

Regulation-aware freeform headlamp reflector design with differentiable ray tracing

XUCHEN WEI,¹  YUCHI HUO,¹  PENGFEI SHEN,²  YIFAN PENG,^{2,3}  HUJUN BAO,¹ AND RUI WANG^{1,4} 

¹The State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China

²The University of Hong Kong, Hong Kong SAR, China

³evanpeng@hku.hk

⁴ruiwang@zju.edu.cn

Abstract: Headlamp design is an essential aspect of the automotive industry, often relying on reflector systems composed of freeform surfaces. Traditional methods depend on manually adjusting surfaces to ensure reflected rays meet regulation requirements, which is time-consuming and often limited by design features. In this work, we propose a regulation-aware approach to freeform headlamp reflector design using differentiable ray tracing. The reflectors are modeled with B-spline surface arrays, and we optimize the control point grid through a customized ray tracing pipeline utilizing triangle mesh acceleration structures for efficient ray-surface intersections. Additionally, we use multi-layer perceptrons (MLPs) to generate light intensity distributions guided by loss functions based on regulation constraints. Experimental results show that this neural process generates the required four types of passing-beam and driving-beam distributions defined in the ECE R112. Using these distributions as targets, we can optimize reflector shapes in about 200 iterations. We also export the optimized reflectors as CAD models, achieving 2.5× faster simulation in our pipeline with accuracy comparable to commercial optical design software, aka, LightTools.

© 2025 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

Automotive headlamp design is critical in ensuring safety and performance, requiring advanced techniques to achieve precise light distribution and high efficiency. Passing-beam and driving-beam modules, as essential components of automotive headlamps, require careful fine-tuning by professional designers to meet detailed regulation requirements, such as ECE R112 in Europe [1]. The design of automotive headlamp systems is divided into three main categories: projection-type [2,3], refraction-type [4,5], and reflection-type [6,7]. Projection-type systems typically use a reflector, baffle, and lens to form specific beam patterns, which may result in energy loss. Refraction-type designs simplify the structure but often suffer from dispersion due to lens reliance. Our work focuses on reflection-type headlamps due to their widespread use, offering flexibility in surface shaping, adaptability to various light sources and component sizes, and precise beam control through direct reflection.

Free-form optics algorithms provide powerful mathematical tools for the design of reflection-type headlamp reflectors. Traditional techniques include the Monge-Ampère method [8,9], supporting quadratic method [10], and ray-mapping method [11,12]. Some works [13,14] have improved traditional algorithms designed for ideal light sources to support extended light sources. However, these methods only involve single- or double-surface freeform systems and are not applicable to multiple freeform surfaces. Headlamp reflectors typically use multiple freeform surfaces connected by interval gap surfaces to construct compact shell structures, thus meeting the requirements of limited space and installation conditions. This allows each reflective surface to control the direction of the reflected rays independently, making it easier to produce uniform illumination or other specific beam shapes. In contrast, using only a few freeform surfaces often

fails to satisfy design and manufacturing needs. Therefore, designing reflector headlamps with multiple freeform surfaces is a complex and challenging task.

Recent advances in differentiable ray tracing algorithms [15] and machine learning tools [16,17] have provided new methods and insights for traditional freeform optical design, enabling automated gradient-based optimization for complex optical system tasks [18]. These tasks include end-to-end lens design [19,20], diffractive optical element design [21,22], and freeform lens design [23,24]. Among existing methods, B-spline surfaces and their variants are the most widely used representations for modeling differentiable freeform surfaces [25,26]. Some studies have also explored addressing differentiable ray B-spline intersection problems, focusing on improving either precision [27] or computational efficiency [28]. To the best of our knowledge, limited research focuses on reflector designs for automotive headlamps using differentiable ray tracing. Meanwhile, existing open-source tools, such as Mitsuba [29,30], or commercial softwares [31–33], can not model the differentiable B-spline surface array. This prompted us to develop a novel differentiable ray tracing pipeline for headlamp reflectors design, combining nondifferentiable triangle mesh intersection with differentiable surface evaluation to enhance the optimization capabilities of multiple freeform surface systems.

Our differentiable ray tracing pipeline aims to optimize reflectors to ensure the simulated light beams meet regulation requirements. Specifically, headlamp regulations define a set of discrete points and zones for passing-beam and driving-beam where the actual light intensity value must satisfy inequality constraints. However, these discrete constraints cannot directly drive surface optimization in our pipeline, as only a small subset of surface points contribute reflected rays to these constraints during optimization, leading to localized gradients and resulting in sharp details rather than smooth, natural surfaces. To address this issue, we must generate a global light intensity distribution across the photometric range as the optimization target, ensuring that as much as ray samples contribute to gradient computation on the sensor. By designing reasonable loss functions and utilizing the embedding mechanism of the MLP, we map the regulation data into a feature space, enabling it to generate desired distributions within a photometric range.

In summary, our work proposes a regulation-aware freeform headlamp reflector design method, utilizing a neural process to generate light intensity distributions that comply with regulation constraints. We use the generated light intensity distribution to drive a differentiable ray tracing pipeline for automated optimization of B-spline surface control point grids. Specifically, we make the following key contributions:

- We present a differentiable ray tracing pipeline tailored for optimizing multiple freeform surfaces in the optical system of headlamp reflectors.
- We explore an innovative regulation-aware neural process capable of generating light intensity distributions that meet regulation constraints within a specified photometric range.
- We develop a hybrid computational framework that integrates ray tracing acceleration structures with differentiable surface computation, greatly improving optimization efficiency and performance.

Figure 1 illustrates the overall pipeline and modules of this work. Section 2 introduces the specific methods and Section 3 presents experimental results that demonstrate the effectiveness of the proposed method.

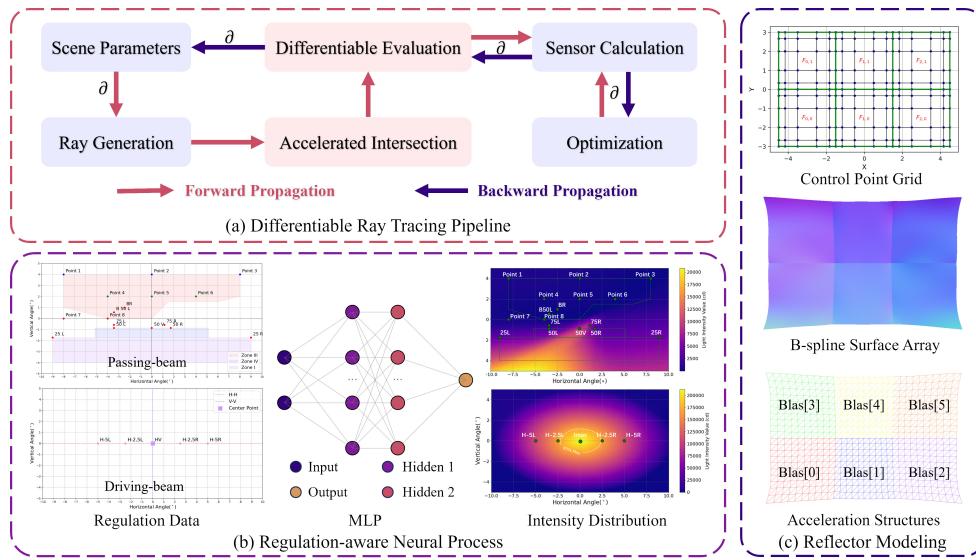


Fig. 1. System overview: The workflows of the differentiable ray tracing pipeline (a), regulation-aware neural process (b), and reflector modeling module (c) are illustrated. We use ∂ to indicate that the calculation is differentiable and colored arrows to show gradient propagation directions; the neural process takes regulation data as input and outputs light intensity distributions through an MLP; and the reflector module constructs surface arrays from a control point grid and triangulates them to build ray tracing acceleration structures.

