LENSDESIGNER: A Self-Improving Agent for Optical Lens Design

Lei Sun[1,*,†] **Haoran Liang**[1,*] **Dannong Xu**[1] **Yao Gao**[2] **Yuyu Geng**[2]
Jinjin Gu[1] **Kaiwei Wang**[2,†] **Danda Pani Paudel**[1] **Luc Van Gool**[1]
[1]INSAIT, Sofia University "St. Kliment Ohridski"
[2]Zhejiang University
[*]Equal contribution. [†]Corresponding authors.
lei.sun@insait.ai wangkaiwei@zju.edu.cn

Abstract

Optical lens design is a complex, non-convex optimization challenge that relies heavily on human experience and intuition. Existing optimized-based automatic lens design methods struggle to navigate this vast parameter space without meticulous manual tuning. In this paper, we present LENSDESIGNER, an autonomous agent framework that mirrors the problem-solving workflow of expert optical engineers. To overcome the initial cold start problem, we construct LensLib100K, an extensive optical lens library, and employ Optics-Aware Retrieval to supply physically valid structural seeds. Within an interactive physical simulation environment, the agent executes macroscopic orchestration while receiving immediate optical feedback. Furthermore, we introduce a continuous self-evolving mechanism guided by a curriculum agent. By iteratively solving design tasks with progressively increasing difficulty, the agent autonomously extracts, accumulates, and reuses design heuristics, effectively evolving its optical lens design expertise over time. At the evaluation level, we introduce LensArena, a standardized evaluation benchmark comprising 120 diverse optical design tasks, covering extreme configurations. Extensive experiments on this benchmark demonstrate that LENSDESIGNER significantly outperforms publicly available baseline algorithms, achieving superior success rates and optimization efficiency. We hope this work sheds light on the emerging field of AI for optics. Project page: https://insait-institute.github.io/LensDesigner/.

1 Introduction

Optical lens design is a fundamental pillar of modern science and technology, underpinning advancements in diverse fields ranging from consumer electronics and augmented reality to medical imaging and space exploration. However, designing a high-quality lens system is a notoriously challenging, non-convex optimization problem involving a vast, high-dimensional parameter space (e.g., curvature, thickness, glass materials). Crucially, navigating this complex space is intrinsically a highly *experience-dependent* endeavor and heavily relies on human expert intuition. As show in Fig. 1, a typical design process requires an experienced optical engineer to propose an initial lens structure based on decades of accumulated domain knowledge, trial, and error, followed by iterative refinements using commercial optical simulation software like *Zemax* [2] or *CODE V*. Without this experience-driven high-quality initial starting point, traditional optimization algorithms easily fall into local optima, making the process highly inefficient.

Recently, artificial intelligence has shown great potential in accelerating scientific discovery. However, applying AI to optical design remains fundamentally bottlenecked by the limitations of current

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

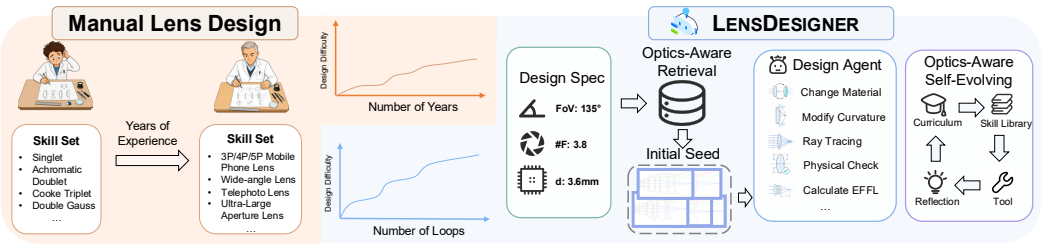

Figure 1: Manual lens design is a highly experience-dependent task that typically requires years of accumulated human expertise. Similarly, LENSDESIGNER mimics this trajectory through a self-evolving mechanism, continuously accumulating skills to advance its optical design capabilities.

methodologies. For instance, approaches utilizing traditional evolutionary algorithms for lens optimization [12, 13, 37] demand extensive fine-grained parameter tuning that is difficult for users lacking expert intuition. Moreover, these techniques suffer from prohibitive computational costs, frequently needing several days to successfully converge on a single lens design. Conversely, methods relying on simple database query and matching [9] fail to handle novel target specifications outside their existing scope, lacking the autonomous intelligence needed for original design.

