loss, written as $\mathcal{L}_q$:

$$\mathcal{L}_q = \frac{1}{8}\sqrt{S_1^2} + \frac{1}{2}\sqrt{S_2^2} + \frac{1}{2}\sqrt{C_1^2} + \frac{1}{2}\sqrt{C_2^2} \tag{3.2}$$

  For simplicity, this weighted sum will be addressed as weighted Seidel hereafter in this thesis.

- **Penalties for Infeasible Cases** The infeasible cases include total internal reflection, overlapping of surfaces, and restriction for the aspect ratio of glass lenses. Recall Equation 2.18, 2.19, and 2.22 in Chapter 2, the penalty for each case writes as:

$$\mathcal{L}_{OP} = \sum_{all\ surfs} \delta z^2$$

$$\mathcal{L}_{TIR} = \sum_{all\ surfs} (f-1)^2$$

$$\mathcal{L}_{AP} = \sum_{all\ surfs} (-\min(|\frac{d}{t}| - 1, 0) + \max(|\frac{d}{t}| - 30, 0))$$

  Ideally, these penalty terms will be 0 when there is no ray tracing failure and all the surfaces are with reasonable thicknesses, and will be positive otherwise.

- **Total Unsupervised Loss** The total unsupervised loss is a weighted sum of the image quality measure and the penalty terms:

$$\mathcal{L}_u = c_q \cdot \mathcal{L}_q + c_{OS} \cdot \mathcal{L}_{OS} + c_{TIR} \cdot \mathcal{L}_{TIR} + c_{AP} \cdot \mathcal{L}_{AP} \tag{3.3}$$

  The c_q, c_{OS}, c_{TIR} and c_{AP} are coefficients reserved to scale these terms. In the experiments, c_q is empirically set to be 10, and the other three all being 1. This is to put a larger weight on the image quality measure, and the weights update of the network will be more affected by this loss.

3.2.3 Hybrid Training

The training of the whole model takes an iterative approach. In the supervised training process, the network will learn from the reference data in the training dataset, while in the unsupervised training process, the network will learn to optimize the optical performance, and also to extrapolate from these known points in the dataset to points (of lens property specifications) where we do not have former reference designs. This comes to taking two batches of samples in each epoch, one for supervised training, that is, performing a regression on the reference data; and the other is for the unsupervised optimization, which is combined with the differentiable ray tracer we implemented. Detailed data generation and process pipeline in the training and testing will be elaborated in Chapter 5. The sum of these two terms are used as the total loss for one training epoch:

$$\mathcal{L}_{total} = \mathcal{L}_u + \lambda_s \mathcal{L}_s \tag{3.4}$$

λ_s is used to balance the two terms. In our experiments, it is set empirically to be 10, with the aim to keep the value of unsupervised loss and supervised loss at a similar scale. An analysis of the respective influences of the supervised and unsupervised terms, by switching on and off each of them, is made in Chapter 6. The classical mini-batch gradient descent is performed. The gradient of this combined loss will be computed and back propagated to the weights of the network. Figure 3.2 concludes the hybrid training framework.

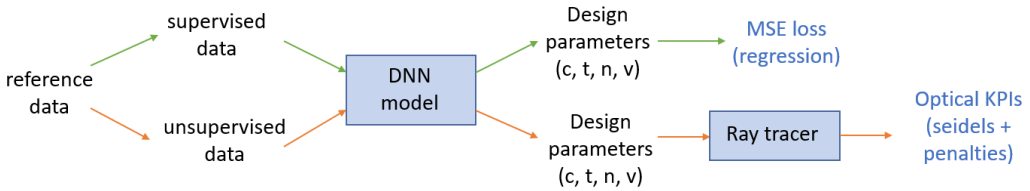


Figure 3.2: The hybrid training framework.

Chapter 4

Bayesian Optimization for Training Samples Selection

This chapter introduces the Bayesian Optimization (BO), and how it will be combined with the DNN to obtain a better global optimization. In section 4.1, the framework and the core concepts in BO will be introduced, including its procedures, the surrogate model, the acquisition function and how the inference is made. In section 4.2, the combined framework of our DNN model in the Bayesian Optimization framework will be introduced.

4.1 Background of Bayesian optimization

In this section, the preliminary knowledge for Bayesian Optimization will be introduced. A general review of the algorithm will be firstly given. Then the core concepts including the surrogate model and the acquisition function will be introduced.

4.1.1 Introduction to Bayesian Optimization

Bayesian Optimization is known for optimizing black-box functions, for which only the input and output are known, and the internal computation process could not be obtained. The problem is to maximize some objective f [30]:

$$\max_{x \in A} f(x) \tag{4.1}$$

where x is the input to the model, and A is the feasible domain.

A statistical model, also known as a surrogate model, is what we used to derive the posterior probability given the prior distribution, based on the Bayes' rule:

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)} \tag{4.2}$$

where w is the parameters of the model, and D is the training data. In this equation, $p(\mathbf{w}|D)$ is the derived posterior; $p(D|\mathbf{w})$ is the likelihood; $p(\mathbf{w})$ is the prior that is put over the model, which represents our belief about the model. Essentially, we are refining the model with information from the newly observed data through Bayesian posterior updating. An acquisition function α is then built with the derived posterior, which incorporates the considerations of exploitation around known maximum, and exploitation to area with a high

uncertainty. The optimization of this acquisition function leads to the next point to be sampled.

A pseudo-code of the classical Bayesian Optimization paradigm can be found in [31], which is shown as below:

Algorithm 1 Bayesian Optimization

1: **for** n=1, 2, ... **do**
2: select new x_{n+1} by optimizing acquisition function α
3: $\mathbf{x_{n+1}} = \arg\max_{\mathbf{x}}$
4: query objective function to obtain $y_{n+1} = f(x_n)$
5: augment data $D_{n+1} = \{D_n, (\mathbf{x_{n+1}}, y_{n+1})\}$
6: update statistical model
7: **end for**

4.1.2 The Gaussian Process

The Gaussian Process (GP) is a widely used surrogate model in Bayesian Optimization. A Gaussian Process is a collection of random variables, and any finite number of variables form a joint Gaussian distribution [32]. A GP can be fully defined with its mean function and variance function. For the real process $f(x)$, a surrogate GP can be defined as:

$$f(x) \sim \mathcal{GP}(m(x), K(x, x')) \tag{4.3}$$

with $m(x)$ and $K(x, x')$ being the mean and covariance function of f:

$$\begin{aligned} m(x) &= \mathbb{E}[f(x)] \\ K(x, x') &= \mathbb{E}[f(x) - m(x)(f(x') - m(x'))] \end{aligned} \tag{4.4}$$

For simplicity, $m(x)$ is often set to be 0. This gives an identical result in the following derivation.

In this surrogate model, the random variable represent the value of $f(x)$ at a certain x. Under a GP, the prior is Gaussian:

$$\mathbf{f}|X, \theta \sim \mathcal{N}(0, \Sigma) \tag{4.5}$$

where $X = \{x_1, x_2, \cdots, x_n\}$ is the training data, and $\mathbf{f} = \{f(x_1), f(x_2), \cdots, f(x_n)\}$ is the collection of values of f. The Σ denotes $K(X, X)$, which is the covariance matrix composed of $K_{i,j} = K(x_i, x_j)$, and θ stand for the hyper-parameters in the model.

When there is noise ϵ in the observation, which is normally assumed to be an i.i.d. Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the observation y can be expressed as:

$$y = f(x) + \epsilon \tag{4.6}$$

The joint distribution of the observations $\mathbf{y}$ and the test output $\mathbf{f}_*$ (with test input X_*) under the prior is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}(0, \begin{bmatrix} \Sigma + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}) \tag{4.7}$$

The predictive distribution then can be derived by conditioning the joint Gaussian prior distribution on the observations, and since $p(\mathbf{y})$, $p(\mathbf{f}_*)$, and $p(\mathbf{y}, \mathbf{f}_*)$ are all Gaussian, $p(\mathbf{f}_*|\mathbf{y})$ is also Gaussian, and can be computed analytically [32]:

$$\mathbf{f}_*|X, \mathbf{y}, X_* \sim \mathcal{N}(\overline{\mathbf{f}_*}, cov(\mathbf{f}_*)) \tag{4.8}$$

where

$$\begin{aligned} \overline{\mathbf{f}_*} &= \mathbb{E}[\mathbf{f}_*|X, \mathbf{y}, X_*] = K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}\mathbf{y} \\ cov(\mathbf{f}_*) &= K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma^2 I]^{-1}K(X, X_*) \end{aligned} \tag{4.9}$$

The covariance function $K(X, X')$ introduced to represent the covariance of the output is also called the kernel, which largely controls the properties of GP. Representative kernels include the radial basis function (RBF) kernel and the Matern kernel, etc, which are discussed in [32], as is shown in Table 4.1.

Table 4.1: Several commonly-used kernel functions. The kernel function is written as either a function of X, X^*, or a function of $r = |X - X^*|$.

covariance function	expression
constant	σ_0^2
linear	$\sum_{d=1}^{D} \sigma_d^2 X_d {X_d}^*$
polynomial	$(X \cdot X^* + \sigma_0^2)^p$
radial basis function	$\exp(-\frac{r^2}{2l^2})$
Matern	$\frac{1}{2^{(\nu-1)}\Gamma(\nu)}(\frac{\sqrt{2\nu}}{l}r)^\nu K_\nu(\frac{\sqrt{2\nu}}{l}r)$
exponential	$\exp(-\frac{r}{l})$

Typically, the RBF kernel and the Matern kernels are the most common choices. The optimization of the hyper-parameters in the kernels (such as the length scale l in the RBF kernel) is to estimate the hyper-parameters that maximize the marginal likelihood $p(\mathbf{y}|X, \theta)$, which is the integral of the product of the likelihood $p(\mathbf{y}|\mathbf{f}, X, \theta)$ and the prior $p(\mathbf{f}|X, \theta)$:

$$\begin{aligned} p(\mathbf{y}|X, \theta) &= \int p(\mathbf{y}|\mathbf{f}, X, \theta)p(\mathbf{f}|X, \theta)d\mathbf{f} \\ &= \int \mathcal{N}(\mathbf{f}, \sigma^2 I)\mathcal{N}(0, \Sigma)d\mathbf{f} \\ &= \mathcal{N}(0, \Sigma + \sigma^2 I) = \mathcal{N}(0, \Sigma_\theta) \end{aligned} \tag{4.10}$$

The log of the marginal likelihood is:

$$\mathcal{L} = ln(p(\mathbf{y}|X, \theta)) = -\frac{1}{2}\mathbf{y}^T\Sigma_\theta^{-1}\mathbf{y} - \frac{1}{2}ln(det\Sigma_\theta) - \frac{n}{2}ln2\pi \tag{4.11}$$

Then by maximizing this log marginal likelihood the estimation for the hyper-parameters can be obtained.

4.1.3 Acquisition Functions

With the above-mentioned ingredients for the surrogate model, the posterior can be estimated. And the next step is to use the acquisition function and decide where to sample next, and iteratively update the posterior using our new observations. When choosing the acquisition function, there is a trade-off between exploitation and exploration: we want to choose the places near a high mean, which are more likely to give the maximum value; we are also interested in the places with high variance, which indicates a higher uncertainty and potential to give high values. Typical acquisition functions including the Thompson Sampling (TS) [33], Expected Improvement [34], Upper Confidence Bound [35], etc.