2. Method

2.1. Differentiable representation for B-spline surface array

We model the headlamp reflector as a B-spline surface array, with classical mathematical definition [34] for each surface:

$$F(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_i^p(u) N_j^q(v) P_{ij}, \quad (1)$$

where $N_i^p(u)$ and $N_j^q(v)$ are the B-spline basis functions of degrees p and q in the u and v directions, and P_{ij} are the control points in \mathbb{R}^3 with a grid size of $(m+1) \times (n+1)$. The z -coordinate of each P_{ij} is differentiable, while the x - and y - coordinates are constant, ensuring the outline shape of each facet remains rectangular. We represent the B-spline surface array as a union of B-spline surfaces sharing a global control point grid:

$$S(x, y) = \bigcup_{i=0}^{N-1} \bigcup_{j=0}^{M-1} F_{ij}(\bar{u}(x), \bar{v}(y)), \quad (2)$$

where F_{ij} represents a facet of the reflector with N rows and M columns, \bar{u} and \bar{v} denotes linear transformation from the reflector coordinate space (x, y) to the local space (u, v) of each facet. The width and height of the reflector are determined by the sum of all facets and the gaps along the x - and y -directions, respectively. All facets in the same row share the same height, the number of control points, and degree in the v -direction, while those in the same column share the same width, the number of control points, and degree in the u -direction. This ensures a global grid that stores the control point coordinates for all facets, allowing the boundary control points between adjacent

facets to be consistent. As shown in Fig. 2, we illustrate various configurations for control point grid layouts. For cases where adjacent facets need to be offset along the z-axis, we introduce gaps to ensure that the boundary control points are separable. For no-gap configurations, adjacent facets share control points along their boundaries. Specifically, facets in the same row share control points along vertical boundaries, while those in the same column share control points along horizontal boundaries. At intersections of rows and columns, corner control points are shared by all adjacent facets.

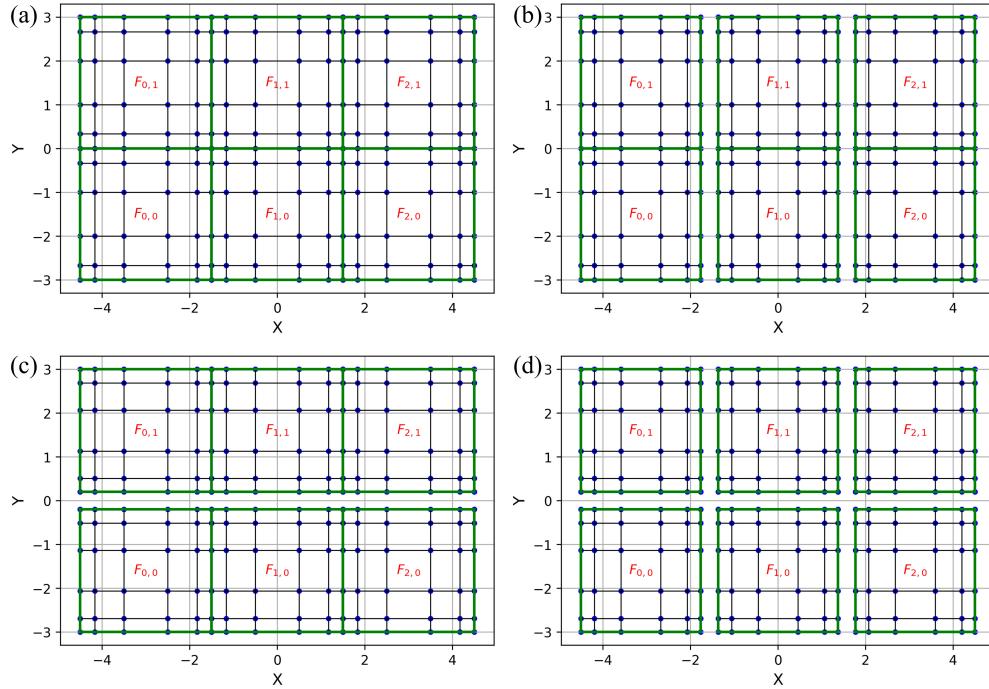


Fig. 2. Control points base grid of B-spline surface array. We present four commonly used modeling strategies, including (a) uv no-gap, (b) u gap and v no-gap, (c) u no-gap and v gap, and (d) uv gap. Green facet boundaries enclose each facet, blue dots mark the control points, and red text $F_{i,j}$ indicates the arrangement index of each facet in the surface array.

In our implementation, the reflector maintains the z -coordinates of the control point grid as a tensor with a gradient. Adjacent boundary points share values between facets, and each facet uses *index_slicing* to access the corresponding values of the global tensor. Each facet performs differentiable computations based on local references to the control point grid, which enables gradients to back-propagate through the grid and enforce consistency at adjacent boundaries. In addition, we use Greville abscissae [35] to calculate the x - and y - coordinates of the control points, aiming to achieve uniform tessellation in the parameter space to construct a triangle mesh. This approach closely aligns with the implementation described in the Ref. [26].

2.2. Accelerated intersection for differentiable ray tracing

The computation of differentiable intersections between rays and freeform surfaces is a challenging problem because it is usually difficult to find analytical solutions. A ray is represented by the equation $\mathbf{o} + t\mathbf{d}$, with \mathbf{o} as the origin, \mathbf{d} as the direction, and t as the parameter denoting propagation distance. Research by dO [19] introduced an efficient method that differentially solves the t using just a few numerical iterations, significantly reducing gradient memory overhead. However,

their work still encounters bottlenecks in the task of differentiable non-sequential ray tracing across multiple freeform surfaces. Because rays need to traverse each surface, even with a small number of iterations per surface, this can lead to high computational costs. Instead, we adopted a two-level acceleration structure for ray tracing to efficiently handle optical systems represented by triangulated meshes. The top-level acceleration structure (TLAS) identifies which facet of the optical system the ray intersects, whereas the bottom-level acceleration structure (BLAS) determines which triangle within that facet is intersected. Our method maps the triangle intersection point directly to the parameter space of the B-spline surface, eliminating the need for numerical iteration. Sufficient tessellation resolution ensures that the error between the triangle mesh and the original surface is small enough to achieve acceptable accuracy for ray-surface intersections. Although the mapped surface point deviates from the original ray direction, it introduces only a slight perturbation, keeping the ray propagation still differentiable.