Meanwhile, although Large Language Models (LLMs) are not inherently designed for precise numerical calculation or optimization, they encode rich optical domain knowledge. More importantly, recent advancements in autonomous agents have proven that LLMs can effectively utilize memory and skill modules to continuously learn, and self-evolve [27, 24, 32, 31, 1]. Therefore, while LLM agents possess immense potential to tackle such highly experience-dependent tasks, their application in automated optical lens design remains an unexplored frontier.

In this paper, we propose LENSDESIGNER, an autonomous agent framework that mirrors the workflow of human experts. Experienced optical engineers rarely design from scratch. Instead, they rely on known configurations as initial seeds. We replicate this crucial heuristic by constructing a large-scale optical lens library, dubbed LensLib100K, based on an enhanced implementation of the state-of-the-art optimization algorithm QGSO [12]. This comprehensive dataset encompasses a wide spectrum of common lens specifications. Leveraging this library, we design Optics-Aware Retrieval strictly grounded in fundamental optical design principles. For every new design requirement, this module retrieves the most suitable configuration from the library to serve as a physically valid initial seed.

Once the initial seeds are retrieved, LENSDESIGNER undertakes substantial macro orchestration within an interactive optical design environment. The agent autonomously executes actions, such as inserting new elements or altering glass materials, and receives immediate evaluation feedback from an underlying physical simulator regarding the optical merit function and structural validity.

To simulate the experience accumulation of a human optical engineer, our framework introduces a continuous evolution mechanism driven by a dedicated curriculum agent. This curriculum agent continuously generates physically valid optical design tasks with progressively increasing difficulty. As LENSDESIGNER attempts to solve these generated problems, it actively interacts with the environment and reflects on both successful and erroneous design trajectories based on the simulator feedback. By analyzing these outcomes, the LLM extracts generalized optical design heuristics and commits these validated strategies to an optical skill library for future reuse. This continuous cycle of problem solving, reflection, and memorization enables LENSDESIGNER to continuously evolve, transforming its approach from basic trial and error into mature design intuition. Finally, the framework invokes traditional local optimization algorithms to perform meticulous parameter tuning, guiding the newly synthesized lens to its precise local optimum.

Beyond these methodological advancements, the field of automated optical design currently lacks a standardized benchmark to fairly evaluate the capabilities of different algorithms. Hence, we introduce LensArena, a comprehensive benchmark comprising 120 diverse optical design tasks. This challenging benchmark systematically covers both common lens configurations and extreme optical systems, including ultra-wide-angle lenses with a Field of View (FoV) up to 180° and large-aperture lenses with an F-number (f/#) as small as 0.7, etc. Utilizing LensArena, we rigorously test all publicly available automatic lens design algorithms to establish a comprehensive evaluation standard. Extensive experiments on this benchmark demonstrate that our proposed LENSDESIGNER

significantly outperforms existing methods by consistently generating valid lens structures, achieving the highest success rate, and yielding the smallest (root-mean-square) RMS spot radius for optimal imaging quality.

In summary, we deliver the following contributions:

- We propose LENSDESIGNER, the first agent framework specifically designed to automate the highly empirical and complex process of optical lens design by mirroring human expert intuition.
- We introduce a continuous evolution mechanism driven by a curriculum agent, enabling the system to autonomously extract, accumulate, and reuse optical design heuristics, progressively advancing its lens design expertise through iterative problem solving.
- We construct LensLib100K, an extensive optical lens library, and introduce LensArena, a standardized evaluation benchmark, laying a robust foundation for future intelligent optical design research.

2 Related Work

Automatic Lens Design. Traditional optical design is highly empirical, relying on expert intuition and software tools like Zemax that function as numerical calculators rather than autonomous decision-makers. Existing automation attempts include global optimization algorithms [12, 19], neural networks [37], and retrieval-based methods like LensNet [9]. However, these approaches often require meticulous manual tuning, suffer from prolonged convergence times, or are constrained by fixed database distributions. In contrast, LENSDESIGNER enables rapid, text-driven generation of compliant configurations, bypassing the need for specialized domain expertise and intensive parameter adjustment.