The Thompson Sampling is a classical heuristic sampling method in the multi-armed bandits problem. The main idea is to choose the action which maximizes the expected reward. Applied as the acquisition function in BO, it will be sampling from the GP and take the maximum [36]:

$$\alpha_{TS}(x; D_n) = p(f^*|x) \tag{4.12}$$

The Expected Improvement is to measure the amount of improvement upon the former optimal τ. The improvement function is defined as:

$$I(x, f, \theta) = (f - \tau)\mathbb{I}(f > \tau) \tag{4.13}$$

The indicator function $\mathbb{I}(f > \tau)$ equals 1 if $f > \tau$, otherwise will be 0. Due to the posterior is a Gaussian, this acquisition function can be derived in an analytical form [34]:

$$\begin{aligned} \alpha_{EI}(x; D_n) &= \mathbb{E}[I(x, f, \theta)] \\ &= (\mu_n(x) - \tau)\Phi(Z) + \sigma_n(x)\phi(Z) \end{aligned} \tag{4.14}$$

with

$$Z = \begin{cases} \frac{\mu_n(x) - \tau - \xi}{\sigma_n(x)} & \text{if } \sigma_n(x) > 0 \\ 0 & \text{if } \sigma_n(x) = 0 \end{cases} \tag{4.15}$$

In Equation 4.14, ϕ is the probability distribution of Z, and Φ is the cumulative density function. It puts more effort into exploration when ξ is larger, by lowering the importance of the predictive mean. A recommended default value of this parameter is 0.01.

Another straightforward, and also widely used acquisition function, the Upper Confidence Bound, writes as follows:

$$\alpha_{UCB}(x; D_n) = \mu_n(x) + \beta_n \sigma_n(x) \tag{4.16}$$

The general idea is to exploit around the predictive maximum point, and β_n is the parameter to control the extent of exploration to points with higher variance.

4.2 The Combined Framework

The main idea of combining Bayesian Optimization and DNN is to use Bayesian Optimization to guide the learning of the neural network. The image quality KPIs, quantified with aberrations, are functions of the network inputs (EFL, Enpd, HFoV) and the lens design parameters (c, t, n, v). Prior knowledge from optical designers says that when the Enpd is high, the first term in the Seidel aberrations is difficult to be optimized. This can be partially

observed from the original series expansion form of Seidel aberrations 2.12: the first term $b_1 r^4$ contains the 4th-order of r, which is highly related to Enpd. But since this effect is coupled with other variables, there is not a determinant function to describe this relation.

In other words, some input triplets (EFL, Enpd, HFoV) are harder samples for the network to optimize the KPIs, but the exact function of this relationship is not known to us. Therefore, we aim to select those harder training samples, and intentionally train the network on those samples in order to have better KPIs on those harder points.

With this motivation, a Gaussian Process is established as the surrogate model from network inputs to the KPIs. But different from the traditional Bayesian Optimization, the black-box function is known to us. It is in fact the network with the ray tracer, and the KPIs can be directly computed from it. Our aim is to use the acquisition function to select those hard samples where the network currently does not behave well, i.e., the generated lens designs are of the highest KPI.

Different from a classical Bayesian Optimization problem, our black-box function will be continuously optimized to have an improving performance, therefore the observed data will also be continuously updated during the BO training process. The surrogate model, which is parameterized using the typical Gaussian Process, will take the input of the network as input, aka the lens property parameter triplets, and outputs the weighted Seidel. As the other three penalty terms are easier to be optimized and will be 0 for most of the time, and also to avoid a coupled behavior at the beginning, we only establish the GP as a single task regression for the weighted Seidel.

In the training of the BO, the GP and DNN will be iteratively optimized. Firstly a batch of random data sampled from the input domain forms the BO training dataset that is fed into the network to train for the first epoch. Then the first batch of observations are made by evaluating these first samples using the trained network, and the GP is updated on this dataset. The next sample(s) is then proposed by optimizing the acquisition function within the bounds of the input variables, and added to the original BO dataset which composes the augmented BO training data. Then, the network is updated on the augmented training data, and we query the network again to obtain the current image quality KPI, i.e., the weighted Seidels, which are the targets of the BO dataset to train the GP. These procedures are repeated until a preset stopping condition is reached, for example, a pre-defined loss threshold or a maximum total training epochs. Algorithm 2 describes the detailed process of the BO training framework.

Algorithm 2 Combined Bayesian Optimization Framework

Train the NN on x_1, compute the Seidel loss s_{11} on these samples from the ray tracer
Obtain the initial BO dataset $D_1 = (x_1, s_{11})$
for n=1, 2, ... (each BO loop) **do**
    Update the GP on D_n
    Select new x_{n+1} by optimizing the acquisition function α:
        $x_{n+1} = \arg\max_{\mathbf{x}} \alpha(x; D_n)$
    Train the NN on $(x_{1:n}, x_{n+1})$
    Query the NN and the ray tracer to obtain $s_{n+1,1}, s_{n+1,2}, ..., s_{n+1,n+1}$
    Augment the dataset $D_{n+1} = (x_1, s_{n+1,1}), \ldots, (x_{n+1}, s_{n+1,n+1})$
end for

Chapter 5

Dataset

In this chapter, the dataset and the data processing operations will be introduced. In section 5.1, a brief introduction about the dataset we use, the Zebase dataset, and the type of lens design investigated will be shown. Section 5.2 will introduces how the data is structured, and the necessary data pre- and post-processing. Section 5.3 then gives the project-oriented data augmentation method for the supervised and unsupervised training that are mentioned in Chapter 3.

5.1 The Zebase Dataset

In this project, the lens designs in the Zebase 6 [37] database are used as the training data. It is a library of over 600 canonical lens designs from M. Laikin's book [38]. They are collected from historic patents and papers, which cover a wide range of usage, e.g., microscopes, eyepieces, magnifiers, projection lenses, telescopes. Each lens design consists of different number of surfaces (ranging from 1 to 36) and different types of surfaces (glass, air, mirror, etc.). The layout of these lenses also vary with their applications and optical elements. In some of them, all the elements are symmetrical and are placed coaxial; but there are also other types in which the optical axis is folded. As a summary of this database, they all serve as good starting points with different property parameters (i.e., the EFL, Wfno, HFoV, etc.) of the system, for optical designers to carry on further optimizations and designs.

5.2 Structurize and Processing of the Data

The complex and different representation of the lens designs brought up a challenge in structurizing the data to be a unified form that can be processed by the machine learning model. Most machine learning algorithms can only deal with input features with a fixed length, and there are specific data type requirements for the input features (numerical, categorical, etc.). Therefore, to adapt to our DNN model, we firstly focus on lenses of a specific sequence of surface types, disregarding their application and usage in a broader sense. This is due to the reason that for different types of surface, the number of interest variables will be different. For air surfaces, their refractive index n and Abbe number v are fixed, therefor we only have c and t for air surfaces. In other words, the number of surfaces in the lens, and the surface types, are essentially embedded with each dimension in the input and output.

5.2.1 Representation of the Lens Designs into Vectors

The type of lens design we chose to work with is the air-spaced doublets, which is a sequence of four surfaces GAGA, where G stands for a glass surface, and A stands for an air surface. For a glass surface, 4 parameters are needed for describing the surface: its curvature c, thickness t, the material related variables refractive index n and Abbe number v; for an air surface, since its material is fixed, we will have $n = 1$ and $v = 1e6$. Therefore, only the other two variables curvature c and thickness t are needed to describe an air surface. In addition, as a way to guarantee the generated system has the expected EFL, the curvature c_l and thickness t_l of the last surface are reserved and solved manually (they can be seen as functions of Enpd, EFL, and the ray height and angle at the next to last surface):

$$\begin{aligned} t_l &= -\frac{y_{l-1}}{u_a} \\ c_l &= \frac{n_l * u_a - n_{l-1} * u}{-y(n_l - n_{l-1})} \end{aligned} \tag{5.1}$$

where n_{l-1} and n_l are the refractive indices at the next to last and the last surfaces; u and y are the ray angle and ray height of the marginal ray at the next to last surface; u_a is the exit angle of the lens to ensure the image is formed at the focus, and is computed by:

$$u_a = \frac{-Enpd}{2 * EFL} \tag{5.2}$$

Therefore, for a GAGA lens, the output vector should be:

$$y = [c_1, t_1, n_1, v_1, c_2, t_2, c_3, t_3, n_3, v_3] \tag{5.3}$$

where the subscripts of numbers represent the index of the surface this variable is affiliated to.

The input to the model should contain the specified property parameters of the system, including the EFL, Enpd, and HFoV. In the meantime, as is in [15]and [17], as enriched features, the allowed thickness ranges for each surface are also provided to the model.

5.2.2 Normalization of the Variables

For the normalization of the optical variables in this project, a special method considering the property of the optical system is adopted. Since in lens design, the lens system and the aberrations in general scale with the EFL, therefore we normalize the variables to the unit EFL. Specifically, all the length-related variables are divided by the EFL:

$$\begin{aligned} EFL' &= 1 \\ Enpd' &= \frac{Enpd}{EFL} \\ c' &= \frac{1}{R/EFL} = c * EFL \\ t' &= \frac{t}{EFL} \end{aligned} \tag{5.4}$$

where the notations with a prime are the ones after normalization. Other non-related variables are not touched, such as the angle HFoV and coefficients n, v. But since the computation

of KPIs are based on the true values of the parameters of the system, in other words, the ray tracer is a function of both the lens parameters and the property parameters, so before computing the KPIs with the ray tracer, the variables need to be de-normalized to the system's original EFL. The same normalization and de-normalized is performed for the GP in the BO framework.

5.2.3 De-correlate the Lens Material Variables

For real glass materials, the refractive index n and Abbe number v have an intimate correlation, as is shown in 5.1. So they are de-correlated by projecting to the two principal axes using a Principal Component Analysis (PCA) model trained on a dataset of n and v values of commonly used lens material, named Schott's glass catalog [39]:

$$(g_1, g_2) = PCA(n, v) \tag{5.5}$$

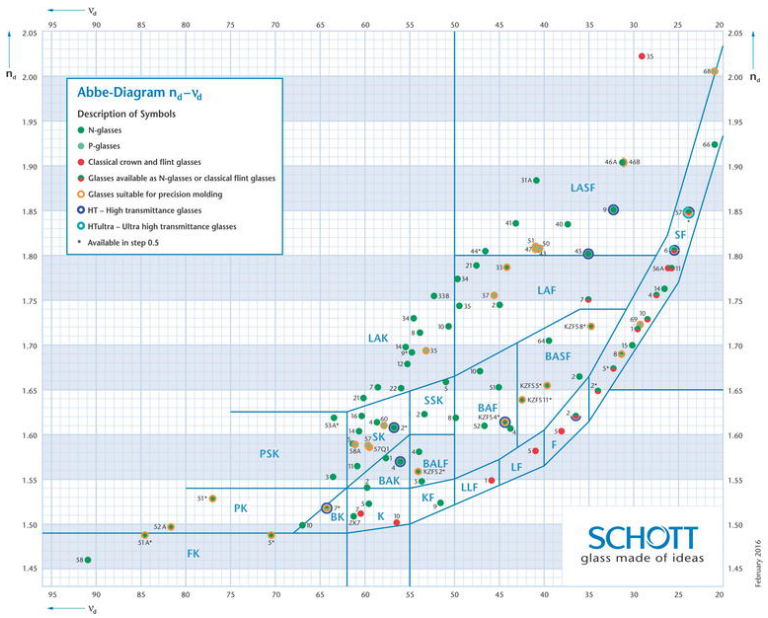


Figure 5.1: The Schott glass catalog.