As shown in Fig. 3, we visualize the z -coordinates deviation and normals deviation between the triangulated mesh and the original surface at different resolutions. In the [Supplement 1](#), we have also designed an experimental case to present the simulation deviation of outgoing rays between the B-spline surface and triangle mesh. It is worth noting that triangulations of any resolution are feasible and can be adjusted based on the requirements of specific optical systems. We provide the pseudocode of the accelerated differentiable ray tracing, as shown in Algorithm 1. After obtaining the t values and facet indices of the intersection points, we use a tensor-based *mask_selected* operation to filter rays hitting each facet. We map the (u, v) coordinates of the intersection points on the triangle meshes to the parameter space of the corresponding B-spline surface, where we perform differentiable evaluation and ray sampling. We integrated the Intel Embree [36] ray tracing library into our customized pipeline to accelerate intersections.

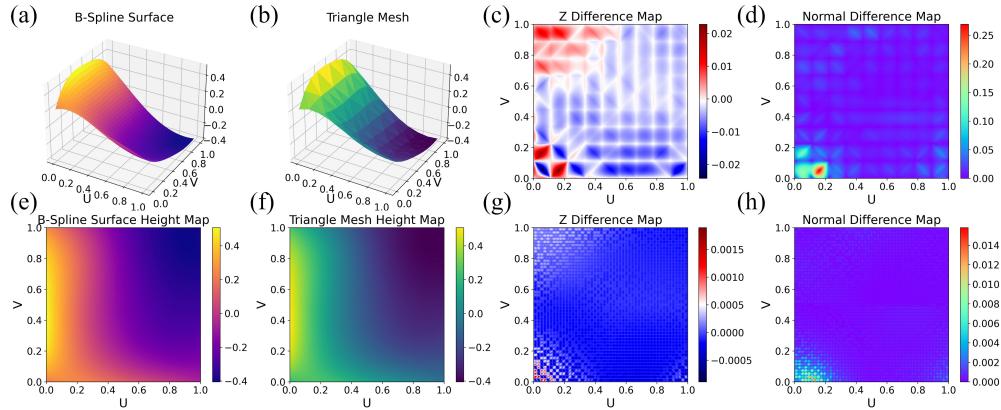


Fig. 3. Deviations between a B-spline facet and its triangulated meshes with different resolution. In (a)-(d), we present a B-spline surface and a 10×10 resolution triangle mesh along with its z -coordinate and normal difference maps. The mean absolute error of the z -coordinate is 3.510×10^{-3} , and the average L2 norm of the normals is 1.422×10^{-2} . In (e)-(h), we show the height map of the surface and a 50×50 triangulated mesh and their difference maps. The mean absolute error of the z -coordinate is 1.197×10^{-4} , and the average L2 norm of the normals is 4.965×10^{-4} .

Our differentiable ray tracing pipeline begins with ray generation from light sources, followed by a looped process driven by iterative ray tracing. At each loop, rays undergo accelerated intersection and differentiable evaluation, with the results accumulated until they reach the sensor. The loop is controlled by the *max_depth* parameter, which determines the maximum propagation depth for ray tracing. Intensity sensors collect simulated rays by mapping 3D ray directions to 2D photometric coordinates and accumulating their energy, measured in candela, onto pixels. We

use bilinear interpolation to distribute energy between the four neighboring pixels [24], ensuring the differentiability of this discrete process. The optimization drives the automatic differentiation process by minimizing the difference between the simulated intensity distribution and the target distribution, propagating gradients with respect to the scene parameters.

Algorithm 1. Accelerated differentiable ray tracing

```

1: procedure RAYTRACING(input_rays, max_depth, AS, surfaces, sensors)
2:   current_rays  $\leftarrow$  input_rays
3:   for d  $\leftarrow 0$  to max_depth do
4:     if current_rays.empty() then break
5:     end if
6:     ts, is, missing_rays  $\leftarrow$  AS.intersect(current_rays)     $\triangleright t_s$ : distances, is: indices
7:     next_rays  $\leftarrow$  []
8:     for each surface in surfaces do
9:       masked_rays  $\leftarrow$  surface.mask_selected(current_rays, is)
10:      uvs  $\leftarrow$  surface.world2local(masked_rays.parameter(ts).xy)
11:      surface_records  $\leftarrow$  surface.evaluate(uvs)     $\triangleright$  Differentiable evaluation
12:      next_rays.append(surface.sample_rays(surface_records, masked_rays))
13:    end for
14:    Collect missing rays to intensity sensor
15:    current_rays  $\leftarrow$  next_rays
16:  end for
17: end procedure

```

2.3. Neural process for regulation-aware intensity distribution

Automotive headlamp regulations impose complex energy constraints at specific test points and zones. Our goal is to use publicly available regulation data to create suitable light intensity distributions within a photometric range. Essentially, this task is a data generation problem aimed at producing a desired $z = I(u, v)$ function within the domain through a training process. Mathematical tools like thin-plate smoothing splines [37] can generate such distributions by interpolating and fitting discrete data points but struggle with complex inequality constraints, which makes it difficult to generate ideal results. Therefore, we leverage the neural network's data embedding capabilities and train a robust MLP under the constraints of the regulations.

Mathematically, our MLP is used to approximate the intensity distribution function $I(u, v)$ in the photometry domain Ω , which maps the angular coordinates (u, v) to non-negative intensity values. We design the MLP with an input layer that takes a 2D coordinate, two hidden layers with ReLU activation functions, and an output layer that produces a scalar intensity value. We can express the forward propagation of the MLP as:

$$I(u, v) = W_3 \cdot \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 \cdot [u, v]^T + b_1) + b_2) + b_3, \quad (3)$$

where W_i and b_i are the weights and biases of the *i*-th layer.

The MLP is trained by minimizing a loss function that combines multiple constraints from the regulation. These include test point constraints, test point group constraints, test zone constraints, and dynamic point constraints, as shown in Fig. 4. We formulate the inequality energy constraints into a unified loss function expression in Eq. (4):

$$\mathcal{L}_k = \frac{\sum_{i \in C_k} \text{ReLU}(\min_value_{k,i} - I_i) + \sum_{i \in C_k} \text{ReLU}(I_i - \max_value_{k,i})}{N_k}, \quad (4)$$

where k indexes the types of constraint ($k \in \{\text{points, groups, zones, global}\}$), C_k is the set of elements for constraint k , I_i is the intensity value at test point i , or the sum of intensities for a

group, or any point in a zone, $\text{min_value}_{k,i}$ and $\text{max_value}_{k,i}$ are the minimum and maximum limits for each element i , N_k is the total number of elements for constraint k . Then, we describe each constraint type as follows:

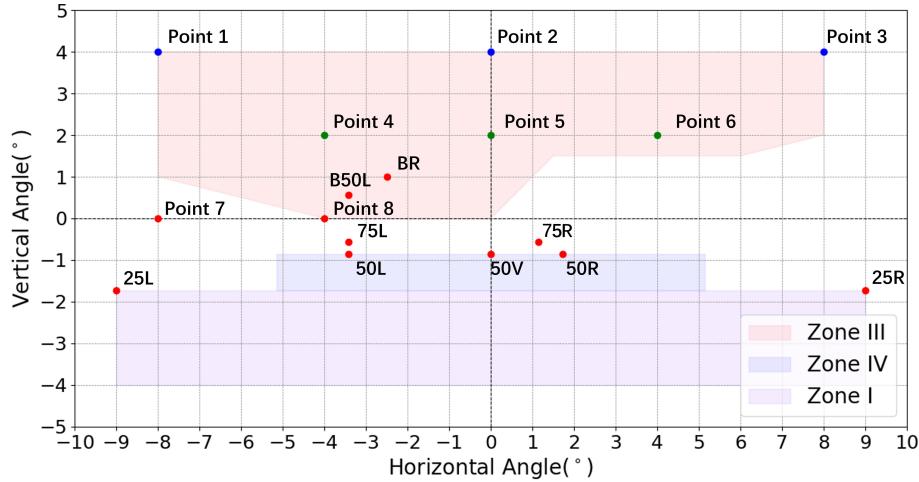


Fig. 4. ECE-R112 [1] regulation plot for passing-beam headlamp. This plot shows test points (50V, 75R, and other red dot points), test points groups (Points 1–3 and Points 4–6), and test zones (Zone I, III, and IV).

