Automated design of compound lenses with discrete-continuous optimization

ARJUN TEH, Carnegie Mellon University, USA
DELIO VICINI, Google, Switzerland
BERND BICKEL, Google, Switzerland
IOANNIS GKIOULEKAS*, Carnegie Mellon University, USA
MATTHEW O'TOOLE*, Carnegie Mellon University, USA

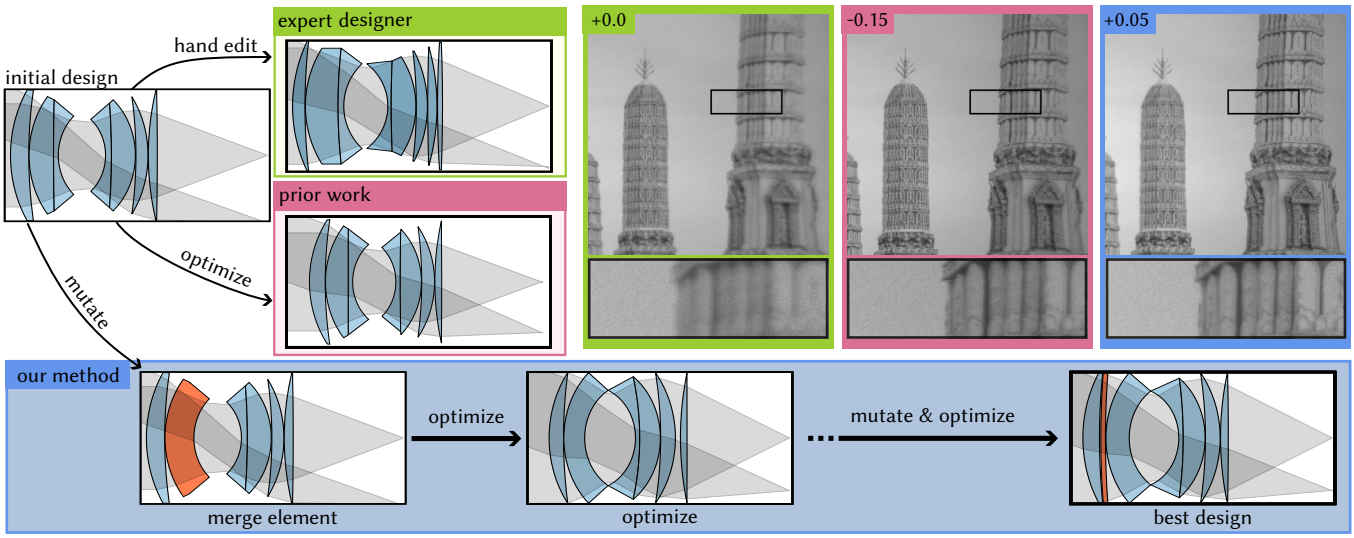

Figure 1. We develop a method that automatically explores the design space of compound lenses, by using Markov chain Monte Carlo sampling to combine gradient-based optimization with discrete changes to the number and type of lens elements. This combination allows our method to find designs that improve the sharpness and throughput of the initial lens design (in this example, the Nikon Nikkor-S 50mm $f/1.4$, released in 1962 [Reiley 2014]), even after it has been optimized by prior gradient-based methods [Teh et al. 2024]. Our method achieves image quality comparable to that of an improved lens designed by an expert (in this example, the Canon FD 50mm $f/1.2$, released in 1980). We report image brightness (top-left number of images) in terms of relative exposure.

We introduce a method that automatically and jointly updates both continuous and discrete parameters of a compound lens design, to improve its performance in terms of sharpness, speed, or both. Previous methods for compound lens design use gradient-based optimization to update continuous parameters (e.g., curvature of individual lens elements) of a given lens topology, requiring extensive expert intervention to realize topology changes. By contrast, our method can *additionally* optimize discrete parameters such as number and type (e.g., singlet or doublet) of lens elements. Our method achieves this capability by combining gradient-based optimization with a tailored Markov chain Monte Carlo sampling algorithm, using trans-dimensional mutation and paraxial projection operations for efficient global exploration. We show experimentally on a variety of lens design tasks that our method effectively explores an expanded design space of compound lenses, producing better designs than previous methods and pushing the envelope of speed-sharpness tradeoffs achievable by automated lens design.

CCS Concepts: • **Computing methodologies** → **Computational photography**; **Ray tracing**.

Additional Key Words and Phrases: lens design, Markov chain Monte Carlo, differentiable rendering, geometric optics

ACM Reference Format:
Arjun Teh, Delio Vicini, Bernd Bickel, Ioannis Gkioulekas, and Matthew O'Toole. 2025. Automated design of compound lenses with discrete-continuous optimization. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3757377.3763850

*indicates equal contribution.

Authors' Contact Information: Arjun Teh, ateh@andrew.cmu.edu, Carnegie Mellon University, Pittsburgh, PA, USA; Delio Vicini, vicini@google.com, Google, Switzerland; Bernd Bickel, berndbickel@google.com, Google, Switzerland; Ioannis Gkioulekas, igkioule@andrew.cmu.edu, Carnegie Mellon University, USA; Matthew O'Toole, motoole2@andrew.cmu.edu, Carnegie Mellon University, USA.

SA Conference Papers '25, Hong Kong, Hong Kong

ACM ISBN 979-8-4007-2137-3/2025/12
https://doi.org/10.1145/3757377.3763850

1 Introduction

Modern lens design demands optimizing increasingly sophisticated compound lenses to meet increasingly challenging performance

requirements. Though computational design tools exist, in practice these optimizations require close supervision by expert designers, who manually tune all aspects of the optical elements making up the compound lens. Obtaining good solutions requires tedious manual intervention, expert intuition, and trial and error.

Gradient-based optimization has become an essential part of the compound lens design process. Recent advances in differentiable rendering facilitate faster gradient-based optimization of continuous lens parameters (e.g., shapes of individual elements or distances between them) [Sun et al. 2021; Teh et al. 2024; Tseng et al. 2021; Wang et al. 2022]. Though such methods are invaluable for improving lens designs with a fixed *topology* (number and type of lens elements), they cannot optimize the lens topology itself. It is up to the expert designer to perform discrete changes manually—strategically adding, changing, or removing elements from a base design—before handing the design back to the optimizer for further improvement. Critically, changing the lens topology is often the *only* option available for meeting stringent performance requirements, for example: maintaining sharpness across the field of view when transitioning a lens from a half-frame to a full-frame sensor; increasing speed when designing a lens for extreme low-light conditions; or maintaining sharpness and speed when adapting a lens to smaller form factors.

Another way to assist expert designers is by generating good starting points. Methods using deep learning or genetic algorithms [Côté et al. 2021; Höschel and Lakshminarayanan 2018; Zoric et al. 2025] can help seed the design process, but the designs they output are often suboptimal: though they can make discrete changes to lens topology, they cannot effectively optimize continuous lens parameters, and thus cannot accurately assess the potential of different lens topologies. As a result, pushing the envelope in lens performance still requires expert designers to manually explore edits and iterate between discrete and continuous optimization.

These considerations highlight a critical gap in automated design of compound lenses: existing methods optimize only continuous or only discrete parameters of a system whose performance critically depends on *joint* optimization of both types of parameters. We address this gap in automated lens design capabilities by developing a method that performs mixed discrete and continuous lens optimization automatically. Our method uses *Markov chain Monte Carlo* (MCMC) sampling to combine gradient-based optimization of continuous lens parameters with *transdimensional mutations* that alter the number of lens elements (Figure 1). We make such a combination practical through two core contributions:

- A sampling algorithm that facilitates mixing gradient updates and mutations without sacrificing optimization performance.
- A set of mutations that use a projection operation to propose lenses with varied topologies that nevertheless remain paraxially equivalent to an original design.

These contributions enable our method to automatically explore a much larger design space of compound lenses than previously possible. We show experimentally that our method finds lenses with improved sharpness and speed compared to prior work that uses only gradient-based optimization. We provide interactive visualizations and an open-source implementation on the project website.[1]

[1]https://imaging.cs.cmu.edu/automated_lens_design

2 Related work

Geometric optics design. Lens designers commonly use computational tools [Synopsys 2023; Zemax 2023] to automatically optimize continuous lens parameters (e.g., curvature, size, and placement of lens elements). However, they typically have to rely on manual tuning and intuition to update the lens topology (e.g., number and type of lens elements) [Smith 2008]. Pedrotti et al. [2017] and Born and Wolf [2013] provide introductions to optics and lens design.