LLM-based Agents for Scientific Design. Large Language Models (LLMs) such as GPT-5 and Qwen-3 have evolved into sophisticated reasoning engines capable of powering autonomous agents [25, 34]. By integrating tool utilization and environmental feedback, these agents have revolutionized specialized fields including chemistry [4], robotics [7], and software engineering [40]. Despite these successes, the application of autonomous entities to the highly empirical and numerically sensitive domain of optical design remains largely unexplored. We bridge this gap by introducing a self-evolving agentic framework specifically optimized for the physics-heavy requirements of lens design. For more detailed related work, please kindly refer to Appendix B.

3 LENSDESIGNER

In this work, we focus on classical imaging lens design, whose goal is to arrange and optimize the shape, spacing, and materials of refractive components to minimize aberrations and project a sharp image onto a sensor. This optimization process ensures the system meets strict imaging specifications, such as field-of-view (FoV), F-number (f/#), and effective focal length (EFFL). As a pioneering effort in this domain, our work focuses exclusively on spherical lenses.

3.1 Lens Design Preliminaries

A standard system comprises four fundamental structures: the object plane originating the target scene, the optical elements sequentially refracting incoming light, the stops restricting light bundles to control brightness, and the image plane capturing the final picture.

In a lens system, the Field of View (FoV) defines the maximum angular extent of the observable environment. The F number (f/#) characterizes the size of the optical aperture, representing the ratio of the focal length to the entrance pupil diameter and dictating the overall light-gathering capability. The optical format (d) denotes the physical diagonal diameter of the image sensor, while its corresponding semi-image height y defines the maximum radial boundary of focused light. Furthermore, the Effective Focal Length (EFFL) represents the focal length of the optical system, and the Total Track (TOTR) indicates the overall physical length of the entire lens assembly. These configurations fundamentally determine the physical characteristics and practical applications of the

lens. Finally, the Root Mean Square (RMS) of the focal spot radius serves as the most fundamental quantitative indicator for reflecting the actual imaging sharpness and evaluating overall optical aberrations. Further theoretical details are provided in Appendix A.

3.2 Overview

Inspired by the empirical workflow of human optical engineers, LENSDESIGNER operates as a closed-loop autonomous system that integrates knowledge retrieval, interactive reasoning, and self-evolution. The overall pipeline is illustrated in Fig. 3.

The Optics-Aware Retrieval identifies physically compatible initial seeds from our LensLib100K to ensure a good starting point. Then, the framework utilizes Macro Orchestration to coordinate an Optical Tool Suite, which executes structural modifications and local optimizations under strict physical guardrails. By continuously reflecting on and accumulating expertise throughout the design process, the system achieves a closed-loop evolution of design intuition.

3.3 LensLib100K

Large-Scale Lens Generation. Considering the highly free and empirical process of optical design, designing a large number of lenses manually is practically infeasible. Although current automated optical design methods remain imperfect, they can produce a substantial volume of satisfactory designs under simpler conditions, such as narrow FoV and large f/#, provided they are guided by manual parameter tuning over several months. QGSO utilizes a genetic algorithm-based framework [17] to optimize a population of randomly initialized lens structures through iterative mutation and selection. By integrating hybrid global and local optimization to produce diverse, high-quality designs, it currently represents the state-of-the-art in optimization-based lens design.

Building upon the QGSO framework, we introduce two enhancements to bridge the gap between simulation-based optimization and practical optical engineering: First, we enforce the use of real glass catalogs instead of virtual materials, ensuring the optimized parameters are practically manufacturable. Second, we integrate a ray aiming [10] mechanism to guarantee high-fidelity ray tracing under large FoVs. Overall, these improvements enhance the manufacturability of the design outcomes, which makes the produced lens more applicable.

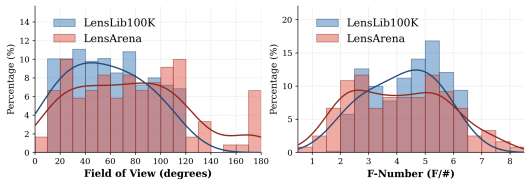

Figure 2: Distribution of LensLib100K and LensArena. LensArena spans more challenging edge cases.