Test Points and Groups Constraints: Test points are critical for evaluating headlamp performance, ensuring compliance with regulations on luminous intensity at specific locations. Groups of test points assess overall performance by summing the intensities at multiple points. In addition to static constraints, regulations impose dynamic constraints, which involve intensity limits that vary depending on location. For instance, in ECE R112 class B passing-beam regulations, any point within Zone I must satisfy $I < 2 \cdot I_{50R}$. Driving-beam regulations restrict the maximum intensity across the domain to 215,000 cd. Ensuring compliance with these dynamic constraints requires a global search during the training and optimization.

Test Zones Constraints: Test zones are enclosed polygonal shapes within specific areas of the beam pattern. These zones require the luminous intensity at any point within them to fall within a specified range, ensuring compliance with regulation standards.

Global Constraints In addition to the above constraints, all intensity values within the global domain must be non-negative. This is enforced with an additional term in the loss function:

$$\mathcal{L}_{\text{global}} = \frac{\sum_{(u,v)} \text{ReLU}(-I(u, v))}{N_{\text{global}}}. \quad (5)$$

The total loss function combines all the above components:

$$\mathcal{L}_{\text{total}} = \sum_k \lambda_k \mathcal{L}_k, \quad (6)$$

where λ_k are weighting coefficients for each loss component.

In our implementation, the photometric range defined by specific regulations is discretized into a pixel grid. For the test points, we calculate the nearest pixels and directly apply the regulation constraints. For the test zones, we first mask the pixels located inside the zone, and then include all these pixel values in the loss function. The ReLU-based loss function drives the training loss to zero, which guarantees both the physical correctness and regulatory compliance of the generated distribution.

3. Results

We conducted experiments using PyTorch 2.4.0 with CUDA 11.8, which supports CPU and GPU tensor computation and differentiable optimization. We performed the experiments on a 12th Gen Intel Core i7-12700H 2.30 GHz with 32 GB RAM and an NVIDIA GeForce RTX 3060 Laptop GPU with 6 GB of device memory. First, we present the results of the neural process proposed in this paper, demonstrating that the designed loss function converges correctly and that the light intensity distributions generated by the MLPs meet the corresponding regulation requirements. Then, we provide three sets of reflector design results, illustrating that our differentiable ray tracing pipeline effectively optimizes the control point grids of B-spline surface arrays to approximate the target light intensity distribution. In addition, we have included schematic diagrams and regulation test plots of the designed optical systems, along with sensor results simulated using *LightTools 2022.03* [31] and detailed regulation test data reports in the [Supplement 1](#).

3.1. Neural process experiments

We present four groups of neural processes training and the corresponding regulation test results for the generated light intensity distributions based on ECE R112 regulations, including class A and class B, each containing passing-beam and driving-beam headlamps, as shown in Fig. 5. We present the detailed regulation test reports in tables in the [Supplement 1](#). The MLP in Eq. (3) contains 128 neurons in both the first and second hidden layers, which is sufficient for this task. We use $\lambda_{points} = 100$, $\lambda_{groups} = 50$, $\lambda_{zone} = 2$, and $\lambda_{global} = 1$ in the loss function in Eq. (6) to combine each loss with different importance. We set the resolution of the target distribution to 200×100 by dividing the angular range of $[-10^\circ, 10^\circ] \times [-5^\circ, 5^\circ]$ into intervals of 0.1° . The training process is optimized using the Adam optimizer with a learning rate of 0.05. Experimental results demonstrate that our MLP can optimize the loss to zero within a reasonable number of training iterations for four different regulations, achieving light intensity distributions that comply with regulation tests.

For passing-beam regulations training, class A requires approximately 2,000 iterations to converge. Class B requires around 4,000 training steps to converge due to dynamic point constraints, but it ultimately achieves clearer lighting characteristics at key points such as 50V and 75R, showing higher contrast and a sharper cut-off line compared to class A. For driving-beam regulations training, where only five test points are available, and one of them includes a dynamic maximum constraint, it is necessary to predefine a valid lighting range to prevent overfitting caused by the limited amount of training data. Practically, we define the range in various shapes, such as rectangular or elliptical, with intensity values peaking at the maximum and gradually decaying toward the boundary. Based on this predefined distribution, the subsequent training process ensures compliance with regulations.

3.2. Example reflector design

We designed an example reflector in *LightTools* to validate the effectiveness of our differentiable ray tracing system, with the process of implementing the example reflector in the software detailed in the [Supplement 1](#). We initialized a B-spline surface array with a grid size of $9mm \times 6mm$, consisting of three facets in the u -direction and two facets in the v -direction, with no gaps in either direction, a degree of three, and five control points in both directions. The z -coordinates of the control points grid are all initialized to 0. The point light source, positioned at $z = 2mm$ with rotation 180° around x -axis, has a power of 1 watt and emits uniformly within the upper hemisphere over a range of $[0^\circ, 90^\circ]$. The photometric range of the intensity sensor was $X \in [-90^\circ, 90^\circ]$ and $Y \in [-90^\circ, 90^\circ]$, with a resolution of 180×180 . We employed the Adam optimizer with a learning rate of 0.02, iterating for 100 epochs and using 10^6 rays per iteration. We use the L1 loss to measure the difference between the target distribution and the