Differentiable rendering for lens design. Differentiable rendering enables solving inverse rendering problems using gradient-based optimization [Jakob et al. 2022; Li et al. 2018; Zhang et al. 2020]. Previous work focused on differentiating visibility discontinuities [Bangaru et al. 2022; Cai et al. 2022; Vicini et al. 2022; Wang et al. 2024; Zhou et al. 2024] and reducing memory usage [Teh et al. 2022; Vicini et al. 2021]. More recently, differentiable ray tracing methods have gained popularity for optimization of complex optical systems [Sun et al. 2021; Teh et al. 2022; Tseng et al. 2021; Wang et al. 2022], often end-to-end with post-processing algorithms [Côté et al. 2023; Yang et al. 2024]. Côté et al. [2023] also extend gradient-based methods to optimize discrete material selection. Compared to prior work, our method uses *aperture-aware differentiable ray tracing* [Teh et al. 2024] for gradient-based optimization, but explores a larger design space by also modifying the number and type of lens elements.

Lens design exploration. The design space of lenses is highly non-convex and contains local minima. Prior work uses simulated annealing [Zoric et al. 2024], genetic algorithms [Höschel and Lakshminarayanan 2018], or large language models [Zoric et al. 2025] to mitigate these issues. These methods can help find performant designs, but use a predetermined number of elements. To expand the search space, Betensky [1993] proposed exchanging parts of a design with paraxially equivalent designs—a strategy our method adopts for mutations. Alternatively, Sun et al. [2015] search over a library of off-the-shelf elements to generate designs without optimization. Côté et al. [2021] use deep learning to generate initial designs for subsequent optimization. Our method combines continuous optimization and discrete mutations, without requiring learning.

MCMC in computer graphics. Prior work in computational design often uses MCMC methods for discrete optimization problems. For example, Yeh et al. [2012] use reversible-jump MCMC to decide the placement of furniture in virtual rooms. Desai et al. [2018] use MCMC to optimize the placement of components of a mechanical assembly. More recently, Barda et al. [2023] use MCMC to optimize the design of sheet metal parts. MCMC also finds application in rendering to estimate light transport integrals [Veach and Guibas 1997]. Related to our work are methods that mix sampling distributions of varying dimensionality [Bitterli et al. 2017; Otsu et al. 2017], or use gradients to guide sampling [Li et al. 2015; Luan et al. 2020].

Quasi-stationary Monte Carlo. Metropolis-Hastings adjusted processes reject samples, which can be inefficient, especially when each sample is costly to generate (e.g., in optimization). Quasi-stationary Monte Carlo (QSMC) methods sample from a target distribution without rejections. Unlike Metropolis-Hastings, QSMC methods do

not require reversible transitions or detailed balance for convergence. Pollock et al. [2020] and Wang et al. [2021] proposed QSMC algorithms that sample from a target distribution using Brownian motion. In rendering, Holl et al. [2024] use the *jump RESTORE algorithm* [Wang et al. 2021] to mix global and local dynamics in Metropolis light transport. We use the jump RESTORE framework to develop a sampler that navigates the design space of lenses.

3 Problem setup

We focus on compound lens design under four assumptions:

1. We assume each compound lens is a collection of refractive singlets or cemented doublets (*lens elements*) that have spherical surfaces and are radially symmetric around the optical axis.
2. We do not include an aperture stop in the lens, thus the refractive elements determine the pupils and speed of the compound lens.
3. We use geometric optics and ignore wave effects (e.g., diffraction).
4. We assume sequential lens design, where rays entering the lens transmit through each refractive surface once and in order from lens entrance to sensor; we thus ignore geometric effects where rays violate this assumption (e.g., interreflections, glare).

With these assumptions, we represent a compound lens as a vector $\theta \in \mathbb{R}^N$ that includes four parameters for each refractive surface, ordered from lens entrance to sensor: curvature, lateral extent, distance to the next surface, and refractive index after the surface. Thus, a compound lens with K refractive surfaces has $N = 4K$ parameters.

We assume the compound lens is imaging a *target plane* to a *sensor plane*, both orthogonal to the optical axis at locations before and after the first and last (resp.) refractive surfaces. We simulate the image formation process using *sequential ray tracing*: Given a ray $\omega := (\mathrm{x}_0, \mathrm{v}_0)$ starting on the target plane at position $\mathrm{x}_0 \in \mathbb{R}^3$ and direction $\mathrm{v}_0 \in \mathcal{S}^2$, we propagate it through the lens with a sequence of refraction and propagation operations. Ray tracing ends when the ray either hits a stop (e.g., lens housing), or reaches the sensor plane at a position we denote $\mathrm{x}_\mathrm{s}(\omega;\theta) \in \mathbb{R}^3$ or $\mathrm{x}_\mathrm{s}(\mathrm{x}_0,\mathrm{v}_0;\theta) \in \mathbb{R}^3$, depending on context. We call rays that reach the sensor *valid rays* and denote their set $\Omega(\theta) \subset \mathbb{R}^3 \times \mathcal{S}^2$. The function $\mathrm{x}_\mathrm{s}(\cdot;\theta)$ is differentiable with respect to θ, and its value and derivatives can be computed efficiently through *primal* and *adjoint* (resp.) sequential ray tracing [Teh et al. 2024], which we use as part of our method.

Design objectives. We optimize θ for a combination of losses that emphasize sharpness, speed, and conformity to focal length specifications. We define each loss for rays starting from a target plane location x_0, then aggregate for many such locations (Equation (6)).

1. *Sharpness:* We use a variance-based *spot size loss*:

$$\mathcal{L}_\mathrm{spot}(\mathrm{x}_0;\theta) := \int_{(\mathrm{x}_0,\mathrm{v}_0)\in\Omega(\theta)} \|\mathrm{x}_\mathrm{s}(\mathrm{x}_0,\mathrm{v}_0;\theta) - \tilde{\mathrm{x}}(\mathrm{x}_0)\|^2 \,\mathrm{dv}_0, \quad (1)$$

$$\tilde{\mathrm{x}}(\mathrm{x}_0) := \frac{\int_{(\mathrm{x}_0,\mathrm{v}_0)\in\Omega(\theta)} \mathrm{x}_\mathrm{s}(\mathrm{x}_0,\mathrm{v}_0;\theta)\,\mathrm{dv}_0}{\int_{(\mathrm{x}_0,\mathrm{v}_0)\in\Omega(\theta)} \mathrm{dv}_0}. \quad (2)$$

2. *Speed:* We use a *throughput loss* inspired by Teh et al. [2024]:

$$\mathcal{L}_\mathrm{throughput}(\mathrm{x}_0;\theta) := 1 - \frac{1}{T_0}\int_{(\mathrm{x}_0,\mathrm{v}_0)\in\Omega(\theta)} \mathrm{dv}_0, \quad (3)$$

where T_0 is the maximum throughput determined by the maximum size of the design, which we specify as a hyperparameter.

3. *Focal length:* We use a loss based on the target focal length f,

$$\mathcal{L}_\mathrm{focal}(\mathrm{x}_0;\theta) := \|\tilde{\mathrm{x}}(\mathrm{x}_0) - \mathrm{x}_\mathrm{thin}(\mathrm{x}_0;f)\|^2, \quad (4)$$

where $\mathrm{x}_\mathrm{thin}(\mathrm{x}_0;f)$ is the sensor point predicted for x_0 from the Gaussian lens formula for a thin lens of focal length f.

We include an additional regularization term that prevents lens elements from becoming thinner than a hyperparameter d_min:

$$\mathcal{L}_\mathrm{thickness}(\theta) := \sum_{k=1}^{K} \max\left(d_\mathrm{min} - t_k, 0\right)^2, \quad (5)$$

where t_k is the thickness of the k-th lens element.

Our total loss is a weighted combination of these losses, aggregated over multiple target locations where appropriate:

$$\mathcal{L}(\theta) := \sum_{m=1}^{M}\big(w_\mathrm{spot}\mathcal{L}_\mathrm{spot}(\mathrm{x}_0^m;\theta) + w_\mathrm{throughput}\mathcal{L}_\mathrm{throughput}(\mathrm{x}_0^m;\theta) + w_\mathrm{focal}\mathcal{L}_\mathrm{focal}(\mathrm{x}_0^m;\theta)\big) + w_\mathrm{thickness}\mathcal{L}_\mathrm{thickness}(\theta). \quad (6)$$

3.1 Design with Markov chain Monte Carlo sampling

When the number K of lens elements (and thus length N of θ) is fixed a priori, the loss of Equation (6) is differentiable using adjoint sequential ray tracing [Teh et al. 2024]. Therefore, for fixed N, the design of a compound lens can be done using gradient-based optimization, an approach that has been popular in recent work [Sun et al. 2021; Teh et al. 2024; Tseng et al. 2021; Wang et al. 2022].