Data Distribution Leveraging our extended QGSO framework, we invested approximately 2000 GPU hours of intensive computation to generate a comprehensive collection of 180000 distinct lens designs. These configurations cover a range of optical specifications. Specifically, the FoV spans from 8° to 118°, and the f/# ranges from 1.6 to 6.0. Data distribution is shown in Fig. 2. To the best of our knowledge, our LensLib100K represents the most extensive optical lens database currently available to the research community. It not only serves as a practical foundation for direct structural retrieval within our LENSDESIGNER framework but also offers a substantial collection of high-quality training data to support future research in data-driven automatic optical design.

Optics-Aware Retrieval We select initial structures using measured optical properties. Before indexing, an offline paraxial ray tracer labels each lens with its EFFL, total track length, RMS spot radius, and aperture. For candidate ℓ and target t, we rank candidates by

$$d(\ell,t) = 10\left|\frac{\mathrm{FoV}_\ell - \mathrm{FoV}_t}{\mathrm{FoV}_t}\right| + w_F\left|\frac{F_\ell - F_t}{F_t}\right| + 0.01\,\mathrm{RMS}_\ell,$$
$$w_F = \begin{cases} 1.0, & F_\ell > F_t, \\ 0.5, & F_\ell \le F_t, \end{cases} \tag{1}$$

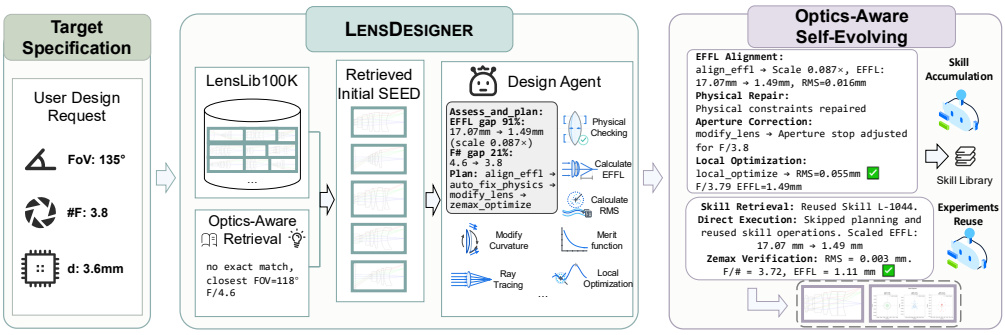

Figure 3: Main pipeline of LENSDESIGNER. Given a design query, the framework retrieves the closest seeds from LensLib100K, followed by Macro Orchestration for systematic optimization and Optics-Aware Self-Evolving for continuous capability enhancement via a curriculum-based agent.

where F denotes the f-number and RMS_ℓ is the precomputed paraxial RMS. The asymmetric weight penalizes candidates requiring aperture enlargement more than those requiring stopping down. FoV receives a larger weight because extending a seed's angular coverage is difficult; RMS breaks ties between otherwise similar candidates. A lightweight LLM reranks the 20 nearest candidates to choose five initial structures. Appendix C details the offline labeling and routing of out-of-distribution queries.

3.4 Self-Evolving for Optical Lens Design

Following the retrieval of seed structures, the design agent refines the configuration through a highly integrated pipeline comprising an optical tool suite, macro orchestration, and an optics-aware self-evolving mechanism.

Optical Tool Suite. The `align_effl` tool scales surface radii, thicknesses, and semi-diameters by the target-to-current EFFL ratio. The `modify_lens` tool rejects semi-diameter edits on non-stop surfaces and recomputes the system f-number after an aperture edit. Table 1 summarizes the main parameters. Table 5 defines the guardrails and workflow terminology. To redistribute optical power, `split_lens` divides a positive lens into two elements using the following power allocation:

$$\phi_{\text{orig}} = \frac{n_d - 1}{R_{\text{orig}}}, \quad \phi_1 = r \cdot \phi_{\text{orig}}, \quad \phi_2 = (1 - r) \cdot \phi_{\text{orig}}, \quad R_i = \frac{n_d - 1}{\phi_i} \tag{2}$$