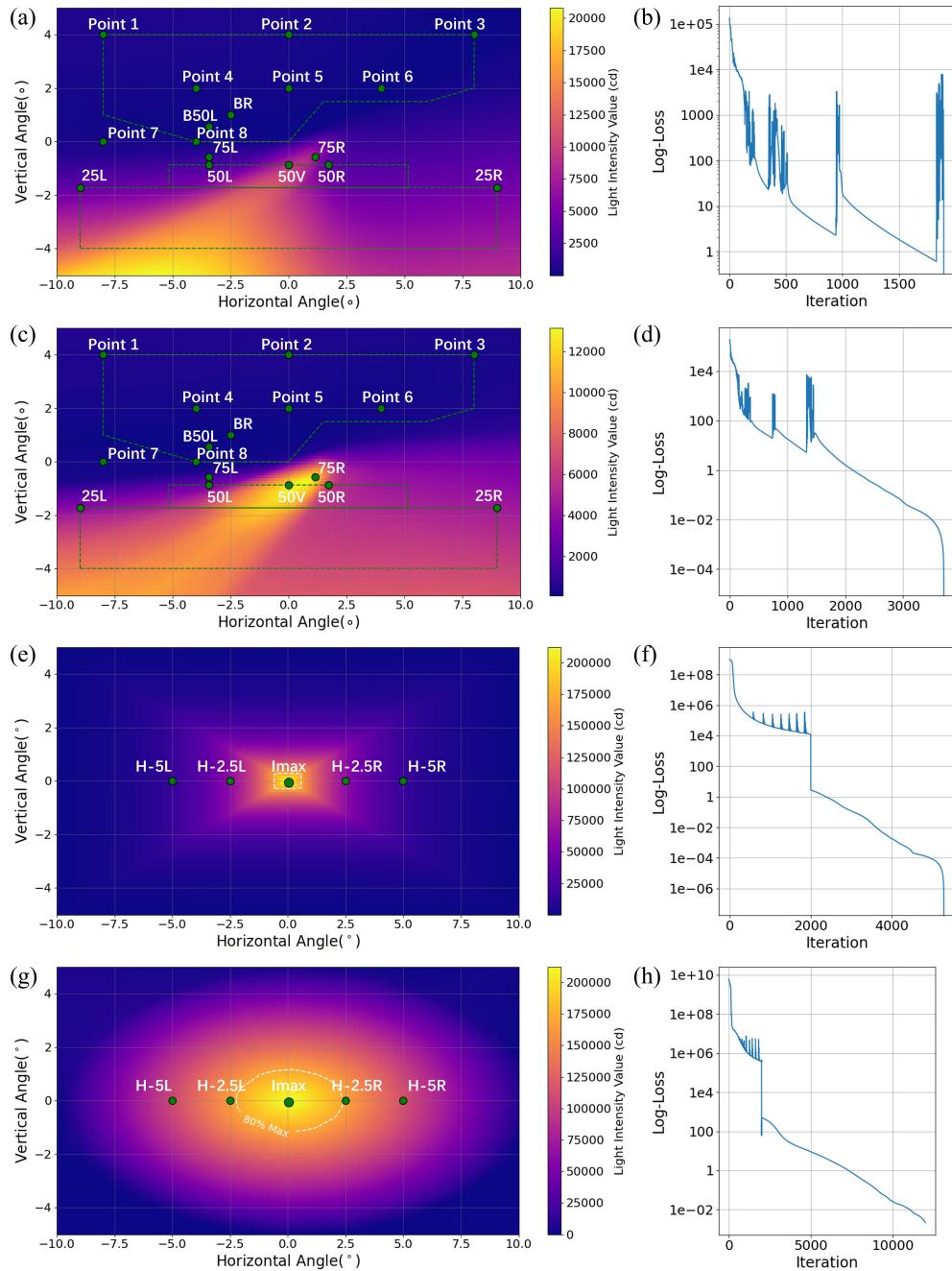


Fig. 5. Neural process experiments results for ECE R112 regulation. We present the generated light intensity distributions and training iteration-loss curves in log space of passing-beam class A(a-b), passing-beam class B(c-d), driving-beam class A(e-f), and driving-beam class B(g-h). The loss functions of Eq. (6) consistently decrease to zero, demonstrating the good generalization capability of the MLP in this task. The regulation test points, shown as green dots, and the test zones, outlined with green dashed lines, indicate that the results meet the regulation requirements.

simulation results to drive the optimization process. We present the simulation results of the training iterations, the comparison with commercial software simulations, the loss function curves, and schematic diagrams of the example reflector scene in Fig. 6.

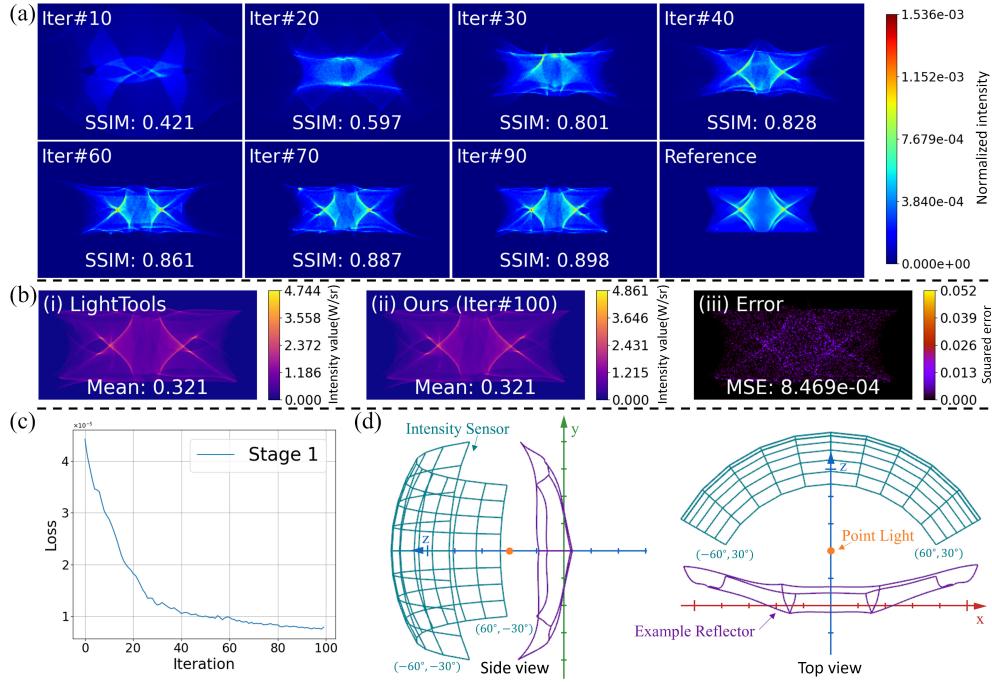


Fig. 6. Example reflector design results. The iteration images (a) are shown every 10 iterations in $X \in [-90^\circ, 90^\circ]$ and $Y \in [-60^\circ, 60^\circ]$. A statistic comparison (b) of mean value and MSE shows the simulation results obtained in $X \in [-60^\circ, 60^\circ]$ and $Y \in [-30^\circ, 30^\circ]$. We plot the L1 loss curve in (c) and views of the system in (d).

We use the structural similarity index (SSIM) to evaluate optimization performance, calculating it based on intensity values normalized by dividing each by the sum of all intensity values in the image. As shown in Fig. 6(a), the simulated light distribution continuously approaches the target pattern as the iterations progress, with the SSIM index monotonically increasing. This demonstrates the feasibility of optimizing the control points grid of the B-spline surface array. We compared the simulation accuracy and speed of our model in our software with those in LightTools, using 10^8 rays for simulation and a sensor resolution of 240×120 , as shown in Fig. 6(b). The designed reflector model was exported in STEP format using OpenCascade [38] and then imported into LightTools for simulation. The results showed similar mean values with a small mean squared error (MSE) of 8.4693×10^{-4} , while our program was 2.5 times faster than LightTools, completing the simulation in just 23 seconds compared to 58 seconds. These results demonstrate the advantages of our hybrid accelerated ray tracing and differentiable evaluation method, which not only significantly improves computational efficiency but also maintains high simulation accuracy.