And due to a material with a high refractive index will be very expensive in manufacturing, especially when above 2.0 (typically, the refractive index of diamond for 587nm light is 2.4), the generated lens material should be forbidden in certain range. Therefore, a *tanh* non-linear transformation is added to the network's raw output of n and v to limit them within the desired range. In this project, the bounds for these two variables are set as $[1.4, 2.0]$ and $[20, 90]$. And a linear mapping will shift the *tanh* output from [-1, 1] to the desired range.

For the output of the network, an inverse PCA will be performed to obtain the true n and v:

$$(n, v) = PCA^{-1}(\tanh(g_1, g_2)) \tag{5.6}$$

5.2.4 Mapping of the Thickness

In order to limit the output thickness within the pre-defined thickness ranges in the input, we follow the operation taken by Cote et al. in [15], which uses a softplus activation to map

the raw thickness into the proper interval $(t_{min}, t_{min} + t_{range})$, using the below equation:

$$t' = t_{min} + softplus(t - t_{min}, \beta) - softplus(t - t_{max}, \beta) \tag{5.7}$$

where t' is the transformed bounded thickness, and t is the raw output thickness of the network. t_{min} and $t_{max} = t_{min} + t_{range}$ are the minimum and maximum allowed thickness values for a surface. These two variables t_{min} and t_{range} are set up for floating the thickness of each surface, similar as in conventional optical design. They will also be given as input to the network. The parameter β is set to be 8 in our experiments.

5.3 Data Augmentation for Learning

To have enough training data for the supervised training and unsupervised training, a strong data augmentation based on the limited dataset is necessary. For the type of design we are working with, the GAGA sequence, there are only 17 available reference designs in the database. Therefore the unsupervised training will be of great importance to the extrapolation to input property specifications where there are no reference designs in the database.

For the thickness ranges, uniform distributions $t_{min} \sim U[a_m, b_m], t_{range} \sim U[a_r, b_r]$ are defined for each of the surfaces as to draw the augmented data. For data used in supervised training, the network aims to learn from those existing reference designs, therefore for each surface the allowed range for floating the thickness of that surface is defined by expanding the true thickness of that surface by ± 0.25 under the unit EFL, as is shown in Table 5.1.

Table 5.1: The lower and upper bounds for thickness ranges.

	t_{min}	t_{range}
a	max(t-1.25, 0)	t+0.25
b	max(t-0.25, 0)	t+1.25

For unsupervised training, the lower and upper bounds for each of the surfaces will be drawn from the global lower and upper bound of that surface in the dataset.

As for EFL, HFoV, and Enpd, in supervised training, they will be fixed as those of the reference designs; in unsupervised training, the strategy is similar to that for the thickness ranges. Each variable will be randomly drawn from a uniform distribution, with its lower and upper bound calculated as the global minimum and maximum of that variable in the dataset, with a $\pm 25\%$ expansion of its original domains. For example, for EFL, this distribution will be:

$$EFL \sim U[EFL_{min} \cdot 0.75, EFL_{max} \cdot 1.25] \tag{5.8}$$

To summarize, the input vector to the model will be a 11-dimensional vector, composed of 3 lens property variables (Enpd, HFoV, EFL), and 8 thickness bounds for the 4 surfaces:

$$(Epnd', HFoV, EFL, t_{1min}, t_{1range}, t_{2min}, t_{2range}, t_{3min}, t_{3range}, t_{4min}, t_{4range}) \tag{5.9}$$

where the number in the subscript for each t indicates the surface index it corresponds to.

During testing, the input lens property specifications (EFL, Enpd, HFoV) will be specified by the user, and the allowed thickness ranges are randomly drawn from the prior distributions and concatenated to the lens property variables as a source of randomness.

The data processing pipeline with the model, from the external input to the final output, can be concluded in Figure 5.2.

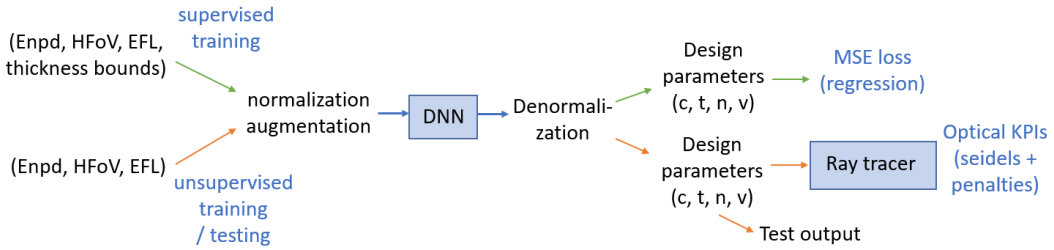


Figure 5.2: The data processing pipeline.

Chapter 6

Experiments

In this chapter, experiments will be described and results will be presented. Comparisons and analysis are firstly done for the baseline framework described in Chapter 3 in section 6.1; in section 6.2, the effectiveness of the proposed Bayesian Optimization framework will be shown.

6.1 The Baseline Framework

In this section, the baseline method with only the DNN will be first tested to show our implemented algorithm's performance. Since some details in Cote et al.'s work [15] are not disclosed, and some improvements are also made, it is necessary to examine the performance of our model, on both the reference dataset and a generated dataset with input specifications across the input space. Then an ablation study will be made to illustrate the effectiveness of the hybrid training framework.

6.1.1 Verification of the Differentiable Ray Tracer Module

In order to train this end-to-end model, the ray tracer is firstly implemented together with the network in PyTorch, as introduced in Chapter 3. This enables the unsupervised optical KPIs to be differentiated and their gradients with respect to the model parameters be back propagated to the network. Therefore, on top of all the experiments, a fundamental step is to verify the correctness and consistency of the implemented PyTorch ray tracer module and the results of the ray tracing in commercial optical design software OpticStudio. For this purpose, a comparison of the Seidel aberrations and chromatic aberrations of the PyTorch ray tracer and those computed by OpticStudio on the reference dataset is made, as is shown in Table 6.1. Since it is verbose to give all the results for each design, only two examples and the averaged relative error on all reference designs is shown here. The detailed results can be found in Appendix B.

It can be seen the differences between them are very small. All the relative difference for the seven aberrations are smaller than 3%, within the tolerance range. The difference is considered to be acceptable because of the accuracy in different programming languages. In Python, the default datatype float has 32 bits where 23 bits are used for the mantissa, therefore only about 7 decimal digits are significant. And since the Seidels are computed in an accumulative way through all the surfaces, the small difference is accumulated as well.

Table 6.1: Comparison of the OS and the PyTorch ray tracer.

		S_1	S_2	S_3	S_4	S_5	C_1	C_2
A_006	OS	0.001630	0.001251	0.003107	0.001205	-0.000107	-0.001126	0.001256
	PyTorch	0.001618	0.001252	0.003107	0.001205	-0.000107	-0.001125	0.001256
A_013	OS	-0.000784	-0.000057	0.000382	0.000277	0.000001	-0.001369	0.000032
	PyTorch	-0.000790	-0.000058	0.000382	0.000277	0.000001	-0.001369	0.000032
Mean (relative)		2.1322%	0.0039%	0.0001%	0.0002%	0.0002%	0.0024%	0.0002%

Another possible reason is some approximations we made when computing the paraxial ray tracing with python (such as $\sin(i) = \tan(i) = i$) might not be totally the same with that in the software. However, it is noted that S_1 is much larger than the other six. This is due to in the computation of S_1 (Equation 2.6), the A is much larger than other terms. It contains the angle of the marginal ray, which is in general much larger that that of the chief ray. Confirmed with our optical designer, we think the implemented PyTorch ray tracer is a usable implementation.

6.1.2 Experiment Settings

The framework and data processing pipeline are implemented fully in PyTorch to take advantage of its automatic differentiation feature. In each epoch, a batch of supervised samples and a batch of unsupervised samples will be generated. The supervised loss, MSE (Equation 3.1), is computed on the supervised batch, and the unsupervised loss, the optical KPIs (Equation 3.3) can be computed on both the supervised samples and unsupervised samples. For simplicity, in the hybrid training, we will only compute the unsupervised loss for it. The total loss is computed according to Equation 3.4 and the total gradients are back propagated to the network parameters. For each batch of samples, the batch size is set as 1024. The learning rate starts from 2e-5 and increases to 2e-3 with a linear warm-up manner. Then it gradually reduced to 2e-5 using a cosine annealing scheduler. The network is trained for 200k epochs and 4% of the total epochs are reserved for the warm-up phase.

In all the experiments, the network structure is fixed as a 5-stack parallel structure with 7 repetitive blocks, and each block contains one fully-connected layer with 32 hidden units, and one SELU activation layer. The output of the last layers are averaged then connected with corresponding non-linear mappings and de-normalization operations to be mapped into the desired ranges.

6.1.3 Performance evaluation for the optical KPIs in the Input Space

In real use cases, a triplet of desired property parameters (EFL, Enpd, HFoV) is given by the users, and the model should give the corresponding design parameters of the lens. Therefore we need to evaluate the model's performance across the input space of (EFL, Enpd, HFoV). The allowed thickness ranges in the input are randomly drawn from the pre-defined ranges and attached to the property triplet as the input to the neural network. For each triplet, 10 groups of thickness ranges are generated and 10 different designs will be given by the network, and the averaged optical KPIs will be computed as the performance on this given property triplet.

Since each of the values is continuous, it is only possible to take discrete points with fixed steps. The upper and lower bound for drawing each variable in the property triplet are set

as the maximum and minimum value of that in the dataset. Figure 6.1 shows the values of the weighted Seidel at a certain EFL value. The color encodes the value of the weighted Seidel. For visibility we have taken the logarithm of the weighted Seidels. A brighter color indicates a bigger value, thus a worse performance at that point. From top to the bottom is the performance profile when EFL=1, 100, and 1000, respectively.