Here, n_d is the refractive index, R_{orig} is the original radius, and $r \in [0.2, 0.8]$ (default 0.5) determines the power assigned to each element. The `local_optimize` tool then adjusts surface radii $\mathbf{R}$ by finite-difference gradient descent, minimizing

$$\widetilde{\mathrm{RMS}}(\mathbf{R}) = \mathrm{RMS}(\mathbf{R}) + \beta \cdot \mathrm{RMS}_0 \cdot \max\left(0, \frac{|\mathrm{EFFL} - \mathrm{EFFL}_0|}{|\mathrm{EFFL}_0|} - \tau\right) \tag{3}$$

where RMS_0 is the initial spot radius, EFFL_0 is the target focal length, $\beta = 0.5$, and $\tau = 0.03$. The penalty is zero within 3% relative EFFL error and grows linearly beyond it, discouraging a smaller spot obtained by sacrificing focal length. Optimization stops at an RMS of $3\,\mu m$. A step that flips a curvature's sign is rejected, with the radius clamped to 95% of its prior value. Appendix D.1 gives the numerical implementation.

Finally, `zemax_optimize` uses Zemax's built-in Damped Least Squares (DLS) optimizer [20] for a final parameter refinement. DLS is an off-the-shelf optimizer; ray tracing supplies the optical metrics used to evaluate its updates. We fix glass center thicknesses and use a custom merit function with per-field RMS weights, EFFL and f-number targets, a total-track-length limit, and a minimum glass thickness (Table 6). The optimized geometry is written back to the agent state.

Macro Orchestration. The system prompt specifies a four-stage standard operating procedure (SOP): retrieve initial structures, align focal length, modify and locally optimize the structure, and verify the result in Zemax. This replaces a free-form decision loop with a fixed sequence of design

Table 1: Key tool and optimization parameters. Detailed guardrails and Zemax merit-function operands are given in Tables 5 and 6.

Component	Parameter or rule
Focal-length alignment	Scale all dimensions by $\text{EFFL}_t/\text{EFFL}_\ell$, up to $3.0\times$.
Aperture editing	Scaling ratio in $[0.5, 3.0]$; stop-surface edits only.
Lens splitting	Power fraction $r \in [0.2, 0.8]$, default 0.5.
Local optimization	EFFL penalty $\beta = 0.5$, tolerance $\tau = 0.03$; stop at $3\,\mu m$ RMS.
Fast specification check	Reject EFFL error $> 2\%$ or FoV error $> 50\%$.
Repeated-action detection	Track 10 actions; change strategy after 3 identical pairs of a tool and its normalized input.
Zemax field weights	RSCE weights $4/5/6$ across the three evaluated fields.
Specification weights	EFFL: 1.0; f-number (WFNO): 2.0.
Geometry constraints	Max-TOTR penalty: 2.0; minimum glass thickness: $2\,mm$, weight 50.0.

stages. Before invoking Zemax, `check_spec` uses paraxial calculations to reject candidates with relative EFFL error above 2% or relative FoV error above 50%. It accepts a smaller-than-target f-number at this stage because the aperture can subsequently be stopped down. To limit repeated actions, the controller tracks the 10 most recent tool calls and forces a strategy change if the same tool and normalized input occur three times. Appendix D.2 details the workflow.

Optics-Aware Self-Evolving Mechanism A curriculum agent generates tasks with progressively more demanding specifications, and a design agent attempts them using the retrieved structures and optical tools (Fig. 3). The design agent records each geometric action together with the resulting RMS and EFFL changes. It extracts reusable operation sequences and boundary conditions from successful and failed attempts, excluding standard textbook practices. The extracted skills are deduplicated, saved in a persistent library, and loaded in subsequent tasks. This is training-free capability evolution: the skill library changes, while LLM weights remain fixed. Appendix D.3 specifies trajectory recording, distillation, and persistence.

3.5 LensArena Benchmark

In the domain of automated classical imaging lens design, recent methods [12, 37] only evaluate their performance on a handful of isolated design cases. This limited testing scope is primarily due to the immense computational overhead associated with traditional optimization pipelines. However, evaluating on restricted samples introduces randomness and bias and fails to rigorously reflect the generalization capabilities of these algorithms. Currently, the field lacks a large-scale, comprehensive testing baseline. Hence, we introduce our LensArena benchmark.