3.3. Passing-beam reflectors design

For the passing-beam reflectors, we initialized a B-spline surface array with the size of $40\text{mm} \times 20\text{mm}$, divided into four facets along the u -direction and two facets along the v -direction. Each facet shares the same degree and number of control points as in the example. We positioned a $2\text{mm} \times 1\text{mm}$ rectangle light at $z = 20\text{mm}$ with a rotation 180° around x -axis, emits rays within

the upper hemisphere over $[0^\circ, 20^\circ]$. We optimized class A and class B passing-beam reflectors using a two-stage training strategy. In the first stage, we configured the photometric range of the intensity sensor to $X \in [-60^\circ, 60^\circ]$ and $Y \in [-40^\circ, 40^\circ]$ with a resolution of 300×200 . This broad range was crucial to initially capturing more valid reflected rays. In the second stage, we narrowed the range to $X \in [-15^\circ, 15^\circ]$ and $Y \in [-10^\circ, 10^\circ]$, maintaining the same resolution as the first stage. We use the Adam optimizer with learning rates of 0.02 and 0.005 for the first and second stages, respectively, simulating 10^6 rays in the first stage and 10^7 rays in the second stage. We iterated training for 100 epochs in each stage, optimizing with the L1 loss function to calculate the differences between the simulated results and the target distributions from the neural generated results in Fig. 5(a) and (c).

We provide a comprehensive display of class A passing-beam reflectors in Fig. 7(a)-(e), including the iteration distributions, regulation test results, loss function curves, and schematic diagrams. Similarly, for class B, these aspects are detailed in Fig. 7(f)-(j). As the training progresses, the increase in SSIM values shows that the pattern of the reflected rays is increasingly aligning with the desired light distribution. The regions of high intensity in the light pattern are aligned fairly accurately, and the outline shape of the passing-beam distribution and cut-off line closely matches the neural-generated distribution. In the regulation test plots, the intensity values of most points, groups, and zones conform to the specified energy range, verifying the effectiveness of our optimized design. Additionally, the loss curves show a significant reduction in the first stage, followed by a more gradual decline in the second stage, indicating that the learning rate settings are appropriate. We support checkpoints and resumption in our optimization pipeline, allowing us to dynamically adjust the learning rate and iteration rounds based on actual training performance, further enhancing design flexibility.

To further explore the capabilities of reflector optimization design, we configured light sources of various sizes ranging from $4mm \times 2mm$ to $32mm \times 8mm$ in the experimental setup for passing-beam class A, assuming a power of 1 watt per mm^2 for each light source. In the first row of Fig. 8, we set a uniform hemispherical emission range of 20° for the four cases. It can be observed that, with other parameters consistent, smaller light sources yield better optimization results. This is because smaller Étendue values result in more concentrated reflected rays, hereby enhancing the performance of the optimizer. We presented another group of results in the second row of Fig. 8, where we used smaller emission ranges for larger light sources, narrowing from 10° to 0.1° . We found that smaller emission ranges help larger light sources achieve better optimization results. Our simulations are based on an intensity sensor that only collects direction of rays, making the size of the light source and the reflector relative. Additionally, our system supports differentiable calculations of light power and emission range, facilitating the potential for joint optimization of light parameters and geometric parameters in optical designs.

3.4. Driving-beam reflectors design

For the driving-beam reflectors, we initialized a $40mm \times 40mm$ B-spline surface array with 4x4 facets, each with degree 3 and 5 control points in the u and v directions. The reflector was rotated 45° around the x -axis, and a $1mm \times 1mm$ rectangular light source was positioned at $y = -20mm$, rotated -90° around the x -axis to emit rays towards the reflector over the upper hemisphere range of $[0^\circ, 45^\circ]$. We applied the same two-stage optimization strategy as the passing-beam experiments. The first stage used an intensity sensor with a range of $X \in [-90^\circ, 90^\circ]$ and $Y \in [-90^\circ, 90^\circ]$ over 180×180 grids. The second stage narrowed the range to $X \in [-15^\circ, 15^\circ]$ and $Y \in [-10^\circ, 10^\circ]$ over 300×200 grids. The Adam learning rate was 0.02 for both stages, with 10^6 rays per iteration and 100 iterations per stage. As shown in Fig. 9(a), we selected specific iteration images from both stages and displayed them using the consistent photometric range of the second stage. Our optimization closely approximated the target pattern in Fig. 5(e) and (g) within 200 iterations, passing corresponding regulation tests.

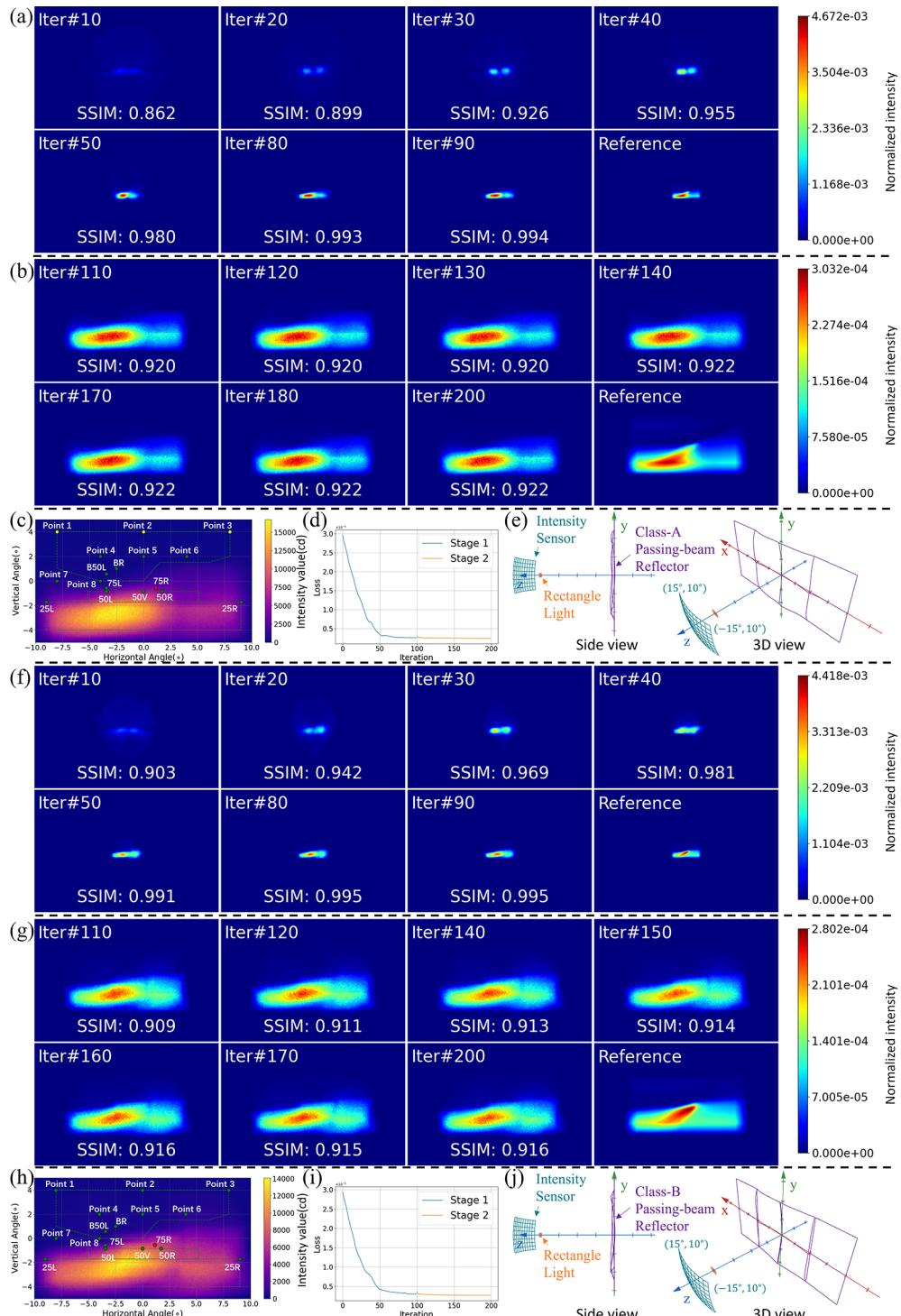


Fig. 7. Passing-beam reflector design results. We present the two-stage intermediate optimization results, regulation test plot, L1 loss function value curves, and simplified schematic diagrams of different views for class A (a)-(e) and class B (f)-(j).