Unlike this prior work, we are interested in methods that design *all aspects* of a compound lens, including both the discrete-valued number K of elements and their continuous-valued parameters θ. Though K is not amenable to gradient-based optimization, we would like any such method to continue using gradient-based optimization for θ. We enable *mixed discrete-continuous design* by treating it as a sampling problem that we attack with *Markov chain Monte Carlo* (MCMC) algorithms. To this end, we first convert the loss in Equation (6) into a positive Boltzmann distribution to be *maximized*:

$$\pi(\theta) := e^{-\frac{\mathcal{L}(\theta)}{T}}, \quad (7)$$

where T is a *temperature* that controls the "sharpness" of the distribution: higher T allows exploring larger regions of the design space, whereas lower T forces exploration of only high-value regions. We treat T as a fixed hyperparameter, though future work could explore *simulated annealing* methods that progressively lower temperature.

Metropolis-Hastings and its pitfalls. The most popular MCMC method is the *Metropolis-Hastings* (MH) algorithm [Hastings 1970]. Given a lens θ, it uses a *proposal distribution* p to sample a lens proposal $\theta' \sim p(\theta \to \theta')$, and compute an acceptance probability:

$$\alpha := \min\left\{1,\ \exp\left(\frac{\mathcal{L}(\theta) - \mathcal{L}(\theta')}{T}\right)\cdot\frac{p(\theta'\to\theta)}{p(\theta\to\theta')}\right\}. \quad (8)$$

The proposal is randomly either accepted with probability α and used to update the lens ($\theta \leftarrow \theta'$), or rejected leaving the lens unchanged. The exponential term in Equation (8) favors accepting proposals θ' that improve (reduce) $\mathcal{L}$. Under certain conditions on p, this algorithm will sample lenses proportionally to $\mathcal{L}$.

The appeal of MH for mixed discrete-continuous design lies in the great flexibility it affords the user in selecting the proposal distribution p. We can use a *mixture* proposal distribution:

$$p(\theta \to \theta') = \beta p_{\text{continuous}}(\theta \to \theta') + (1-\beta) p_{\text{discrete}}(\theta \to \theta'), \quad (9)$$

which at random updates lens parameters θ either continuously by sampling $p_{\text{continuous}}$—a *perturbation* that leaves the dimension N the same—or discretely by sampling p_{discrete}—a *mutation* that changes N. Moreover, for $p_{\text{continuous}}$, we can use *Langevin perturbations* $\theta' \sim \mathcal{N}(\theta - \eta\, \mathrm{d}\mathcal{L}/\mathrm{d}\theta, \eta)$ [Roberts and Tweedie 1996], or equivalently:

$$\theta' \leftarrow \theta - \eta \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta} + \sqrt{\eta} \cdot \varepsilon, \quad (10)$$

where ε is a vector of normal random variates, and the gradient term can be computed using adjoint sequential ray tracing. The Langevin perturbations in Equation (10) mimic gradient descent, except for the addition of noise, allowing MH to incorporate gradient-based optimization of continuous lens parameters. In practice, it is common to use adaptive step sizes η in Equation (10) to improve sampling performance [Li et al. 2015; Luan et al. 2020], analogously to gradient descent algorithms such as Adam [Kingma and Ba 2017].

MH with mixed Langevin perturbations and discrete mutations has been successful elsewhere in graphics, notably in Monte Carlo rendering [Li et al. 2015; Luan et al. 2020]. Unfortunately, we have empirically found it difficult to use this approach for mixed discrete-continuous design of compound lenses, for two reasons:

1. The sharpness and speed of a compound lens θ with many elements (e.g., $K \geq 4$) is *very* sensitive to random perturbations. Langevin perturbations include noise ε (Equation (10)), and thus almost always produce badly performing lenses that are rejected. Preventing this behavior requires keeping the noise variance—and thus step size η—very small. The result in either case is that optimization requires prohibitively many sampling iterations.
2. Lens mutations that have high acceptance probability generally require refining a lens after increasing or decreasing its elements. Such post-mutation refinement requires continuous optimization, which makes it challenging—or impossible—to compute the reverse proposal distribution $p(\theta' \to \theta)$ in Equation (8).

We provide experimental evidence for both issues in Section 6. These issues arise from the requirement for *reversible proposals* in MH, which necessitates the addition of noise in Equation (10) and the computation of $p(\theta' \to \theta)$ in Equation (8). In Section 4, we address both issues by using a different MCMC algorithm that lifts the reversibility requirement alongside providing other advantages. Then, in Section 5, we design effective discrete lens mutations, leveraging the flexibility from no longer being constrained by reversibility.

4 The RESTORE algorithm

We propose to use an MCMC algorithm based on so-called *randomly exploring and stochastically regenerating* (RESTORE) processes [Wang et al. 2021]—we use the name RESTORE also for the algorithm associated with these processes. RESTORE uses two proposal distributions, one for small perturbations, and another for large changes. The RESTORE literature refers to these distributions as *local dynamics* and *regeneration probability* (resp.); for clarity, we continue to use the terms *perturbation* and *mutation* (resp.) from Section 3.1.

Algorithm 1 SingleRestoreStep($\theta, R, \gamma, \pi, C$)

Input: *A compound lens θ, a reservoir R, a weight parameter γ, a target distribution π, a constant C.*
Output: *A new compound lens $\tilde{\theta}$, an updated reservoir R.*

▷ *Perform a gradient step*
$\tilde{\theta} \leftarrow$ GradientStep(π, θ)
▷ *Compute the termination probability*
$\kappa \leftarrow \frac{\pi(\theta)-\pi(\tilde{\theta})+C}{\pi(\theta)+C}$
▷ *Check whether to terminate perturbation sequence*
if SampleUniform$[0,1] < \kappa$ **then**
  ▷ *Update the reservoir with the current lens*
  $R \leftarrow$ UpdateReservoir($R, \tilde{\theta}$)
  ▷ *Sample and mutate new lens*
  if SampleUniform$[0,1] < \gamma$ **then**
    $\tilde{\theta} \leftarrow$ SampleGlobal()
  else
    $\tilde{\theta} \leftarrow$ SampleReservoir(R)
  $\tilde{\theta} \leftarrow$ MutateLens($\tilde{\theta}$)
▷ *Return new lens and updated reservoir*
return $\tilde{\theta}, R$

Starting from an initial lens θ, RESTORE performs a sequence of perturbations updating θ ("local dynamics simulation"). At each perturbation, it uses θ to compute a *termination probability* to randomly decide whether to continue perturbations or terminate the sequence. Upon termination, it updates θ using the mutation distribution ("regeneration"), then restarts a perturbation sequence. Compared to MH, RESTORE has no rejections, and does not require reversible perturbation and mutation distributions—both differences that benefit optimization [Holl et al. 2024]. Instead of reversibility, RESTORE relies on careful selection of the mutation distribution and termination probability to ensure it samples the target distribution π [Wang et al. 2021]. We first detail our choices for the perturbation and mutation distributions, then explain how to determine the termination probability; along the way, we elaborate on advantages over MH. We summarize our version of RESTORE in Algorithm 1.

Perturbation distribution. As RESTORE does not require reversible proposals, we use gradient perturbations *without noise*:

$$\theta \leftarrow \theta - \eta \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\theta}. \quad (11)$$

Thus, a perturbation sequence becomes equivalent to gradient descent. The lack of noise is crucial for performance: Whereas in Equation (10) the noise term required keeping step size η small, in Equation (11) we can use large step sizes to accelerate optimization. We use Adam [Kingma and Ba 2017] to determine adaptive step sizes from the history of the current sequence. As we explain below, the sequence terminates randomly with a probability that adapts to how much progress gradient steps make towards improving θ.