Our LensArena focuses specifically on classical imaging lens design. Since commonly used optical systems are highly mature and easily solved, modern engineering increasingly focuses on more challenging and extreme configurations like ultra-wide FoVs and large apertures. To rigorously test the limits of state-of-the-art automated methods against these frontiers, we structured our benchmark to reflect these modern demands. Given that full physical optical optimization is exceptionally time-consuming, our carefully curated set of 120 unique design queries forms our evaluation suite. Specifically, as shown in Fig. 2, LensArena deliberately includes a high proportion of challenging designs, covering a broad and diverse spectrum of specifications with the FoV ranging from 5° to 180° and the f/# spanning from 0.7 to 8.0. Notably, this benchmark is exceptionally challenging, as *70%* of the design queries are *out-of-distribution* relative to LensLib100K. Furthermore, the benchmark queries are systematically categorized into three distinct difficulty levels. For each specific task, we define a target RMS spot radius threshold dictated by its intrinsic difficulty. A generated lens configuration is recorded as a success strictly if its final optimized radius falls below this predefined standard. To the best of our knowledge, LensArena stands as the first benchmark dedicated to the systematic evaluation of automated optical system design. For examples from the LensArena benchmark, please kindly refer to Appendix H

Table 2: Main results evaluated on the LensArena Benchmark.

Method	Success Rate (%) ↑			Avg RMS (μm) ↓		
	Easy	Medium	Hard	Easy	Medium	Hard
[Existing Baselines]						
LensNet [9]	50.0	39.3	30.0	3.89	8.99	7.99
QGSO [12]	25.0	22.7	13.3	24.2	17.7	11.1
DeepLens [37]	4.2	1.5	0	29.6	41.1	-
[In Context Learning Baselines]						
`Gemini-3.0-flash`	37.5	36.3	26.7	43.56	94.7	120.4
`ChatGPT-5.2`	33.3	27.3	26.7	52.68	90.85	112.4
`Kimi-2.4`	33.3	16.6	23.3	44.7	87.45	108.7
`Qwen-3-8b`	8.3	12.1	20	96.32	147.7	215.3
`Qwen-3-32b`	16.6	21.2	20	67.83	124.23	176.43
LENSDESIGNER (Ours)						
w/ `Gemini-3.0-flash`	87.5	83.3	80.0	3.7	7.5	12.7
w/ `ChatGPT-5.2`	87.5	83.3	66.7	6.5	6.9	14.8
w/ `Kimi-2.4`	87.5	74.2	56.7	9.0	7.8	11.0
w/ `Qwen-3-8b`	70.8	51.5	40.0	6.5	9.2	14.8
w/ `Qwen-3-32b`	75.0	75.8	70.0	4.9	8.2	7.8

4 Experiments

4.1 Experimental Setup

Implemental Details. We leverage `Gemini-3.0-flash` [14], `Kimi-2.4` [21], `ChatGPT-5.2` [22], and `Qwen3-32b` [34] as the foundation LLMs, respectively, and Gemini-3-flash [14] for the optics-aware retrieval reranker and the self-evolve distiller, with lens-spec embeddings produced by text2vec-base [33] on a FAISS index. All LLM temperatures are set to 0 to ensure reproducibility. Our optical simulator couples an in-house paraxial ray-tracer (for offline Optics-Aware Retrieval labeling, online verification, and gradient-based local optimize) with `Ansys Zemax OpticStudio` [2] accessed through ZOS-API/pythonnet behind a Flask bridge. `Zemax` is the sole source of truth for final RMS / EFFL / f/#. See Appendix F.1 for more details

Evaluation Metrics. Given the absence of prior similar works in this domain, we formally define three primary metrics to evaluate the generated lens designs. The *Success Rate* measures the percentage of tasks where the final RMS spot radius successfully reaches a predefined target threshold, which varies dynamically based on the specific problem difficulty. The *Avg RMS* denotes the average RMS spot radius, sampled at 0, 0.7, and 1.0 times the maximum FoV. As the standard metric for optical evaluation, a smaller RMS indicates tighter ray convergence and superior image quality. Note that the *Avg RMS* is averaged only over the subset of successful designs, excluding failed queries.