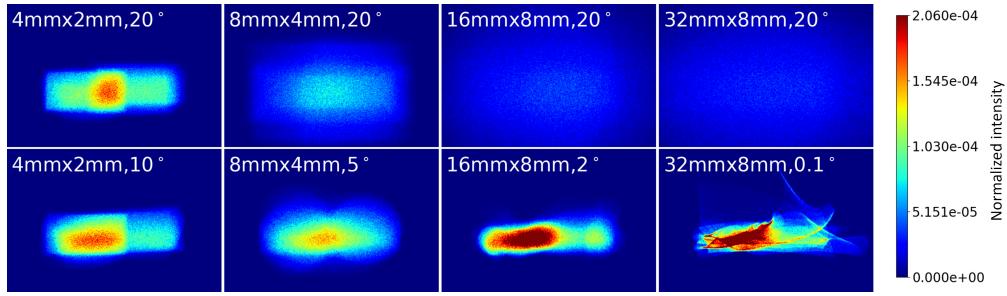


Fig. 8. Different light Étendue value optimizations under passing-beam class A distribution.

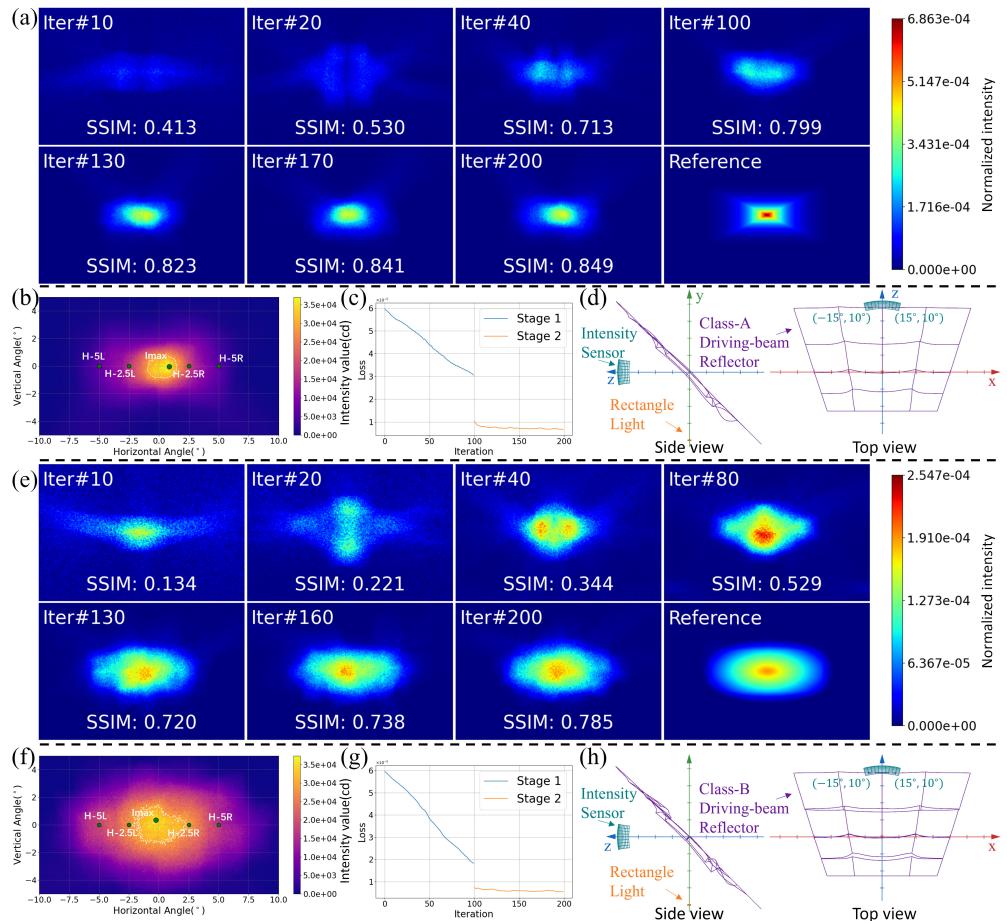


Fig. 9. Driving-beam reflector design results. We present the two-stage intermediate optimization results together, accompanied by a regulation test plot, training loss curves, and simplified schematic diagrams of different views for driving-beam reflectors class A (a)-(d) and class B (e)-(h).

4. Conclusion

We have proposed a technique for optimizing automotive headlamp reflectors through freeform surface arrays. Specifically, we develop a differentiable ray tracing pipeline and acceleration structures for intersection calculations. This approach has enabled automated gradient descent-based optimization of B-spline surface control point grids, further enhancing the modeling capabilities of differentiable freeform optical systems. Additionally, we introduce a feasible neural process targeting automotive headlamp regulations, training an MLP to generate light intensity distributions that strictly meet regulation inequality constraints. Experimental results have demonstrated that this method is fast-converging, accurate, and highly generalizable. Importantly, it can generate diverse light patterns for specified photometric ranges, enabling the automated optimization-based design of reflector shapes.

Future research can focus on improving the modeling representation of freeform surface arrays, including radially distributed freeform surfaces, and optimizing gap connection surfaces, enabling differentiable ray tracing systems to design more physically accurate reflector models. We can further utilize the gradient information from the optimization process to help calculate the manufacturing tolerances of the model, ensuring that the design meets manufacturing requirements. We also aim to explore more generalized headlamp designs, including refractive and projection types. We will also develop multi-objective optimization systems with higher simulation precision and faster speeds. We will develop these systems to tackle diverse application-specific challenges, optimize optical components, properties, and features comprehensively, and pave the way for more advanced and efficient optical system designs.

Funding. National Key Research and Development Program of China (No. 2024YDLN0011); Key Research and Development Program of Zhejiang Province (No. 2023C01039); National Natural Science Foundation of China (62322217); Research Grants Council of Hong Kong (ECS27212822, GRF17208023).

Acknowledgments. We thank the anonymous reviewers for their valuable comments and suggestions on this paper. And we also thank Bailin Deng from Cardiff University, Qi Wang from North China Electric Power University and Hanmo You from Tianjin University for their kindness help and constructive discussions.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Supplemental document. See [Supplement 1](#) for supporting content.