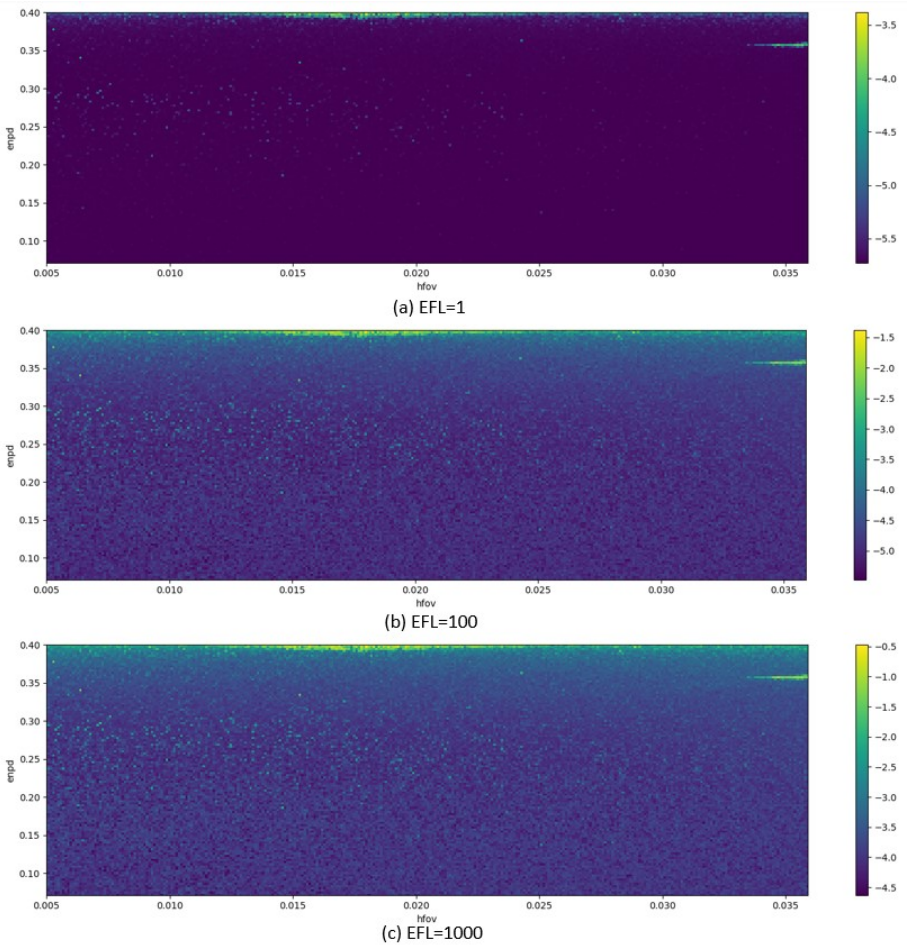


Figure 6.1: Performance profile of the weighted seidel at a certain EFL in the input (HFoV, Enpd) space. (a) EFL=1. (b) EFL=100. (c) EFL=1000.

In general, the network is able to produce lens designs that have a small weighted Seidel at most (HFoV, Enpd) coordinates, except for the top region when the Enpd is large (in other words, when the Wfno of the system is small), and a narrow strip around the top right area of the input space. The first area can be understood from Equation 2.12, since the r is highly related to Enpd and is much larger than other variables, therefore the Seidels are harder to be reduced at a large Enpd. This is also a normal case we will encounter when we buy camera lenses, that a lens with a smaller Wfno (or a high Enpd) is in general more expensive.

The mean of the weighted Seidel throughout the test space is also computed as the overall performance of the model. For the infeasible cases to be penalized, which are the Total Internal Reflection (TIR), Overlapping of surfaces (OP), and an aspect ratio which is smaller than 1 or greater than 30 (AP), a different approach is adopted for evaluation. Whenever the

value of the penalty term is not zero, that lens design is regarded as a failure. The average failure rate for each case can be therefore computed for each EFL value. These statistics are shown in Table 6.2.

Table 6.2: Comparison of the OS and the PyTorch ray tracer.

EFL	Seidel_mean	TIR	OP	AP
1	2.73677e-6	0.074%	100%	0.028%
100	1.00948e-4	0.074%	6.95%	0.028%
1000	9.68709e-4	0.074%	4.09%	0.028%

It can be clearly seen from Table 6.2 that the value of the weighted Seidel scales roughly with the EFL of the system. For all three EFLs, the generated lens design all have a rather small mean weighted Seidel across the input space. The failure rates of the TIR and AP are also very small. This means most of the generated designs are free from TIR, and the aspect ratios of the glass surfaces are within the desired ranges. It should be noted that in the computation of OP, a threshold for the minimum gap is also included, which will become significant when the EFL is very small. This contributes to a high failure rate of OP when EFL=1.

6.1.4 Comparison to the Reference Lens Designs

Comparison of the generated lenses to the reference dataset is also made. In this comparison, the input property triplets will take the values of those values in the reference dataset. For each of the lens design, 100 groups of thickness ranges will be randomly drawn and attached to the triplet as input to the network. Then similar to in the previous experiment, the averaged weighted Seidel of the outputs, and the failure rates of each infeasible cases will be computed, but only for each specific design. Results are shown in Table 6.3.

Table 6.3 shows the weighted Seidel, TIR and OP of the reference data (the ref column) and the generated designs (the gen column) of a specific design. The APs are all 0, therefore to save space they are not shown. For the reference lens, the three values are the exact values of that lens design; for the generated lens, the Seidel is the average of the 100 output designs, and the TIR and OP are the failure rates of the 100 designs. For most property triplets, the generated lens designs have smaller Seidels and most of them are successful designs with no failures. Though in the first three lines, the averaged weighted Seidels of the generated lenses are not better than the reference design, some better ones among the 100 outputs can still be found. In fact, it is not surprising that the network is able to come up with designs with a much smaller weighted Seidel, since the reference designs in the database are not only optimized for the Seidel aberrations. In the conventional lens design process, the optical designers will first have those designs with a very small Seidel as starting points, and then use the optical designing software to do a local optimization with their customized merit functions for different goals. Therefore, it is expected our network should be able to produce lens designs with a small weighted Seidel as those good starting points in the first place.

Table 6.3: Comparison to the reference dataset.

name	EFL	HFoV	Enpd	ref			gen		
				Seidel	TIR	OP	Seidel	TIR	OP
A_006	100	2	25	0.002020	0	0	0.0034254	0	0.12
A_007	100	2	25	0.001520	0	0	0.0040912	0	0.14
A_008	100	2	25	0.001672	0	0	0.0040047	0	0.09
A_010	40	1.5	6.249	0.000121	0	0.11	0.0000102	0	0.14
A_011	2800	0.5	200	0.000664	0	1.14	0.0001347	0	0
A_012	2000	0.5	200	0.000826	0	0	0.0011370	0	0
A_013	100	1	14.29	0.000882	0	0	0.0000139	0	0.03
A_014	100	1	14.28	0.000679	0	0	0.0000130	0	0.1
A_015	100	1	14.28	0.000172	0	0	0.0000112	0	0.03
A_017	100	1	33.33	0.004407	0	0	0.0001114	0	0.1
A_018	100	1	35.71	0.004306	0	0	0.0013946	0	0.07
A_019	100	1	35.71	0.006999	0.023	0	0.0011093	0.02	0.09
A_020	100	1	13.99	0.0002765	0	0	0.0000128	0	0.09
A_024	1000	0.5	125	0.0022022	0	0	0.0001111	0	0
A_025	1000	0.5	125	0.0014837	0	0	0.0000961	0	0
A_028	100	1	20	0.0002764	0	0	0.0000300	0	0.05
A_029	100	1	20	0.008974	0.016	0	0.0000415	0	0.1

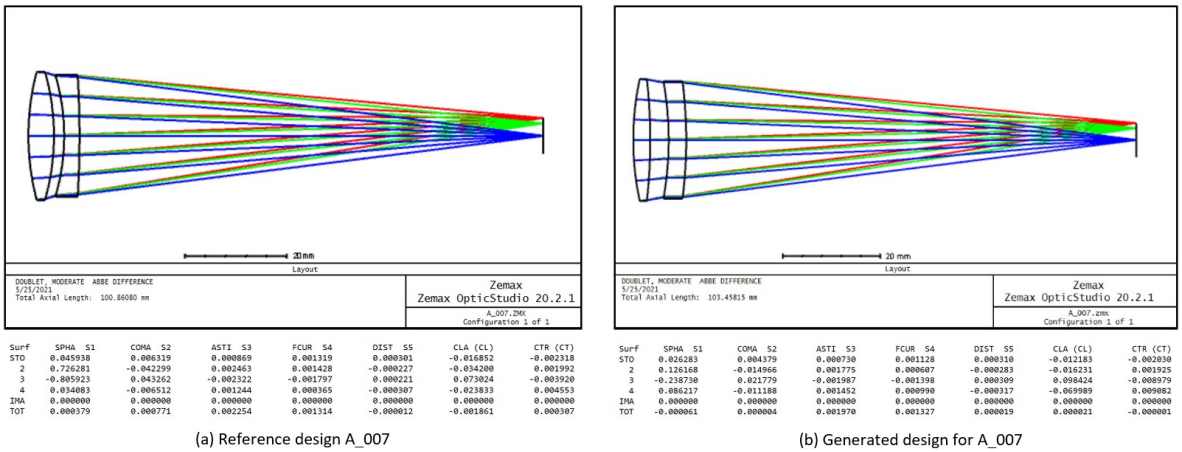
6.1.5 Visualization and Simulation Analysis of Generated Lens Designs

To illustrate the generated design in a more direct way, we also exported the output lens parameters into OpticStudio to plot their layouts and analyze them using the tool. Two generated lens designs are plotted and their corresponding reference designs, A_007 and A_012 are also plotted for comparison. Figure 6.2 and Figure 6.3 show the layout of each group and the computed Seidels of the reference design and the generated design.

Table 6.4 and Table 6.5 show their corresponding design parameters. For each parameter type in (c, t, n, and v), the four lines from top to the bottom correspond to the value from the first to the last surface, respectively.

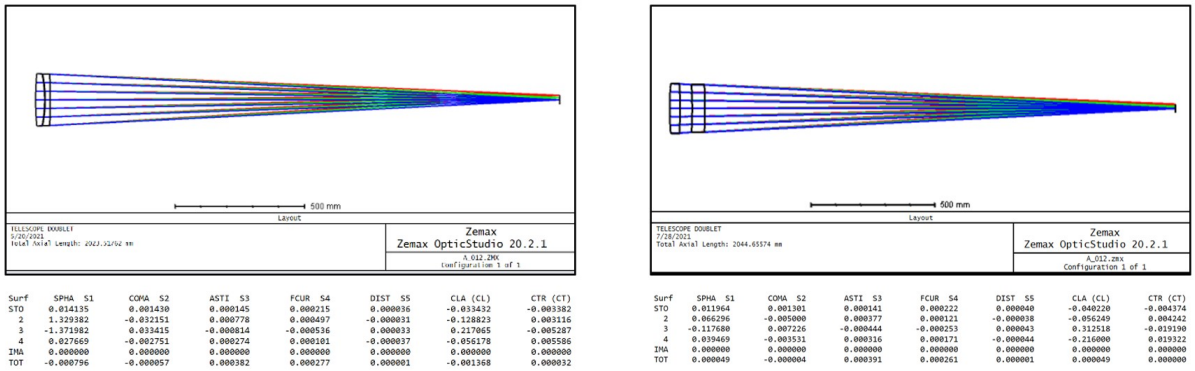
Table 6.4: Design parameters for the lenses in Figure 6.2 (A_007).