Mutation distribution. RESTORE requires that the mutation distribution can sample any lens θ with non-zero probability. In practice, the choice of mutation distribution is further constrained by its role

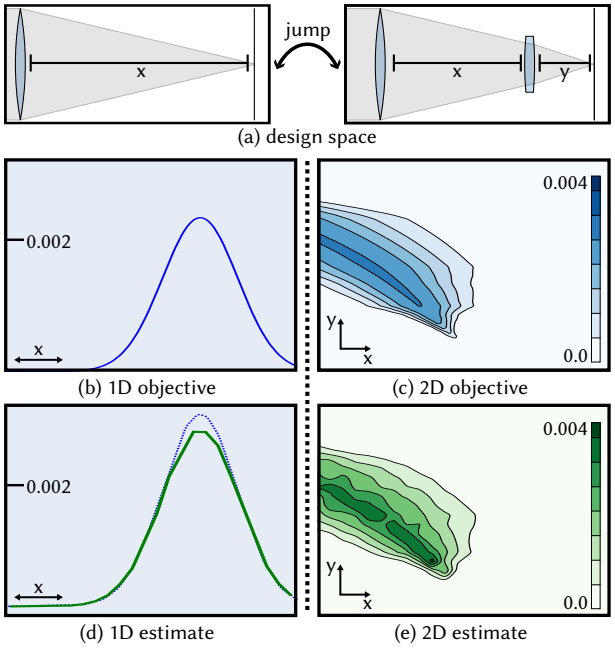

Figure 2. We construct a toy lens design problem (a) to validate the sampling accuracy of our method. The sampler can choose between optimizing the distance of a singlet from the sensor (1D target distribution, left), or the distances between two singlets and the sensor (2D target distribution, right). Even with an approximate termination probability, our method (d–e) correctly samples from the target distributions (b–c) in both 1D and 2D.

in determining the termination probability—through an integral operator relationship that is difficult to compute analytically except for trivial distributions (e.g., uniform) [Wang et al. 2021, Section 3]. We follow Pollock et al. [2020], who show that using a mutation distribution that selects from a reservoir of previous samples allows a tractable approximation of the termination probability.

In particular, we first sample a lens θ from a mixture distribution

$$\gamma p_{\text{global}}(\theta) + (1-\gamma) p_{\text{reservoir}}(\theta, R), \tag{12}$$

where: 1. p_{global} samples θ entries independently and uniformly within some upper and lower bounds; 2. $p_{\text{reservoir}}$ samples θ from a reservoir R of previously samples, as we explain below. We set the hyperparameter γ to a small value, to ensure that the mutation distribution satisfies the RESTORE requirement while mostly sampling from the reservoir. After sampling θ, we mutate it as in Section 5.

We populate R by storing the top N lenses (in terms of π) sampled upon perturbation chain terminations. The reservoir effectively allows "backtracking" to a previous high-performing lens when a perturbation chain gets trapped exploring a bad region of the design space. Without this backtracking, the mutation distribution would be extremely unlikely to find a reasonable lens by randomly sampling the design space. Using the reservoir allows "warm starting" a new perturbation chain from a previous reasonable lens.

Termination probability. Pollock et al. [2020] show that using gradient steps for perturbations and reservoir sampling for mutations

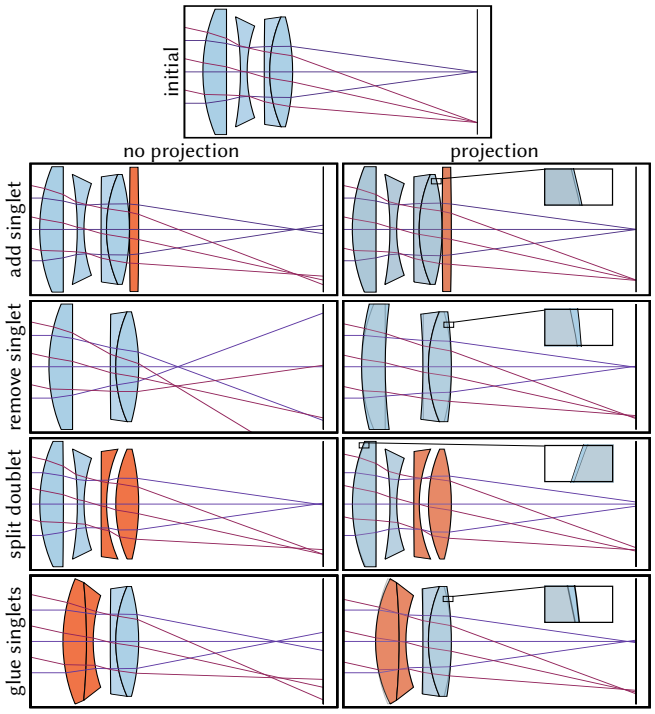

Figure 3. Left column: We consider four types of mutations of a compound lens. Applying such a mutation often leads to a lens with significantly worse performance than the original (diverging rays on the sensor plane). Right column: We alleviate this issue by using a *paraxial projection* refinement process that updates the mutated lens to have the same first-order focusing behavior as the original lens. For visualization, we overlay the original lens in grey over the refined one. Even though paraxial projection makes only small changes to lens elements, it greatly improves performance (converging rays on the sensor plane), while being very efficient to run.

simplifies the termination probability to:

$$\kappa \leftarrow \frac{\pi(\theta) - \pi(\tilde{\theta}) + C}{\pi(\theta) + C}, \tag{13}$$

where $C > 0$ is a hyperparameter. Intuitively, κ increases, and thus termination becomes more likely, when the relative improvement of π after a gradient step becomes small. Thus, RESTORE *adaptively* determines to stop perturbations and use a mutation when gradient optimization gets stuck at a local maximum of π. Though Equation (13) is an approximation, we show in Figure 2 that using it in RESTORE still allows sampling from the target distribution π.

5 Lens mutations

We implement four mutations (Figure 3) inspired by strategies in lens design textbooks [Smith 2008]: singlet addition, singlet removal, singlet gluing, and doublet splitting. Smith also suggests strategies for choosing an element and mutation (e.g., splitting high-curvature singlets). However, for simplicity we use uniform element and mutation sampling, which empirically performed comparably to sampling schemes derived from these strategies. Following the mutation, we refine the lens using a *paraxial projection* process, which we explain

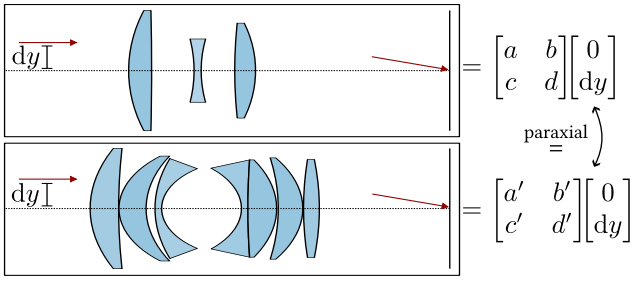

Figure 4. The 2×2 ray transfer matrix describes how a light ray parallel and close to the optical axis propagates through a compound lens. If two lenses have the same effect on such a ray, they are *paraxially equivalent*.

next using singlet addition as an example—the same process applies to the other mutations. We can perform this refinement because the overall mutation distribution does not need to be reversible.

If we simply add a singlet to the current compound lens, the resulting lens will almost certainly have drastically worse focusing and throughput performance than the original lens. Though RESTORE will still attempt to optimize the mutation proposal, optimization will likely get stuck fast, requiring a new mutation—and thus wasting the previous ones. We therefore need a way to quickly fine-tune the lens from a mutation proposal so that it performs closer to the original lens. We do so using *ray transfer matrix analysis*—a paraxial approximation to ray tracing—as a proxy for Equation (6). We emphasize that we use the paraxial approximation only for this refinement process: our overall method uses full ray tracing to design lenses while accounting for non-paraxial effects (Section 3).

Paraxial lens optics. With the paraxial approximation of radially symmetric optics, we parameterize a ray as the 2D vector of its distance y and angular deviation ϕ from the optical axis. We also model propagation of the ray through a sequential compound lens as matrix-vector multiplication with 2×2 *ray transfer matrix* M:

$$\begin{bmatrix} \phi_\mathrm{f} \\ y_\mathrm{f} \end{bmatrix} = M \begin{bmatrix} \phi \\ y \end{bmatrix}, \tag{14}$$

We overview this matrix for compound lenses comprising spherical elements, and refer to Pedrotti et al. [2017, Chapter 18] for details.