References

1. United Nations Economic Commission for Europe (UNECE), “UN Regulation No. 112: Uniform provisions concerning the approval of motor vehicle headlamps emitting an asymmetrical passing-beam or a driving-beam and equipped with filament, gas-discharge light sources or LED modules,” <https://unece.org/transport/documents/2023/09/standards/un-regulation-no-112-revision-4> (2023). Accessed: 2025-03-23.
2. H. Wang, X. Wang, Y. Li, *et al.*, “Design of a newly projected light-emitting diode low-beam headlamp based on microlenses,” *Appl. Opt.* **54**(7), 1794–1801 (2015).
3. H.-C. Chen, J.-H. Zhou, and Y. Zhou, “Stacking illumination of a confocal reflector light emitting diode automobile headlamp with an asymmetric triangular prism,” *Appl. Opt.* **56**(4), 1087–1093 (2017).
4. P. Ge, Y. Li, Z. Chen, *et al.*, “Led high-beam headlamp based on free-form microlenses,” *Appl. Opt.* **53**(24), 5570–5575 (2014).
5. H. Wang, X. Li, and P. Ge, “Design of an optical lens combined with a total internal reflection (tir) freeform surface for a led front fog lamp,” *Opt. Laser Technol.* **88**, 11–16 (2017).
6. M.-S. Tsai, C.-C. Sun, T.-H. Yang, *et al.*, “Robust optical design for high-contrast cut-off line in vehicle forward lighting,” *OSA Continuum* **2**(4), 1080–1088 (2019).
7. Y. Yang, D. Qiu, Y. Zeng, *et al.*, “Design of a reflective led automotive headlamp lighting system based on a free-form surface,” *Appl. Opt.* **60**(28), 8910–8914 (2021).
8. R. Wu, L. Xu, P. Liu, *et al.*, “Freeform illumination design: a nonlinear boundary problem for the elliptic monge–ampère equation,” *Opt. Lett.* **38**(2), 229–231 (2013).
9. N. Yadav, J. ten Thije Boonkkamp, and W. IJzerman, “Computation of double freeform optical surfaces using a monge–ampère solver: Application to beam shaping,” *Opt. Commun.* **439**, 251–259 (2019).
10. V. Oliker, “Controlling light with freeform multifocal lens designed with supporting quadric method(sqm),” *Opt. Express* **25**(4), A58–A72 (2017).

11. C. Bösel and H. Gross, "Ray mapping approach for the efficient design of continuous freeform surfaces," *Opt. Express* **24**(13), 14271–14282 (2016).
12. K. Desnijder, P. Hanselaer, and Y. Meuret, "Ray mapping method for off-axis and non-paraxial freeform illumination lens design," *Opt. Lett.* **44**(4), 771–774 (2019).
13. S. Wei, Z. Zhu, W. Li, *et al.*, "Compact freeform illumination optics design by deblurring the response of extended sources," *Opt. Lett.* **46**(11), 2770–2773 (2021).
14. Z. Zhu, S. Wei, Z. Fan, *et al.*, "Freeform illumination optics design for extended led sources through a localized surface control method," *Opt. Express* **30**(7), 11524–11535 (2022).
15. T.-M. Li, M. Aittala, F. Durand, *et al.*, "Differentiable monte carlo ray tracing through edge sampling," *ACM Trans. Graph.* **37**(6), 1–11 (2018).
16. M. Abadi, P. Barham, J. Chen, *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*, (USENIX Association, Savannah, GA, USA, 2016), pp. 265–283.
17. J. Ansel, E. Yang, H. He, *et al.*, "Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, (Association for Computing Machinery, New York, NY, USA, 2024), pp. 929–947.
18. Y. Nie, J. Zhang, R. Su, *et al.*, "Freeform optical system design with differentiable three-dimensional ray tracing and unsupervised learning," *Opt. Express* **31**(5), 7450–7465 (2023).
19. C. Wang, N. Chen, and W. Heidrich, "do: A differentiable engine for deep lens design of computational imaging systems," *IEEE Trans. Comput. Imaging* **8**, 905–916 (2022).
20. Q. Sun, C. Wang, Q. Fu, *et al.*, "End-to-end complex lens design with differentiate ray tracing," *ACM Trans. Graph.* **40**(4), 1–13 (2021).
21. Q. Liao, H. Wang, Z. Feng, *et al.*, "Differentiable design of freeform diffractive optical elements for beam shaping by representing phase distribution using multi-level b-splines," *Opt. Express* **32**(23), 41041–41056 (2024).
22. X. Yang, M. Souza, K. Wang, *et al.*, "End-to-end hybrid refractive-diffractive lens design with differentiable ray-wave model," in *SIGGRAPH Asia 2024 Conference Papers*, (ACM, 2024), pp. 1–11.
23. H. Wang, Y. Luo, H. Li, *et al.*, "Extended ray-mapping method based on differentiable ray-tracing for non-paraxial and off-axis freeform illumination lens design," *Opt. Express* **31**(19), 30066–30078 (2023).
24. H. Tang, H. Li, Z. Feng, *et al.*, "Differentiable design of a double-freeform lens with multi-level radial basis functions for extended source irradiance tailoring," *Optica* **11**(5), 653–664 (2024).
25. A. Heemels, B. de Koning, M. Möller, *et al.*, "Optimizing freeform lenses for extended sources with algorithmic differentiable ray tracing and truncated hierarchical b-splines," *Opt. Express* **32**(6), 9730–9746 (2024).
26. B. de Koning, A. Heemels, A. Adam, *et al.*, "Gradient descent-based freeform optics design for illumination using algorithmic differentiable non-sequential ray tracing," *ACM Trans. Graph.* **25**(3), 1203–1235 (2024).
27. Y. Huang, X. Zhang, G. Shi, *et al.*, "Optimizing algorithm for high-precision imaging stitching systems based on spline surfaces," *Opt. Express* **32**(21), 36093–36108 (2024).
28. H. Tang, Z. Feng, D. Cheng, *et al.*, "Parallel ray tracing through freeform lenses with nurbs surfaces," *Chin. Opt. Lett.* **21**(5), 052201 (2023).
29. M. Nimier-David, D. Vicini, T. Zeltner, *et al.*, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Trans. Graph.* **38**(6), 1–17 (2019).
30. W. Jakob, S. Speierer, N. Roussel, *et al.*, "Dr.jit: a just-in-time compiler for differentiable rendering," *ACM Trans. Graph.* **41**(4), 1–19 (2022).
31. Synopsys, Inc., "Illumination design software - lighttools," <https://www.synopsys.com/optical-solutions/lighttools.html> (2025). Accessed: 2025-04-10.
32. I. Ansys, "Ansys zemax opticstudio: Comprehensive optical design software," <https://www.ansys.com/products/optics/ansys-zemax-opticstudio> (2025).
33. I. Synopsys, "Code v: Advanced optical design software for imaging systems," <https://www.synopsys.com/optical-solutions/codev.html> (2025).
34. L. Piegl and W. Tiller, *The NURBS book (2nd ed.)* (Springer-Verlag, 1997).
35. G. Farin, "8 - b-spline curves," in *Curves and Surfaces for CAGD (Fifth Edition)*, G. Farin, ed. (Morgan Kaufmann, San Francisco, 2002), The Morgan Kaufmann Series in Computer Graphics, pp. 119–146, 5 edition ed.
36. I. Corporation, "Intel embree: High performance ray tracing," <https://www.embree.org/> (2024). Accessed: 2025-03-23.
37. MathWorks, "tpaps - thin-plate smoothing spline," <https://ww2.mathworks.cn/help/curvefit/tpaps.html> (2025). Accessed: 2025-04-01.
38. Open CASCADE SAS, "Open cascade technology | collaborative development portal," <https://dev.opencascade.org> (2025). Accessed: 2025-04-10.