	Curvatures (c)	Thicknesses (t)	Refractive indexes (n)	Abbe numbers (v)
Ref	0.020309, -0.021988, -0.023971, -0.0048665	5, 2, 3, 90.8608	1.5168, 1, 1.6483, 1	64.1673, 1e6, 33.8412, 1e6
Gen	0.016766377, -0.009023238, -0.0146893505, -0.0104005793	3.4962142, 3.7846093, 3.6049125, 92.57241058	1.5457263, 1, 1.9979012, 1	75.8944, 1e6, 20.258234, 1e6



(a) Reference design A_007 (b) Generated design for A_007

Figure 6.2: Visualization and analysis of a generated lens in OpticStudio (A_007). (a) the reference design, weighted seidel=0.00152. (b) the generated design, weighted seidel=0.00002.



(a) Reference design A_012 (b) Generated design for A_012

Figure 6.3: Visualization and analysis of a generated lens in OpticStudio (A_012). (a) the reference design, weighted seidel=0.000826. (b) the generated design, weighted seidel=0.00003.

It can be seen from the parameters and layouts that the generated lens designs are different from the reference design, and are indeed satisfying and usable ones with a better KPI.

6.1.6 Ablation Study

In order to show the influence of the supervised and unsupervised terms in the hybrid framework, an ablation study is made. In this part, the network will be trained with mere the supervised data and the unsupervised data respectively, and results will be compared to the hybrid framework. Figure 6.4 is an illustration of the other two, the pure supervised and unsupervised training schemes.

In the supervised training, only the supervised batch of data will be used. The unsupervised term will be also computed for the generated designs. And for the unsupervised training, only the unsupervised data will be used, and the MSE will not be computed since

Table 6.5: Design parameters for the lenses in Figure 6.3 (A_012).

	Curvatures (c)	Thicknesses (t)	Refractive indexes (n)	Abbe numbers (v)
Ref	0.00086266, -0.0019956, -0.001966, -0.00036909	34.8887, 2, 19.318, 1967.311	1.4866, 1, 1.5584, 1	84.468, 1e6, 54.1572, 1e6
Gen	0.0008024741, -0.00043567273, -0.0006658985, -0.00045022965	41.626442, 49.408054, 51.44473, 1902.1765136	1.5724175, 1, 1.9967525, 1	72.61028, 1e6, 20.399572, 1e6

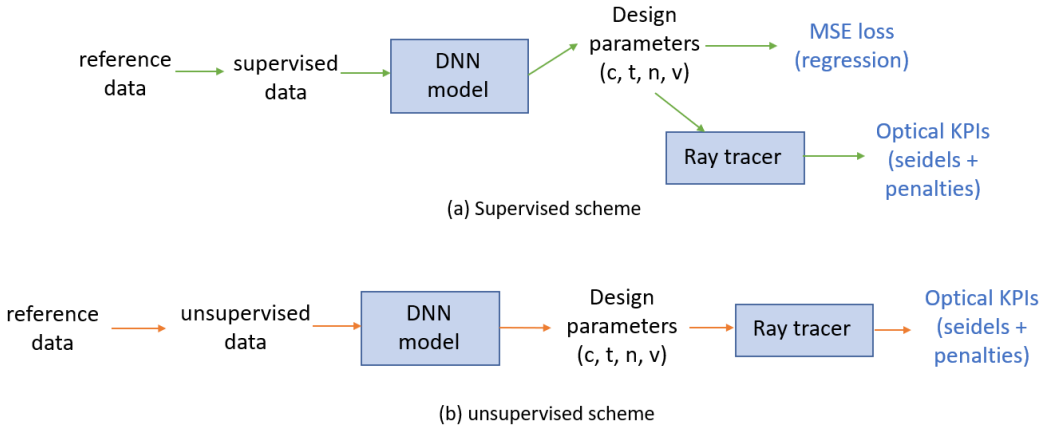


Figure 6.4: Illustration of the single supervised and unsupervised training schemes. (a) supervised scheme. (b) unsupervised scheme.

we do not have any labeled data. The other settings follow the same as in the hybrid training scheme.

Similar to section 6.1.3, the trained models are tested across the whole (HFoV, Enpd) space, but only for EFL=100 since this is a good value to compare both the weighted Seidel and the penalty terms. The mean weighted Seidel for each scheme and the performance profile of the three schemes are shown in Table 6.6 and Figure 6.5.

Table 6.6: Comparison of the hybrid training, supervised training and unsupervised training schemes.

	Seidel_mean	TIR	OP	AP
Hybrid	7.66691e-5	0.11%	6.76%	0.07%
Supervised	2.41006e-2	10.92%	7.17%	31.32%
Unsupervised	4.10439e-3	0.015%	0.01%	4.06%

It can be seen from Table 6.6 that the hybrid scheme provides the best mean weighted Seidel (the Seidel mean in the Table) among the three methods, and the supervised training

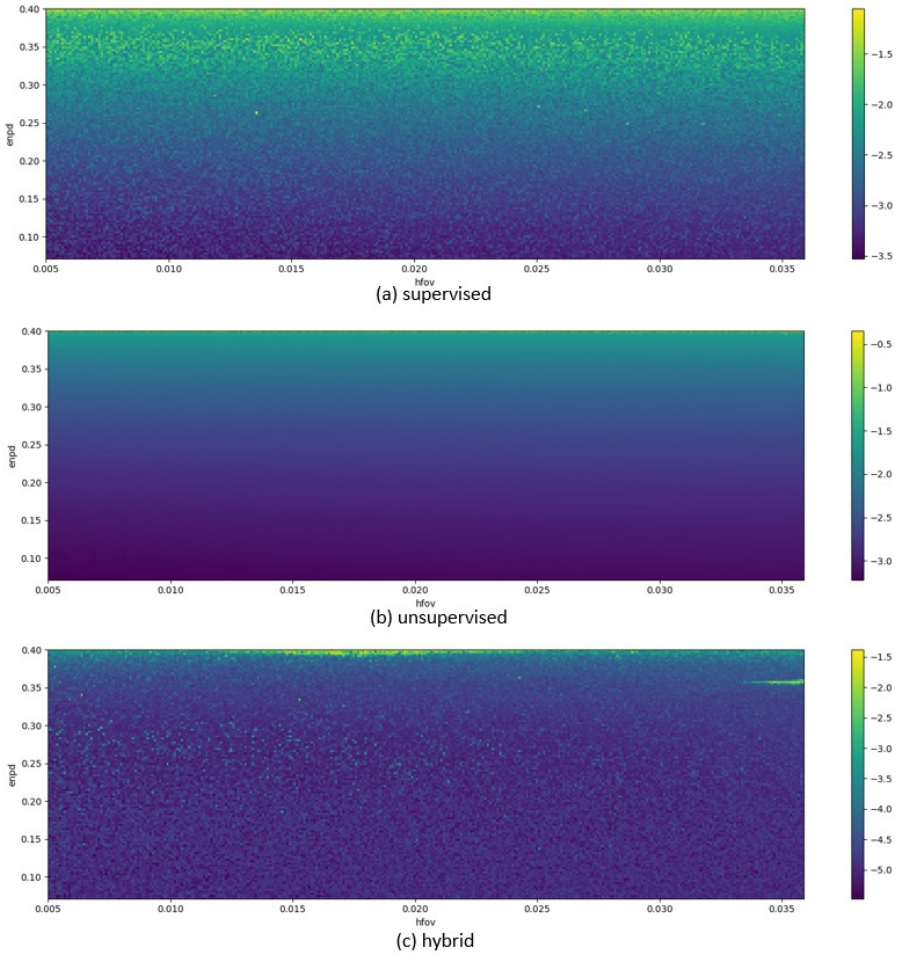


Figure 6.5: Comparison of the seidel profile for the three schemes. (a) supervised. (b) unsupervised. (c) hybrid.

alone is not able to extrapolate to input property triplets that are not in the dataset, as can be seen from the many distinct points with a high value of Seidel in Figure 6.5(a). It has the highest mean weighted Seidel among the three. The unsupervised training alone, provides a moderate level of Seidel, but is still inferior to the hybrid training scheme. However, an interesting phenomenon is the growing trend of the Seidels from the bottom to the top in the unsupervised training, as in Figure 6.5(b). It conforms to the fact in optical design that lenses with a smaller Wfno is more difficult to design and manufacture, as introduced in Chapter 4.

The loss curves of the three schemes are plotted in Figure 6.6. Combined with the loss curves, we can find that the supervised training leads to a significant reduce in the Seidel at the beginning of the training, which helps the network to converge faster; and the unsupervised training in effect encourages the network to try at places other than those in the dataset, therefore leads to a better extrapolation. The combined way inherits the merits of both ways and yields the best performance.

Comparisons to the reference lens designs are also made. For each input property triplet in the reference dataset, 100 designs are generated by the network and their averaged KPIs are computed. Results are shown in Table 6.7. Because of the space limitation, only the

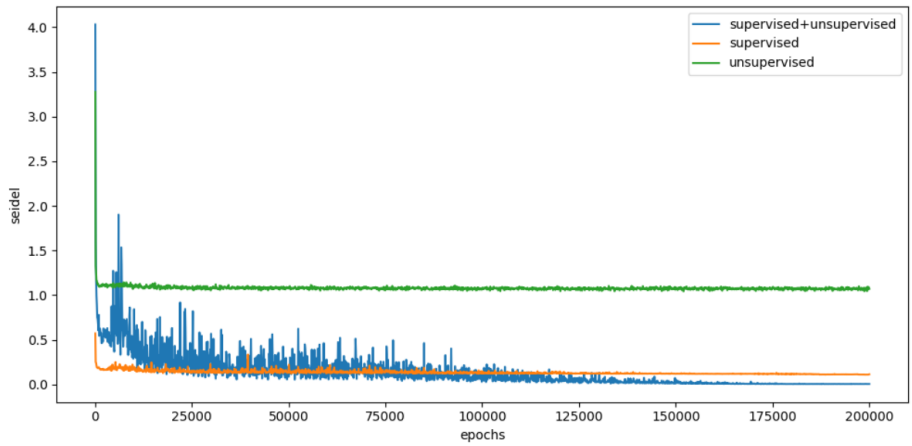


Figure 6.6: The loss curves of the three schemes during training.

weighted Seidel is shown here. The detailed results including the failure rates for the three infeasible cases can be found in Appendix C.

Table 6.7: Comparison of the supervised, unsupervised and hybrid training on the reference dataset.

name	EFL	HFoV	Enpd	Seidel (reference)	Seidel (supervised)	Seidel (unsupervised)	Seidel (hybrid)
A_006	100	2	25	0.0020198	0.0068457	0.009129	0.0034254
A_007	100	2	25	0.0015204	0.0073526	0.0090359	0.0040912
A_008	100	2	25	0.0016723	0.0071939	0.009043	0.0040047
A_010	40	1.5	6.249	0.0001214	0.0030492	0.0013179	0.0000102
A_011	2800	0.5	200	0.0006635	0.1561404	0.0178965	0.0001347
A_012	2000	0.5	200	0.000826	0.0984978	0.0246195	0.0011370
A_013	100	1	14.29	0.000882	0.0151772	0.0026499	0.0000139
A_014	100	1	14.28	0.0006794	0.0040694	0.0026558	0.0000130
A_015	100	1	14.28	0.0001723	0.0195816	0.0026497	0.0000112
A_017	100	1	33.33	0.0044074	0.0323314	0.0166836	0.0001114
A_018	100	1	35.71	0.0043064	0.0361291	0.0194597	0.0013946
A_019	100	1	35.71	0.0069992	0.0320959	0.0199378	0.0011093
A_020	100	1	13.99	0.0002765	0.0170383	0.0025199	0.0000128
A_024	1000	0.5	125	0.0022022	0.0235061	0.0194048	0.0001111
A_025	1000	0.5	125	0.0014837	0.041862	0.0193032	0.0000961
A_028	100	1	20	0.0002764	0.004118	0.0052507	0.0000300
A_029	100	1	20	0.008974	0.0090515	0.0049588	0.0000415

The previous conclusion still holds. Among the three methods, the hybrid training gives the best results. And the supervised training alone with the optimization for the Seidels does not lead to better designs than the reference designs. This is because for the supervised training alone, some reference lens designs in the dataset have similar property triplets as inputs with close or even identical values, but they correspond to very different design parameters (c, t, n, v). The network is essentially a regressor which learns to minimize the MSE loss and in consequence tends to average all the labels with the same input. In this case, the supervised signal from the dataset may accelerate the training of the network at the beginning, but eventually serves as a negative guidance for the training. Nevertheless, the supervised signal leads to a faster convergence of the training, and the extrapolation from the unsupervised signal leads to a good performance throughout the input space.