As a ray traveling through a lens undergoes a sequence of propagation and refraction operations, first we model each such operation as a ray transfer matrix. Propagation by a distance d and refraction at a spherical surface of curvature κ correspond to matrices:

$$M_\mathrm{p}(d) \coloneqq \begin{bmatrix} 1 & 0 \\ d & 1 \end{bmatrix}, \quad M_\mathrm{r}(\kappa, \eta_\mathrm{o}, \eta_\mathrm{i}) \coloneqq \begin{bmatrix} \eta_\mathrm{i}/\eta_\mathrm{o} & \kappa(\eta_\mathrm{i}-\eta_\mathrm{o})/\eta_\mathrm{o} \\ 0 & 1 \end{bmatrix}, \tag{15}$$

where η_i and η_o are the refractive indices before and after refraction.

We can then model a compound lens as the product of the refraction and propagation matrices corresponding to its elements. For example, a singlet is represented paraxially as $M_\mathrm{r} M_\mathrm{p} M_\mathrm{r} M_\mathrm{p}$, where the last matrix considers the distance from the final lens surface to the sensor plane. Given a compound lens θ, we denote its ray transfer matrix $M(\theta)$, which can always be written in the form:

$$M(\theta) = \begin{bmatrix} a & -1/f \\ b & c \end{bmatrix}, \tag{16}$$

where f is the focal length, and a, b, c are coefficients encoding other first-order properties of the compound lens. $M(\theta)$ provides a fixed-sized abstraction of the compound lens, no matter its complexity (e.g., number K of elements). Moreover, two compound lenses will have similar (up to first order) focusing behavior if their ray transfer matrices are equal, no matter how different their internal designs. We thus define the following notion of *approximate* equivalence between two compound lenses—emphasizing focusing of rays parallel to the optical axis and at infinite conjugacy ($y = 1, \phi = 0$).

Definition 1 : Paraxial equivalence

Two compound lenses θ_a and θ_b are *paraxially equivalent* if

$$M(\theta_a) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = M(\theta_b) \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Paraxial projection. After adding a singlet to a lens θ, we refine the augmented lens θ^* so that it is paraxially equivalent to θ. We formulate this refinement as a constrained optimization problem, which we solve using Newton's method and Lagrange multipliers:

$$\min_{\theta^*} \left\| \theta - \theta^* \right\|^2 \quad \text{s.t.} \quad \left(M(\theta) - M(\theta^*) \right) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0. \tag{17}$$

We keep entries of θ^* for the added singlet fixed. Empirically, we found that this process, which we term *paraxial projection*, results in much larger objective improvements than gradient descent with full ray tracing for equal runtime (30 iterations). The paraxial projection is locally unique and thus has a full-rank Jacobian of θ^* with respect to θ (Appendix A). This property, though not needed by our method, allows using paraxial projection in MH-based samplers requiring reversibility, and we compare with such a sampler in Section 6.

Comparison with automatic design search. Commercial software [Dilworth 2025] includes *automatic design search* procedures, which mutate compound lens designs through *automatic lens insertion* (AEI) and *automatic lens delete* (AED) methods, similar in spirit to our singlet-addition and singlet-removal mutations (resp.). AEI randomly adds singlets that initially have near-zero thickness, so that they do not impact the design. In our experiments, we found that such singlets remained thin even after several gradient iterations, and mitigating this issue required using expensive line search methods and a large weight on the thickness penalty loss (Equation (5)). AED removes elements by first attempting to make them have near-zero thickness through the addition of extra thickness penalties to the loss function. In our experiments, we found that AED was very sensitive to hyperparameter tuning, and significantly slowed down convergence. By contrast, our method supports a richer set of mutations that modify the design without causing gradient descent to get stuck or requiring dynamically changing the loss function.

6 Experiments

We show results from experiments on several lens design tasks. The project website includes interactive visualizations and more results.

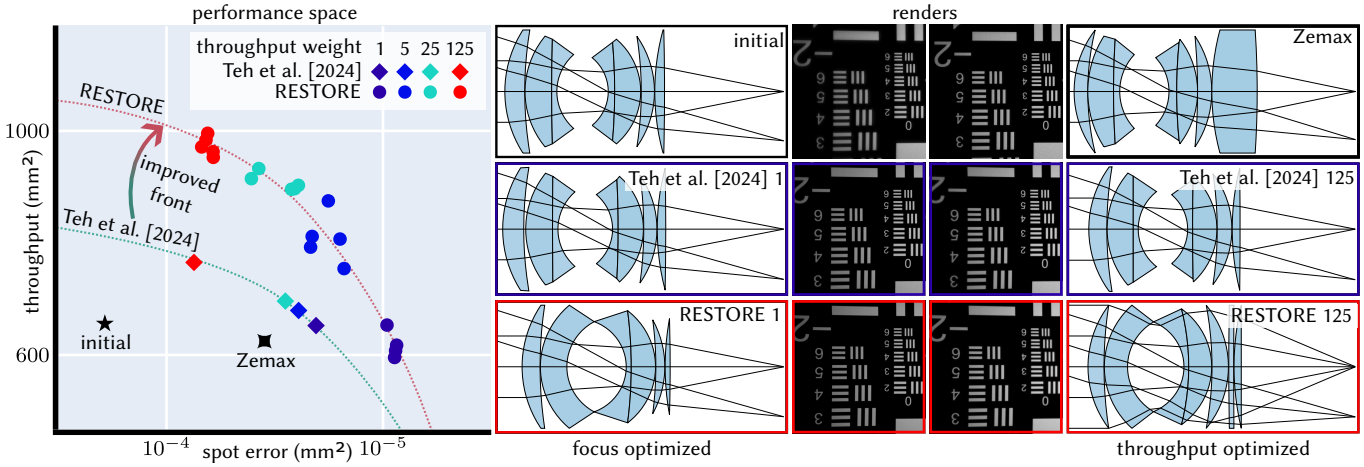

Figure 5. By varying the throughput and spot error weights in the optimization loss, we can explore designs that achieve different tradeoffs between lens speed and sharpness, tracing a Pareto front. Without mutations, lens designs are limited to the Pareto front determined by the initial design's topology. As our method can add and remove elements from the design, it is able to explore a larger space of designs and expand the Pareto front to achieve better tradeoffs.

Implementation details. We implemented our method in Python, using JAX [Bradbury et al. 2018] for GPU acceleration, and Optimistix [Rader et al. 2024] for paraxial projection. To compute derivatives of Equation (6), we implemented aperture-aware differentiable ray tracing [Teh et al. 2024], which was previously shown to improve lens optimization compared to alternatives [Wang et al. 2022]. We assume a 35 mm full-frame sensor, and optimize for four target plane positions in Equation (6), selected so that their thin-lens sensor-plane projections are uniformly distributed. We set the temperature T in Equation (7) so that $\pi(\theta_0) \approx 0.5$, the size of the reservoir R to 5 and $\gamma = 0.02$ in Equation (12), and $C = 2$ in Equation (13). Initial designs are from Reiley [2014]. On a workstation with an NVIDIA RTX 3090 GPU, our implementation performs 12,000 iterations in 40 min. We provide an open-source implementation of our method on the project website. When we use Zemax [2023], we optimize the default merit function for spot error and added objectives for focal length, running Zemax's damped least squares until convergence.

Expanding the lens sharpness-speed Pareto front. Figure 5 shows that our method allows exploring a larger tradeoff space of lens designs than using just gradient-based optimization or commercial tools [Zemax 2023]. Exploring the design space involves changing the weights of losses for sharpness versus speed in Equation (6), creating a Pareto front of lens designs. Initialized with lenses from the Pareto front produced using only gradient-based optimization [Teh et al. 2024], our method with the same total loss finds designs that move past this front to more favorable parts of the design space. By varying both continuous and discrete lens parameters, our method produces lenses with better overall performance.

Paraxial projection ablation. Figure 6 and Table 1 show results from an ablation study evaluating the utility of paraxial projection. We run our method with and without paraxial projection for 5000 iterations. The table shows that our method samples lenses that improve on the initialization much more often with paraxial projection than without. Figure 6 shows the distribution of mutations during sampling for the 135 mm lens, with and without paraxial projection. In both cases, there is a "warm-up" period where our method must first populate the reservoir with good lenses, before it starts consistently finding better lenses. Using paraxial projection greatly shortens this period, resulting in overall better lenses.

Table 1. Ablation study for paraxial projection. We run our method with and without paraxial projection, and report the fraction of iterations in which the sampled lenses have better objective value than the initial lens. We report results after 1000 and 5000 iterations.

	projection		no projection	
lens	1k	5k	1k	5k
wide-angle (28 mm)	0.356	0.358	0.225	0.241
normal (50 mm)	0.964	0.972	0.554	0.828
macro (105 mm)	0.118	0.424	0.0	0.015
telephoto (135 mm)	0.087	0.281	0.009	0.122

Comparison with Metropolis-Hastings. In Figure 7, we compare our method with a reversible-jump MCMC sampler that uses the Metropolis-adjusted Langevin algorithm [Roberts and Tweedie 1996]. As we explain in Section 3.1, the need to include noise in Langevin perturbations forces the MH-based sampler to use very small step sizes to ensure perturbations are accepted, slowing down optimization. The MH-based sampler also rejects most mutation proposals. By contrast, our method uses large step sizes and, as it has no rejections, attempts to optimize mutation proposals before another mutation. As a result, our method consistently finds better lenses.

Comparison with brute-force search. In Figure 8, we compare our method against a baseline using brute-force search—adding or removing a singlet at every possible location in the compound lens,

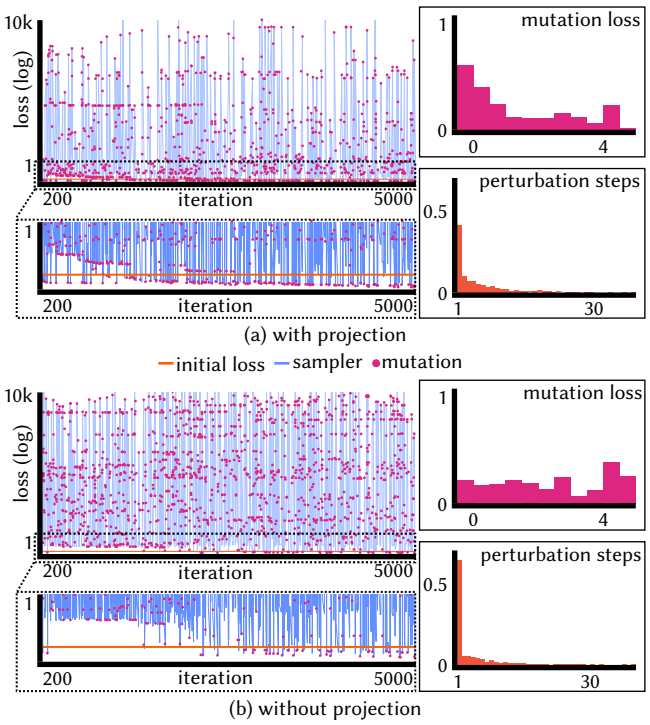

Figure 6. Two runs of our method with (top) and without (bottom) paraxial projection. We initialize both optimizations with the 135 mm lens and run for 5000 iterations. We also plot the loss of the initial lens (orange) for comparison. Using paraxial projection results in better mutations with lower losses, and perturbation sequences with longer durations. As the reservoir saturates with good lenses, both runs more consistently find better lenses. However, paraxial projection reduces this "warm-up" period.

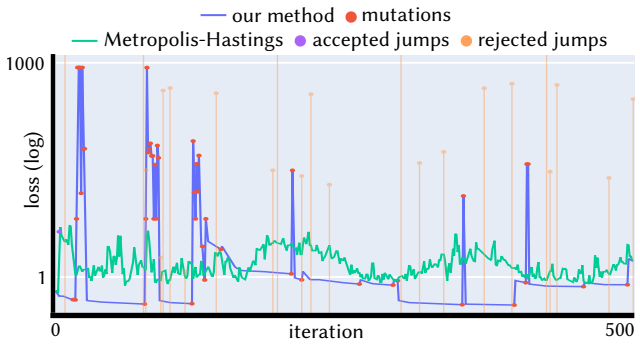

Figure 7. We compare our method with an MH-based sampler. The MH-based sampler (green) rejects most mutation proposals and can only perform small gradient steps. By contrast, our method (blue) efficiently optimizes all mutation proposals before trying another mutation.

then optimizing with gradient descent. Inserted singlet are sampled with random thickness (mean and standard deviation 1 mm) and curvatures (mean zero, standard deviation 0.01 mm^{-1}). This baseline is simple to implement, but is sensitive to local minima, and becomes intractable as we increase the number of elements in the original lens or the number of elements we want to add. By contrast, our method better avoids local minima and explores more lens designs.

We use four initial lenses: a 28 mm wide-angle lens, a 50 mm normal lens, a 105 mm macro lens, and a 135 mm telephoto lens. We run our method and the baseline for 40 min, optimizing lenses generated from additions and removals by the baseline for an equal number of gradient iterations. In all cases, our method finds lenses with better objective values, sharpness, and speed.

7 Conclusion and limitations

We presented a method for automated design of compound lenses that, uniquely among related work, can optimize both continuous and discrete parameters of a lens design. It uses MCMC sampling to combine gradient-based optimization of continuous parameters, and tailored mutations altering the number and type of lens elements. Our method uses the RESTORE algorithm for effective sampling, and paraxial projection to improve mutations. It finds better lenses than using only gradient-based optimization, and expands the Pareto front possible when considering the lens speed-sharpness tradeoff. We discuss limitations that point towards future research directions.

Initialization dependence. Though our method explores a larger design space than using only gradient-based optimization, it is still sensitive to initialization, requiring high-quality initial designs. Sampling such designs at random is extremely unlikely, but data-driven methods [Côté et al. 2021] provide a potential solution to this problem. Such methods can work in conjunction with ours to, e.g., prepopulate the reservoir. Alternatively, we can combine our method with curriculum learning, which has proven effective in mitigating initialization dependence for lens design tasks [Yang et al. 2024].

Manufacturing constraints and tolerances. Though our method outputs lenses with good simulated performance, ensuring that these lenses are manufacturable and robust to manufacturing errors is essential for practicality. For example, some designs (Figure 8) include very thin optical elements that can be either impossible to manufacture, or very unstable due to manufacturing tolerances. As a result, realizations of these designs may in practice underperform realizations of alternative designs that are worse in simulation but more reliable to manufacture. Incorporating manufacturing constraints and tolerances in the design process is possible to some extent through better engineering of the lens design objective (e.g., more strongly penalizing thin and high-curvature elements). However, given how sensitive the performance of a compound lens is to small changes in its elements (Figure 3), developing more principled solutions to these issues is an important future research direction.

Other design objectives. Our method optimizes objectives (Equation (6)) for a single wavelength and focal length. Lens design tasks often require considering large spectral ranges (achromats and apochromats that minimize chromatic aberrations) or focal length ranges (zoom lenses that remain sharp and fast with varied focal length). Our method can be adapted to such tasks by augmenting Equation (6) to average losses for different wavelengths and focal lengths, and allowing element positions to vary with focal length setting [Teh et al. 2024, Figure 10]. Gradient perturbations trivially extend to