6.2 Comparison of Baseline with BO

In this section, the effectiveness of the proposed Bayesian Optimization framework will be tested. Experiment settings are firstly explained in section 6.2.1. The results, analysis and visualization are given in the following sections.

In the proposed BO framework, the neural network has the same structure as in the baseline. The difference lies in the samples that are fed into the network. Essentially, the way of sampling is changed. In the baseline approach, the samples in each training epoch are drawn in a totally random manner. While in the BO approach, the next points to sample are proposed by the surrogate model, i.e., the Gaussian Process, by considering the places which are more possible to have a higher Seidel using the acquisition function. Therefore, the training dataset is augmented dynamically according to the current performance of the network, in the hope that within limited tries the BO can explore those more valuable points. In each BO loop, when the training dataset needs to be augmented with a new sample, the current most "difficult" point for the network, where the network has the worst performance, i.e., the highest Seidel, is expected to be suggested by the acquisition function and then added to the training data of the NN.

6.2.1 Experiment Settings

In our experiments, the BO framework is incorporated and tested with the unsupervised scheme in 6.1 for simplicity, without losing the power to verify its effectiveness. It is also possible to be incorporated with the hybrid training scheme, but in this case two GPs will be needed, since the input spaces for the supervised and unsupervised schemes are different. In the supervised scheme, the property triplet is fixed as those in the dataset, and the network will look in the thickness ranges for the next samples. So the input space for the supervised scheme is the 8-dimensional thickness ranges. In the unsupervised scheme, the property triplet is also randomly drawn, therefore it has 11 dimensions. Therefore we will need 2 GPs for the two processes. Meanwhile the training time will be at least doubled, which is expensive to carry on initial experiments and parameter tuning. Another reason is that our goal is to observe the difference brought by changing the sampling method, and the influence by adding the reference data can be coupled with the effect, therefore is to be avoided.

The BO is implemented using the BoTorch [40] library, which is built on top of the efficient and scalable Gaussian Process library GPyTorch [41] in PyTorch, and takes advantage of its GPU acceleration and provides light-weighted high-level APIs. The network is firstly trained on a set of initial samples, and the number is set to be 100. In each BO loop, after the training dataset is augmented with a new sample, the network will be trained for a certain number of epochs, which is set to be 10 in our experiments. The BO loop is repeated for 900 times to get a total number of 1000 samples. This is the BO phase. And since the later added samples may not be seen by the network for enough times, after the training of the BO phase, the network will be trained for another 51,000 epochs as an extension. As the total number of training samples is greatly smaller than in the baseline method, the total number of training epochs is also reduced, which adds up to 60,000. Since the number of training samples is rather small, this is pretty enough for the network to overfit on the dataset. An early stop policy will be adopted, where the network will be tested throughout the input space for multiple times during the training, and as long as an increasing trend of the Seidel is observed, the training will be stopped. The batch size is set as 1024 in each epoch. These

parameters are all tuned empirically through experiments.

As for the training of the GP, in order to save labor in tuning the parameters, the default optimizer in BoTorch, which is the L-BFGS-B optimizer with the default parameters is used. And as for the kernel to be used in the GP, and the acquisition function in the BO process, there are multiple options. Due to the previous experiment results, no obvious periodical tendency can be observed from the seidel profile, therefore two popular and widely used kernels are tested: the RBF kernel and the Matern 3/2 kernel. As for the acquisition function, the Thompson Sampling method, the Upper Confidence Bound and the Expected Improvement are tested.

6.2.2 Comparison with the Baseline

Evaluation is made using the same method by computing the mean of the weighted Seidels throughout the (HFoV, Enpd) space at EFL=100. In the meantime, in order to show the effectiveness of integrating the BO, the standard deviation and maximum value of the Seidels throughout the input space is also computed. A smaller standard deviation indicates a more even performance of the model throughout the input space, and a smaller maximum Seidel indicates a better worst case of the model's performance. Results with different kernels and acquisition functions are shown in Table 6.8.

Table 6.8: Comparison of the mean, std, and max of the Seidels of the baseline and BO approaches.

		mean	std	max
baseline		0.004121	0.005151	1.417223
RBF	TS	0.00372	0.004406	0.032865
	UCB	0.003648	0.004344	0.031931
	EI	0.003974	0.00481	0.032964
Matern3/2	TS	0.004041	0.004885	0.032321
	UCB	0.003675	0.004404	0.032479
	EI	0.003974	0.00481	0.032964

In terms of the mean, standard deviation, and the maximum of the Seidels across the input space, the BO method with all option combinations all give a smaller value than the baseline method. The difference between the kernels are not very significant. As for the acquisition functions, the UCB method seems to be a bit better that the other two. Nonetheless, all of the 6 combinations reduced the mean Seidel level, with a more even performance and a better worst case. Essentially, the BO approach selects the worst point with the acquisition function in each BO epoch, and add it to the training dataset for the network, so it focuses more on the hard points, and in the end improves the worst performance of the model. And as an overall effect, the difference of the performance throughout the input space is reduced, which gives rise to a smaller standard deviation of the Seidels. And the mean is also reduced, which indicates that this optimization focusing on the hard points is more efficient than the random sampling optimization.

Figure 6.7 shows the distribution of all the generated lens designs of the three acquisition functions using the Matern kernel with $\nu = 3/2$. The Seidels of all generated lenses have been

plotted, without taking the mean for a specific input specification, therefore more bad cases can be seen. To show them more clearly, the logarithm of the Seidel values have been taken.

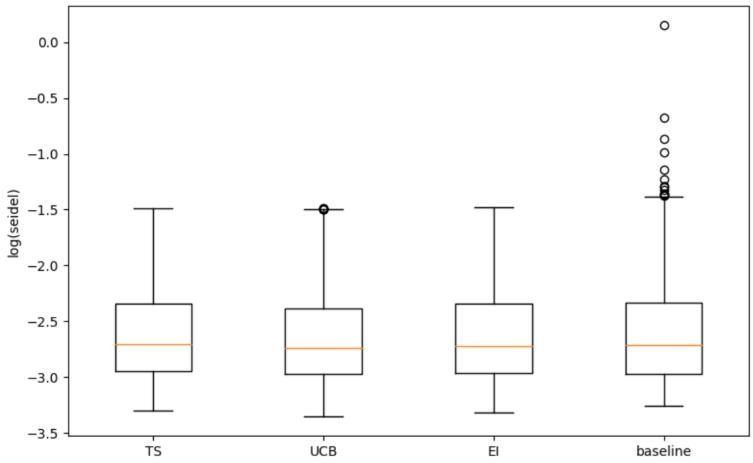


Figure 6.7: Distribution of the seidels of the generated lens designs for the BO approaches and the baseline.

Compared to the three BO approaches with different acquisition functions, there are more number of lens designs which have a very high Seidel, which are outliers in the fourth boxplot. This overall view shows that the BO based approach has less failures throughout the input space.

6.2.3 Visualization and Analysis

A visualization of the proposed samples of different methods is also made. Since it is only able to plot 3 dimensions of the input data, only the values of the (EFL, HFoV, Enpd) triplets of the data are plotted here. Figure 6.8 shows those with the Matern 3/2 kernel. The samples with the RBF kernel are pretty similar, therefore we only give the visualization for one case.

Compare to the random sampling in the baseline method, the changes of the sampling method are directly reflected in the spacial distribution of the samples. Differences are not significant when choosing different kernels, but can be found when comparing different acquisition functions. For the Thompson Sampling and Upper Confidence Bound, it is more straightforward as can be seen in the analytical form of their acquisition functions. They focus more on the high Seidel area by considering the posterior mean in the input space. Therefore the area with a high EFL and Enpd is more frequently exploit, which is the difficult zone. And when that area is over-exploit, the uncertainty at other places also increases and becomes significant, which lead to the exploration to the low Enpd zone. But still the EFL is the dominant factor in the Seidels. For the Expected Improvement, another pattern can be observed. The sampled points spread over the EFL range and stretch to higher Enpd and HFoV values when the EFL is low. This can be partially explained by looking at Equation 2.12. The Enpd is highly related to the radius r in Equation 2.12, therefore a high Enpd will lead to a large S_1. And the height of the ray, $r\cos\varphi$, grows linearly with HFoV. The EFL is however not directly visible in the equation, but it is inter-related to the height of the ray and the parameter HFoV. Lenses with a very small EFL and a large Enpd, normally indicate

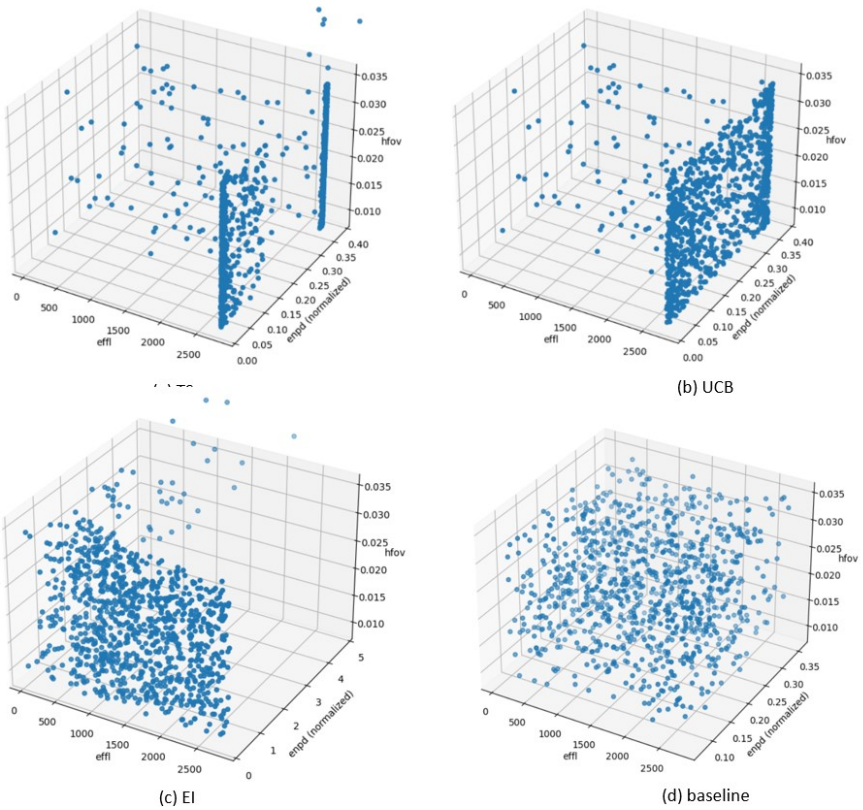


Figure 6.8: Samples by different acquisition functions and in the baseline, using the Matern 3/2 kernel. (a) Thompson Sampling. (b) Upper Confidence Bound. (c) Expected Improvement. (d) Baseline.

a high HFoV, which means the lens needs to have a very strong ability to converge the rays coming from a wider view. As a fact, there are very few reference designs of this type, which to some extent indicates they are difficult to design.