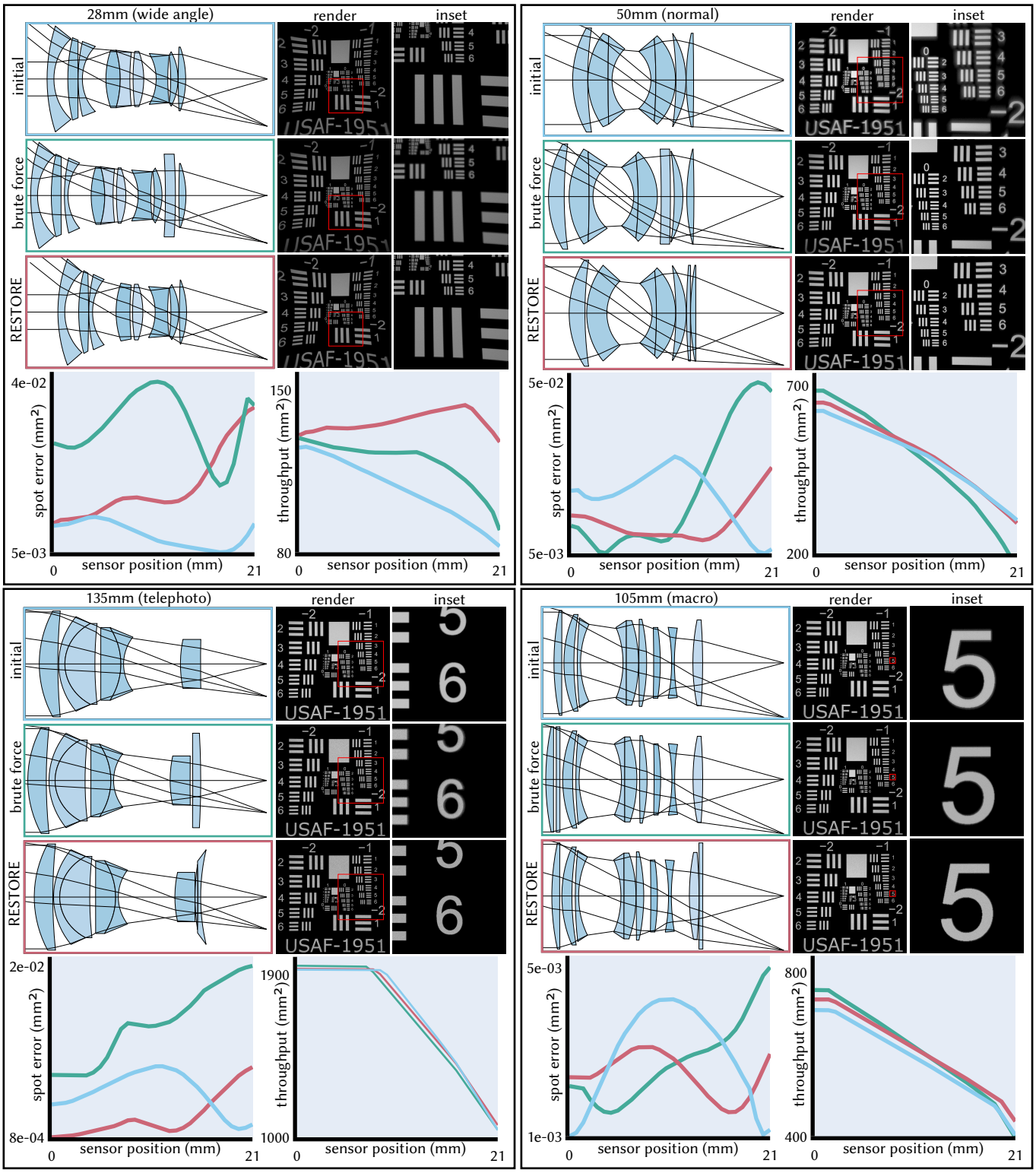

Figure 8. Comparison of our method with a baseline using brute force search. The baseline adds or removes an element at every possible location in the original compound lens, then uses gradient-based optimization to improve the resulting lens. Both methods run for 40 min. We experiment on four different lens types: a 28 mm wide-angle lens, a 50 mm normal lens, 105 mm macro lens, and a 135 mm telephoto lens. In all cases, our method finds better lenses than the baseline for the given objective function.

such augmented objectives, which remain differentiable. However, effective discrete optimization would require two non-trivial modifications to our method, which we leave as future research: 1. Adding mutations that change material (e.g., creation of achromatic doublets, insertion of low-dispersion singlets). 2. Modifying paraxial projection to account for multiple wavelengths or focal lengths.

Other geometric optical elements. Our method is limited to only spherical elements, due to our use of ray transfer matrices for paraxial projection. First-order approximations to other element types (e.g., aspherics, cylindrical lenses, or even reflective elements) exist but have lower accuracy than those for spherical elements. Thus it is unclear how effective they would be when designing lens mutations.

Wave optics. As we rely on geometric optics for ray tracing and paraxial projection, our method is limited to elements well approximated by geometric optics. Several recent works extend continuous lens design methods using gradient-based optimization to account for wave effects (e.g., diffraction) [Chen et al. 2023; Ho et al. 2024]. Incorporating these works into our method would allow likewise extending mixed continuous-discrete lens design.

Acknowledgments

We thank Andi Wang for discussions about Restore, and Ozan Cakmakci for discussions on commercial lens design tools. This work was supported by the National Science Foundation (awards 2047341 and 2238485), the Air Force Office of Scientific Research (FA 95502410244), Alfred P. Sloan Research Fellowship FG202013153 for Ioannis Gkioulekas, and a gift from Google Research.

A Jacobian of paraxial projection

We can compute the Jacobian of the solution θ^* of Equation (17) with respect to the pre-mutation parameters θ analytically, using the implicit function theorem. To this end, we first write the Lagrangian of the constrained optimization problem in Equation (17):

$$\mathcal{L}(\theta^*, \theta, \lambda) \coloneqq \|\theta - \theta^*\|^2 + \lambda^\top \left(M(\theta) - M\big(\theta^*\big)\right) \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{18}$$

From the method of Lagrange multipliers, the solution of Equation (17) is a stationary point of the Lagrangian, and thus satisfies the following system of equations:

$$\nabla_1 \mathcal{L} = 0, \tag{19}$$

$$\nabla_3 \mathcal{L} = 0, \tag{20}$$

where ∇_i refers to differentiation with respect to the i-th argument. Differentiating both sides of these equations with respect to θ,

$$\nabla_1\nabla_1\mathcal{L}\frac{\mathrm{d}\theta^*}{\mathrm{d}\theta} + \nabla_2\nabla_1\mathcal{L} + \nabla_3\nabla_1\mathcal{L}\frac{\mathrm{d}\lambda}{\mathrm{d}\theta} = 0, \tag{21}$$

$$\nabla_1\nabla_3\mathcal{L}\frac{\mathrm{d}\theta^*}{\mathrm{d}\theta} + \nabla_2\nabla_3\mathcal{L} + \nabla_3\nabla_3\mathcal{L}\frac{\mathrm{d}\lambda}{\mathrm{d}\theta} = 0. \tag{22}$$

$\nabla_1\mathcal{L}$ and $\nabla_2\mathcal{L}$ have the same dimensionality, whereas $\nabla_3\mathcal{L}$ has dimension equal to the number of constraints (two in Equation (18)). We can write Equations (21) and (22) in matrix form as:

$$\begin{bmatrix} \nabla_1\nabla_1\mathcal{L} & \nabla_3\nabla_1\mathcal{L} \\ \nabla_1\nabla_3\mathcal{L} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\theta^*}{\mathrm{d}\theta} \\ \frac{\mathrm{d}\lambda}{\mathrm{d}\theta} \end{bmatrix} = \begin{bmatrix} -\nabla_2\nabla_1\mathcal{L} \\ -\nabla_2\nabla_3\mathcal{L} \end{bmatrix}. \tag{23}$$

Then using Equation (18),

$$\begin{bmatrix} \mathbf{I} + \lambda^\top\nabla_1\nabla_1 g & \nabla_1 g \\ \nabla_1 g^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\theta^*}{\mathrm{d}\theta} \\ \frac{\mathrm{d}\lambda}{\mathrm{d}\theta} \end{bmatrix} = \begin{bmatrix} -\left(\mathbf{I} + \nabla_2\nabla_1 g\right) \\ -\nabla_2 g^\top \end{bmatrix}, \tag{24}$$

where g is the function describing the constraints in Equation (17):

$$g(\theta^*, \theta) \coloneqq \left(M(\theta) - M\big(\theta^*\big)\right) \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{25}$$

References

Sai Praveen Bangaru, Michael Gharbi, Fujun Luan, Tzu-Mao Li, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. 2022. Differentiable Rendering of Neural SDFs through Reparameterization. In *SIGGRAPH Asia 2022 Conference Papers*. Article 5, 9 pages. doi:10.1145/3550469.3555397

Amir Barda, Guy Tevet, Adriana Schulz, and Amit Haim Bermano. 2023. Generative Design of Sheet Metal Structures. *ACM Trans. Graph.* 42, 4, Article 116 (July 2023), 13 pages. doi:10.1145/3592444

Ellis I. Betensky. 1993. Postmodern lens design. *Optical Engineering* 32, 8 (1993), 1750 – 1756. doi:10.1117/12.145079

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. *ACM Trans. Graph.* 37, 1, Article 1 (Oct. 2017), 12 pages. doi:10.1145/3132704

Max Born and Emil Wolf. 2013. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light.* Elsevier.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/jax-ml/jax

Guanyuan Cai, Kai Yan, Zhao Dong, Ioannis Gkioulekas, and Shuang Zhao. 2022. Physics-Based Inverse Rendering using Combined Implicit and Explicit Geometries. *Computer Graphics Forum* 41, 4 (2022), 129–138. doi:10.1111/cgf.14592

Ni Chen, Congli Wang, and Wolfgang Heidrich. 2023. ∂H: Differentiable Holography. *Laser & Photonics Reviews* 17, 9 (2023), 2200828. doi:10.1002/lpor.202200828

Geoffroi Côté, Jean-François Lalonde, and Simon Thibault. 2021. Deep learning-enabled framework for automatic lens design starting point generation. *Opt. Express* 29, 3 (Feb 2021), 3841–3854. doi:10.1364/OE.401590

Geoffroi Côté, Fahim Mannan, Simon Thibault, Jean-François Lalonde, and Felix Heide. 2023. The Differentiable Lens: Compound Lens Search Over Glass Surfaces and Materials for Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20803–20812.

Ruta Desai, James McCann, and Stelian Coros. 2018. Assembly-aware Design of Printable Electromechanical Devices. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 457–472. doi:10.1145/3242587.3242655

Don Dilworth. 2025. Synopsys Lens Design Software. https://osdoptics.com

Wilfred K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (04 1970), 97–109. doi:10.1093/biomet/57.1.97

Chi-Jui Ho, Yash Belhe, Steve Rotenberg, Ravi Ramamoorthi, Tzu-Mao Li, and Nicholas Antipa. 2024. A Differentiable Wave Optics Model for End-to-End Computational Imaging System Optimization. arXiv:2412.09774 [cs.CV] https://arxiv.org/abs/2412.09774

Sascha Holl, Hans-Peter Seidel, and Gurprit Singh. 2024. Jump Restore Light Transport. arXiv:2409.07148 [cs.GR] https://arxiv.org/abs/2409.07148

Kaspar Höschel and Vasudevan Lakshminarayanan. 2018. Genetic algorithms for lens design: a review. *Journal of Optics* 48, 1 (Dec. 2018), 134–144. doi:10.1007/s12596-018-0497-3

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. DR.JIT: a just-in-time compiler for differentiable rendering. *ACM Trans. Graph.* 41, 4, Article 124 (July 2022), 19 pages. doi:10.1145/3528223.3530099

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] https://arxiv.org/abs/1412.6980

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6, Article 222 (Dec. 2018), 11 pages. doi:10.1145/3272127.3275109

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic Gaussian mutations for metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Trans. Graph.* 34, 6, Article 209 (Nov. 2015), 13 pages. doi:10.1145/2816795.2818084

Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020. Langevin monte carlo rendering with gradient-based adaptation. *ACM Trans. Graph.* 39, 4, Article 140 (Aug. 2020), 16 pages. doi:10.1145/3386569.3392382

Hisanari Otsu, Anton S. Kaplanyan, Johannes Hanika, Carsten Dachsbacher, and Toshiya Hachisuka. 2017. Fusing state spaces for markov chain Monte Carlo rendering. *ACM Trans. Graph.* 36, 4, Article 74 (July 2017), 10 pages. doi:10.1145/3072959.3073691

Frank L Pedrotti, Leno M Pedrotti, and Leno S Pedrotti. 2017. *Introduction to optics*. Cambridge University Press.

Murray Pollock, Paul Fearnhead, Adam M. Johansen, and Gareth O. Roberts. 2020. Quasi-Stationary Monte Carlo and The Scale Algorithm. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 82, 5 (10 2020), 1167–1221. doi:10.1111/rssb.12365

Jason Rader, Terry Lyons, and Patrick Kidger. 2024. Optimistix: modular optimisation in JAX and Equinox. arXiv:2402.09983 [math.OC] https://arxiv.org/abs/2402.09983

Daniel Reiley. 2014. Lens Designs. https://www.lens-designs.com/ Accessed: 2025-05-18.

Gareth O. Roberts and Richard L. Tweedie. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2, 4 (1996), 341 – 363.

Warren J Smith. 2008. *Modern optical engineering: the design of optical systems*. McGraw-Hill Education.

Libin Sun, Brian Guenter, Neel Joshi, Patrick Therien, and James Hays. 2015. Lens Factory: Automatic Lens Generation Using Off-the-shelf Components. arXiv:1506.08956 [cs.GR] https://arxiv.org/abs/1506.08956

Qilin Sun, Congli Wang, Qiang Fu, Xiong Dun, and Wolfgang Heidrich. 2021. End-to-end complex lens design with differentiate ray tracing. *ACM Trans. Graph.* 40, 4, Article 71 (jul 2021), 13 pages. doi:10.1145/3450626.3459674

Synopsys. 2023. Code V Optical Design Software. https://www.synopsys.com/optical-solutions/codev.html

Arjun Teh, Ioannis Gkioulekas, and Matthew O'Toole. 2024. Aperture-Aware Lens Design. In *ACM SIGGRAPH 2024 Conference Papers*. Article 117, 10 pages. doi:10.1145/3641519.3657398

Arjun Teh, Matthew O'Toole, and Ioannis Gkioulekas. 2022. Adjoint nonlinear ray tracing. *ACM Trans. Graph.* 41, 4, Article 126 (July 2022), 13 pages. doi:10.1145/3528223.3530077

Ethan Tseng, Ali Mosleh, Fahim Mannan, Karl St-Arnaud, Avinash Sharma, Yifan Peng, Alexander Braun, Derek Nowrouzezahrai, Jean-François Lalonde, and Felix Heide. 2021. Differentiable Compound Optics and Processing Pipeline Optimization for End-to-end Camera Design. *ACM Trans. Graph.* 40, 2, Article 18 (June 2021), 19 pages. doi:10.1145/3446791

Eric Veach and Leonidas J. Guibas. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 65–76. doi:10.1145/258734.258775

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path replay backpropagation: differentiating light paths using constant memory and linear time. *ACM Trans. Graph.* 40, 4, Article 108 (July 2021), 14 pages. doi:10.1145/3450626.3459804

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable signed distance function rendering. *ACM Trans. Graph.* 41, 4, Article 125 (July 2022), 18 pages. doi:10.1145/3528223.3530139

Andi Q. Wang, Murray Pollock, Gareth O. Roberts, and David Steinsaltz. 2021. Regeneration-enriched Markov processes with application to Monte Carlo. *The Annals of Applied Probability* 31, 2 (2021), 703 – 735. doi:10.1214/20-AAP1602

Congli Wang, Ni Chen, and Wolfgang Heidrich. 2022. dO: A Differentiable Engine for Deep Lens Design of Computational Imaging Systems. *IEEE Transactions on Computational Imaging* 8 (2022), 905–916. doi:10.1109/TCI.2022.3212837

Zichen Wang, Xi Deng, Ziyi Zhang, Wenzel Jakob, and Steve Marschner. 2024. A Simple Approach to Differentiable Rendering of SDFs. In *SIGGRAPH Asia 2024 Conference Papers*. Article 119, 11 pages. doi:10.1145/3680528.3687573

Xinge Yang, Qiang Fu, and Wolfgang Heidrich. 2024. Curriculum learning for ab initio deep learned refractive optics. *Nature Communications* 15, 1 (Aug. 2024). doi:10.1038/s41467-024-50835-7

Yi-Ting Yeh, Lingfeng Yang, Matthew Watson, Noah D. Goodman, and Pat Hanrahan. 2012. Synthesizing open worlds with constraints using locally annealed reversible jump MCMC. *ACM Trans. Graph.* 31, 4, Article 56 (July 2012), 11 pages. doi:10.1145/2185520.2185552

Zemax. 2023. OpticStudio. https://www.ansys.com/products/optics/ansys-zemax-opticstudio

Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. 2020. Path-space differentiable rendering. *ACM Trans. Graph.* 39, 4, Article 143 (Aug. 2020), 19 pages. doi:10.1145/3386569.3392383

Siwei Zhou, Youngha Chang, Nobuhiko Mukai, Hiroaki Santo, Fumio Okura, Yasuyuki Matsushita, and Shuang Zhao. 2024. Path-Space Differentiable Rendering of Implicit Surfaces. In *ACM SIGGRAPH 2024 Conference Papers*. doi:10.1145/3641519.3657473

Nenad Zoric, Marie-Anne Burcklen, Lijo Thomas, Momcilo Krunic, Yunfeng Nie, and Simon Thibault. 2025. Combined rule-based and generative artificial intelligence in the design of smartphone optics. *Opt. Continuum* 4, 9 (Sep 2025), 2233–2253. doi:10.1364/OPTCON.573424

Nenad Zoric, Yunfeng Nie, Simon Thibault, Radomir Prodanovic, and Lijo Thomas. 2024. Global search algorithms in an automated design of starting points for a deep-UV lithography objective. *Appl. Opt.* 63, 26 (Sep 2024), 6960–6968. doi:10.1364/AO.532057