Chapter 7

Conclusions and Further Discussion

In this work, the application of deep learning to the problem of starting points generation in optical lens design is investigated. A stacked 7-layer baseline neural network is firstly built to generate lens parameters given the desired property parameters of the lens system. A hybrid training method is adopted, in which the network learns both from the reference data and optimizes the optical KPIs. In addition, due to the scarcity of optical data, a strong data augmentation is used.

Results show that for the achromatic aplanat doublets, the network is able to produce designs for any given input property triplet, with small optical KPIs, and most of them are even better than those in the reference dataset. Compared to the supervised training or unsupervised training alone, the hybrid training which combines the two yields designs with better optical KPI values.

To improve the network's performance at the area in the input space with a high KPI value, a Bayesian Optimization based joint learning framework is proposed. A Gaussian Process is built as the surrogate model to describe the function from network inputs to optical KPIs, in order to select the hard, or most valuable points as the samples for the network to be trained on. Results show the effectiveness of this improved method integrated with the Bayesian Optimization framework. Experiments are done with multiple options for the kernel and acquisition function, and all of them exceed the performance of the baseline approach. The BO assisted approaches have better worst case performance than the baseline approach.

The type of design which is studied in this work, the air-spaced doublet achromatic aplanat, turns our to be a relatively simple lens for the neural network to design. If the input space is sufficiently sampled, the proposed baseline approach with random sampling is already able to generate satisfying results. And the advantage of the proposed Bayesian Optimization approach is not very notable if compared to the sufficient random sampling. However, the BO approach indeed shows its superiority when the input space cannot be sufficiently sampled, therefore it is very likely that the difference will become significant with the increase of the dimension of the problem, in other words, when the type of design to be investigated is more complex.

Nonetheless, this work serves as an initial step of the machine learning assisted automated starting point generation in optical lens design, with the focus on a simple and fundamental building block of optical lenses. And the framework as well as the differentiable ray tracer built from scratch, are highly extendable for more complex lens types in further study. In future work, many meaningful and interesting aspects can be the directions. For example,

investigating into more complex types of lens designs with increased number of elements; or improving the glass model to be closer to preferred materials; or incorporating objectives which consider the problems in manufacturing, and so on.

Bibliography

[1] Wikipedia: Optical lens design. https://en.wikipedia.org/wiki/Optical_lens_design. Accessed March 25, 2021. 2

[2] M. J. Kidger. Fundamental optical design. SPIE Bellingham, 2001. 2, 13, 14, 54, 56

[3] W. T. Welford. *Aberrations of optical systems*. Routledge, 2017. 2, 7, 8, 13, 14, 15, 20

[4] Geometric lens distortion: a common problem in photography. https://www.dxo.com/technology/lens-distortions/. Accessed August 11, 2021. 3

[5] What causes chromatic aberration in photography and how do camera lenses correct it. https://www.reddit.com/r/explainlikeimfive/comments/6x1rpr/eli5_what_causes_chromatic_aberration_in/. Accessed August 11, 2021. 3

[6] Opticstudio. https://www.zemax.com/products/opticstudio. Accessed July 28, 2021. 3, 19

[7] Codev. https://www.synopsys.com/optical-solutions/codev.html. Accessed July 28, 2021. 3, 19

[8] C. Gagné, J. Beaulieu, M. Parizeau, and S. Thibault. Human-competitive lens system design with evolution strategies. *Applied Soft Computing*, 8(4):1439–1452, 2008. 4

[9] C. Menke. Application of particle swarm optimization to the automatic design of optical systems. In *Optical Design and Engineering VII*, volume 10690, page 106901A. International Society for Optics and Photonics, 2018. 4

[10] M. van Turnhout, P. van Grol, F. Bociort, and H. P. Urbach. Obtaining new local minima in lens design by constructing saddle points. *Optics express*, 23(5):6679–6691, 2015. 4

[11] R. S. Hegde. Accelerating optics design optimizations with deep learning. *Optical Engineering*, 58(6):065103, 2019. 4

[12] C. Gannon and R. Liang. Using machine learning to create high-efficiency freeform illumination design tools. *arXiv preprint arXiv:1903.11166*, 2018. 4

[13] 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. *Optics express*, 27(12):17228–17238, 2019. 4

[14] C. C. Nadell, B. Huang, J. M. Malof, and W. J. Padilla. Deep learning for accelerated all-dielectric metasurface design. *Optics express*, 27(20):27523–27535, 2019. 4

[15] G. Côté, J. F. Lalonde, and S. Thibault. Extrapolating from lens design databases using deep learning. *Optics express*, 27(20):28279–28292, 2019. 4, 5, 22, 32, 33, 36

[16] G. Côté, J. F. Lalonde, and S. Thibault. Introducing a dynamic deep neural network to infer lens design starting points. In *Current Developments in Lens Design and Optical Engineering XX*, volume 11104, page 1110403. International Society for Optics and Photonics, 2019. 4, 22

[17] G. Côté, J. F. Lalonde, and S. Thibault. Deep learning-enabled framework for automatic lens design starting point generation. *Optics Express*, 29(3):3841–3854, 2021. 4, 32

[18] M. Katz. *Introduction to geometrical optics*. World scientific, 2002. 6, 12

[19] Wikipedia: Snell's law. `https://en.wikipedia.org/wiki/Snell%27s_law`. Accessed July 28, 2021. 6

[20] Wikipedia: Fraunhofer lines. `https://en.wikipedia.org/wiki/Fraunhofer_lines`. Accessed July 28, 2021. 9, 15

[21] Five seidel aberrations. `https://www.jeol.co.jp/en/words/emterms/search_result.html?keyword=Five%20Seidel%20aberrations`. Accessed August 12, 2021. 16

[22] Wikipedia: Chromatic aberrations. `https://en.wikipedia.org/wiki/Chromatic_aberration`. Accessed August 12, 2021. 17

[23] Wikipedia: Saggita (optics). `https://en.wikipedia.org/wiki/Sagitta_(optics)`. Accessed July 28, 2021. 17

[24] E. Hecht. *Optics, 5e*. Pearson Education India, 2002. 18

[25] Optimax: sphere limits. `https://www.optimaxsi.com/capabilities/spheres/#manufacturing-limits`. Accessed July 28, 2021. 18

[26] V. P. Noël, S. Masoumi, E. Parham, G. Genest, L. Bégin, M. A. Vigneault, and D. C. Côté. Tools and tutorial on practical ray tracing for microscopy. *Neurophotonics*, 8(1):010801, 2021. 19

[27] A. Gerrard and J. M. Burch. *Introduction to matrix methods in optics*. Courier Corporation, 1994. 19

[28] Pytorch. `https://pytorch.org/`. Accessed July 28, 2021. 22

[29] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-normalizing neural networks. In *Proceedings of the 31st international conference on neural information processing systems*, pages 972–981, 2017. 22

[30] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. 26

[31] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015. 27

[32] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. 27, 28

[33] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. 29

[34] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. 29

[35] J. Močkus. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975. 29

[36] K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos. Parallelised bayesian optimisation via thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142. PMLR, 2018. 29

[37] Zemax optical design collection: Zebase 6. `http://zemax.tw/ZMXLLC/media/PDFLibrary/Brochures/Zebase-sell-sheet_141120-(1).pdf?ext=.pdf`. Accessed July 29, 2021. 31

[38] M. Laikin. *Lens design*. CRC Press, 2018. 31

[39] Schott glass catalog. `https://www.schott.com/advanced_optics/english/products/optical-materials/optical-glass/optical-glass/index.html`. Accessed March 25, 2021. 33

[40] Botorch. `https://botorch.org/`. Accessed July 29, 2021. 45

[41] Gpytorch. `https://gpytorch.ai/`. Accessed July 29, 2021. 45

Appendix A

Derivation of the Spherical Aberration (S_1)

The Spherical Aberration can be derived in many ways. Here one simple version from [2] is shown.

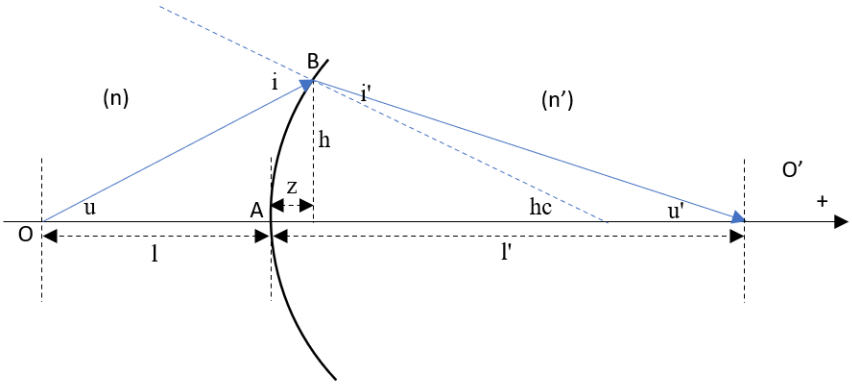


Figure A.1: Spherical aberration at a single surface.

In Figure A.1, AB is a spherical lens surface with a radius R. Point O is the object point on the optical axis, and O' is the corresponding image point. The height of the ray at this surface is denoted as h. The object distance and image distance is denoted as l and l', and the refractive index of the first and second medium is n and n', respectively. The Spherical Aberration in the total wavefront aberration is computed as the difference of the lengths of two optical paths $[OBO']$ and $[OAO']$:

$$\begin{aligned} W &= [OAO'] - [OBO'] \\ &= n(OA - OB) + n'(AO' - BO') \end{aligned} \tag{A.1}$$

The problem is then transformed into finding the appropriate approximation of OB and BO'.

In geometrical optics, there are several sign conventions:

- Ray heights of the rays are positive if the rays are above the optical axis, and are negative if the rays are below the optical axis.

- Distances are positive if they are to the right of a surface, and negative to the left of a surface.
- Ray angles are positive if their slopes in the meridian plane are positive, and are negative if the slopes are negative.

For OB,

$$OB^2 = (z - l) + h^2 \tag{A.2}$$

where z is the axial displacement of the ray's intersection at the surface, computed by:

$$z = \frac{1 - \sqrt{1 - c^2h^2}}{c} \tag{A.3}$$

If we considers zc, it can be expressed using a power series in h^2:

$$\begin{aligned} zc &= 1 - \sqrt{1 - c^2h^2} \\ &= 1 - \left(1 - \frac{1}{2}c^2h^2 - \frac{1}{8}c^4h^4 - \frac{1}{16}c^6h^6 - \cdots\right) \end{aligned} \tag{A.4}$$

Therefore we have an approximation of z:

$$z = \frac{1}{2}ch^2 + \frac{1}{8}c^3h^4 + \frac{1}{16}c^5h^6 + \cdots \tag{A.5}$$

In deriving the Seidel aberrations, the terms with h^6 and higher orders are ignored. If we substitute Equation A.2 with Equation A.5, we obtain:

$$\begin{aligned} OB^2 &= \left(\frac{1}{2}ch^2 + \frac{1}{8}c^3h^4 - l\right) + h^2 \\ OB^2 &= l^2\left[1 - \left(c - \frac{1}{l}\right)\frac{h^2}{l} - \left(c - \frac{1}{l}\right)\frac{c^2h^4}{4l}\right] \end{aligned} \tag{A.6}$$

In the paraxial zone, $h \ll l$, hence we have:

$$OB = -l + \frac{1}{2}h^2\left(c - \frac{1}{l}\right) + \frac{1}{8}c^2h^4\left(c - \frac{1}{l}\right) + \frac{1}{8l}h^4\left(c - \frac{1}{l}\right)^2 \tag{A.7}$$

and

$$OA - OB = -\frac{1}{2}h^2\left(c - \frac{1}{l}\right) - \frac{1}{8}c^2h^4\left(c - \frac{1}{l}\right) - \frac{1}{8l}h^4\left(c - \frac{1}{l}\right)^2 \tag{A.8}$$

Similarly, we can obtain

$$AO' - BO' = -\frac{1}{2}h^2\left(c - \frac{1}{l'}\right) - \frac{1}{8}c^2h^4\left(c - \frac{1}{l'}\right) - \frac{1}{8l'}h^4\left(c - \frac{1}{l'}\right)^2 \tag{A.9}$$

Then the Spherical Aberration can be computed by:

$$\begin{aligned} W &= n(OA - OB) + n'(AO' - BO') \\ &= \frac{1}{2}h^2\left[n'\left(c - \frac{1}{l'}\right) - n\left(c - \frac{1}{l}\right)\right] \\ &+ \frac{1}{8}c^2h^4\left[n'\left(c - \frac{1}{l'}\right) - \left(c - \frac{1}{l'}\right)\right] \\ &+ \frac{1}{8l'}h^4\left[\frac{n'}{l'}\left(c - \frac{1}{l'}\right)^2 - \frac{n}{l}\left(c - \frac{1}{l'}\right)^2\right] \end{aligned} \tag{A.10}$$

If O' is a paraxial image point, we have [2]:

$$\frac{n'}{l'} = \frac{n}{l} + (n' - n)c \tag{A.11}$$

or

$$n'\left(c - \frac{1}{l'}\right) = n\left(c - \frac{1}{l}\right) \tag{A.12}$$

Hence, Equation A.10 can be simplified to:

$$W = \frac{1}{8l'}h^4\left[\frac{n'}{l'}\left(c - \frac{1}{l'}\right)^2 - \frac{n}{l}\left(c - \frac{1}{l'}\right)^2\right] \tag{A.13}$$

Since in the paraxial zone,

$$\begin{aligned} u &= -\frac{h}{l} \\ u' &= -\frac{h}{l'} \\ i &= u + hc \end{aligned} \tag{A.14}$$

Substitute them into Snell's law $A = ni = n'i'$ (in the paraxial zone $\sin(i) = i$),

$$\begin{aligned} A = n(hc + u) &= n'(hc + u') \\ nh\left(c - \frac{1}{l}\right) &= n'h\left(c - \frac{1}{l'}\right) \end{aligned} \tag{A.15}$$

This gives us:

$$\begin{aligned} c - \frac{1}{l} &= \frac{A}{nh} \\ c - \frac{1}{l'} &= \frac{A}{nh'} \end{aligned} \tag{A.16}$$

Substitute them into Equation A.13, we get:

$$\begin{aligned} W &= \frac{1}{8}h^4\left[\frac{n'}{l'}\left(\frac{A}{n'h}\right)^2 - \frac{n}{l}\left(\frac{A}{nh}\right)^2\right] \\ &= \frac{1}{8}A^2h\left(\frac{h}{n'l'} - \frac{h}{nl}\right) \\ &= \frac{1}{8}A^2h\left(\frac{u}{n} - \frac{u'}{n'}\right) \\ &= -\frac{1}{8}A^2h\delta\left(\frac{u}{n}\right) \end{aligned} \tag{A.17}$$

with

$$\delta\left(\frac{u}{n}\right) = \frac{u'}{n'} - \frac{u}{n} \tag{A.18}$$

which gives the final form in Equation 2.6.

Appendix B

Comparison of the implemented ray tracer on the dataset.

		SPHA S1	COMA S2	ASTI S3	FCUR S4	DIST S5	CLA S6	CTR S7
A_006	OS	0.001630	0.001251	0.003107	0.001205	-0.000107	-0.001126	0.001256
	PyTorch	0.001618	0.001252	0.003107	0.001205	-0.000107	-0.001125	0.001256
A_007	OS	0.000389	0.000771	0.002254	0.001314	-0.000012	-0.001869	0.000307
	PyTorch	0.000370	0.000772	0.002254	0.001314	-0.000012	-0.001869	0.000307
A_008	OS	-0.000228	0.000719	0.002076	0.001326	0.000007	-0.002416	0.000157
	PyTorch	-0.000212	0.000719	0.002076	0.001326	0.000007	-0.002416	0.000157
A_010	OS	0.000248	0.000004	0.000165	0.000120	0.000001	-0.000171	-0.000005
	PyTorch	0.000248	0.000004	0.000165	0.000120	0.000001	-0.000171	-0.000005
A_012	OS	-0.000784	-0.000057	0.000382	0.000277	0.000001	-0.001369	0.000032
	PyTorch	-0.000790	-0.000058	0.000382	0.000277	0.000001	-0.001369	0.000032
A_013	OS	0.004425	-0.000129	0.000105	0.000097	-0.000006	-0.000431	0.000097
	PyTorch	0.004418	-0.000130	0.000105	0.000097	-0.000006	-0.000430	0.000097
A_014	OS	0.001850	-0.000140	0.000158	0.000111	0.000001	-0.000746	-0.000010
	PyTorch	0.001845	-0.000140	0.000158	0.000111	0.000001	-0.000745	-0.000010
A_015	OS	-0.000037	0.000076	0.000166	0.000111	0.000003	-0.000242	-0.000018
	PyTorch	-0.000039	0.000076	0.000166	0.000111	0.000003	-0.000241	-0.000018
A_017	OS	0.015911	0.000988	0.001042	0.000585	-0.000003	-0.003502	0.000364
	PyTorch	0.015763	0.000991	0.001042	0.000585	-0.000003	-0.003494	0.000364
A_018	OS	0.015445	0.000214	0.001157	0.000538	-0.000008	-0.004047	0.000495
	PyTorch	0.015417	0.000215	0.001157	0.000538	-0.000008	-0.004045	0.000495
A_019	OS	0.023586	0.001579	0.001031	0.000694	0.000004	-0.006347	0.000148
	PyTorch	0.023787	0.001575	0.001031	0.000694	0.000004	-0.006355	0.000148
A_020	OS	-0.000642	-0.000002	0.000169	0.000109	0.000000	-0.000379	0.000013
	PyTorch	-0.000633	-0.000002	0.000169	0.000109	0.000000	-0.000380	0.000013
A_024	OS	-0.003934	0.000652	0.000308	0.000206	0.000000	-0.002709	0.000057
	PyTorch	-0.003952	0.000652	0.000308	0.000206	0.000000	-0.002712	0.000057
A_025	OS	-0.000323	-0.000059	0.000298	0.000205	0.000000	-0.002720	0.000105
	PyTorch	-0.000339	-0.000059	0.000298	0.000205	0.000000	-0.002714	0.000105
A_028	OS	0.000342	0.000013	0.000331	0.000206	0.000000	-0.000382	0.000072
	PyTorch	0.000321	0.000014	0.000331	0.000206	0.000000	-0.000379	0.000072
A_029	OS	-0.017991	-0.010798	0.000024	0.000134	-0.000021	-0.002365	0.000296
	PyTorch	-0.018085	-0.010799	0.000024	0.000134	-0.000021	-0.002361	0.000296
Average relative difference		2.1322%	0.0039%	0.0001%	0.0002%	0.0002%	0.0024%	0.0002%

Appendix C

Detailed evaluation of the supervised, unsupervised and the hybrid training schemes on the reference dataset.

fileName	seidel ref	seidel su	seidel s	seidel u	TIRref	TIRsu	TIRs	TIRu	OPref	OPsu	OPs	OPu	APref	APsu	APs	APu
A_006	0.0020198	0.0034254	0.0068457	0.009129	0	0	0	0	0	0.12	0	0	0	0	0	0.1
A_007	0.0015204	0.0040912	0.0073526	0.0090359	0	0	0	0	0	0.14	0	0	0	0	0	0.1
A_008	0.0016723	0.0040047	0.0071939	0.009043	0	0	0	0	0	0.09	0	0	0	0	0	0.1
A_010	0.0001214	0.0000102	0.0030492	0.0013179	0	0	0.05	0	0.11	0.14	0.1	0	0	0	0	0
A_011	0.0006635	0.0001347	0.1561404	0.0178965	0	0	0.27	0	1.14	0	0	0	0	0.01	0.7	0
A_012	0.000826	0.0011370	0.0984978	0.0246195	0	0	0.18	0	0	0	0	0	0	0	0.6	0
A_013	0.000882	0.0000139	0.0151772	0.0026499	0	0	0.02	0	0	0.03	0	0	0	0	0.1	0
A_014	0.0006794	0.0000130	0.0040694	0.0026558	0	0	0.06	0	0	0.1	0.1	0	0	0	0.1	0
A_015	0.0001723	0.0000112	0.0195816	0.0026497	0	0	0.02	0	0	0.03	0	0	0	0	0.1	0
A_017	0.0044074	0.0001114	0.0323314	0.0166836	0	0	0.07	0	0	0.1	0.1	0	0	0	0	0.3
A_018	0.0043064	0.0013946	0.0361291	0.0194597	0	0	0.01	0	0	0.07	0.2	0	0	0	0.1	0.2
A_019	0.0069992	0.0011093	0.0320959	0.0199378	0.024	0.02	0.01	0	0	0.09	0.1	0	0	0	0.1	0.3
A_020	0.0002765	0.0000128	0.0170383	0.0025199	0	0	0.05	0	0	0.09	0.1	0	0	0	0.1	0
A_024	0.0022022	0.0001111	0.0235061	0.0194048	0	0	0.06	0	0	0	0	0	0	0	0.2	0
A_025	0.0014837	0.0000961	0.041862	0.0193032	0	0	0.07	0	0	0	0	0	0	0	0.3	0
A_028	0.0002764	0.0000300	0.004118	0.0052507	0	0	0	0	0	0.05	0	0	0	0	0	0.1
A_029	0.008974	0.0000415	0.0090515	0.0049588	0.016	0	0.01	0	0	0.1	0	0	0	0	0	0