Baselines To comprehensively evaluate our proposed approach, we compare it against three publicly available automated lens design methods and specifically constructed language model baselines. LensNet [9] serves as a data-driven baseline, utilizing a database retrieval mechanism combined with simple geometric scaling. QGSO [12] is an optimization-based approach powered by genetic algorithms. It stands as the current SOTA in automated optical design and can generate most conventional lens structures, though it suffers from exceedingly slow optimization speeds. DeepLens [37] represents a deep-learning-based framework that employs curriculum learning strategies to formulate lens configurations. Given the complete absence of existing LLM-based frameworks for this specific domain, we introduce LLM-based in-context-learning (ICL) baselines. This method relies on standard in-context learning and is driven by the exact same foundation language model as our framework to ensure a rigorously fair comparison.

4.2 Results

Comparison on the LensArena Benchmark. Table 2 compares the methods across three difficulty levels. With `Gemini-3.0-flash`, LENSDESIGNER achieves success rates of 87.5% on Easy and 80.0% on Hard queries. The strongest existing baseline on Hard queries is LensNet at 30.0%, while DeepLens solves none. The agent achieves an Avg RMS of 12.7 μm on successful Hard queries. It

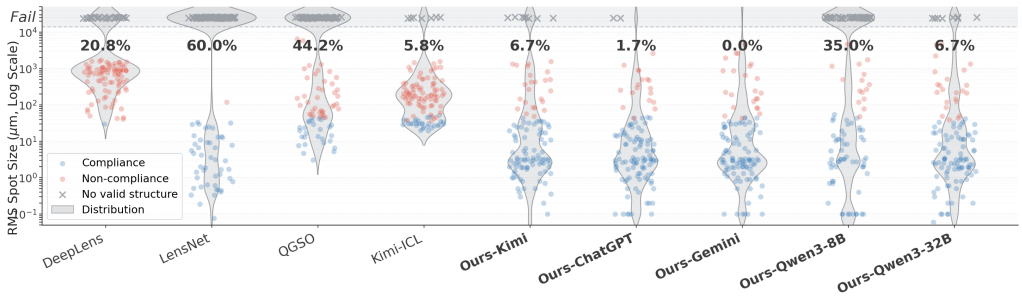

Figure 4: Results on the LensArena benchmark. The comparison reports the proportion of valid, ray-traceable structures, task success rates, and RMS spot radii for each method.

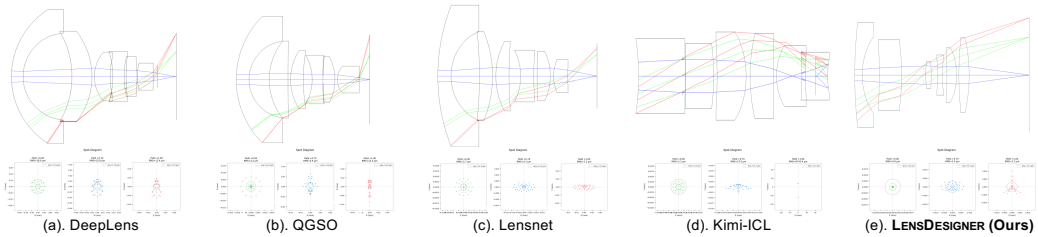

Figure 5: Visualized results of lens layouts and spot diagrams from the LensArena benchmark. LENSDESIGNER consistently yields optimal structural configurations and achieves the minimal RMS spot size, outperforming other methods in both design rationality and imaging quality.

produces solutions in tens of seconds, compared with several hours for conventional optimization methods.

As shown in Fig. 4, since LensNet is a retrieval-based method, it achieves a relatively high success rate for structures covered by its database. However, 60% of its generated structures fail during optical ray tracing because they fall outside the scope of its library. Conversely, although DeepLens produces valid structures in approximately 80% of cases, its final performance compliance rate is notably low. In contrast, our proposed method consistently yields a vast majority of structurally valid designs while achieving a superior compliance rate for the required RMS spot radius.

The ICL baselines achieve success rates of 8.3–37.5% and Avg RMS values of 43.56–215.3 μm, depending on the backbone and difficulty (Table 2). The full framework improves both metrics over the corresponding ICL baseline for every evaluated backbone. For example, with `Gemini-3.0-flash`, the Hard success rate rises from 26.7% to 80.0%, while Avg RMS falls from 120.4 to 12.7 μm. These results support combining retrieval, optical tools, and a prescribed workflow for lens design.

Example Analysis. As the lens layouts and the corresponding RMS spot diagrams shown in Fig. 5, LENSDESIGNER generates highly practical and physically valid optical structures. The layout visualization demonstrates a rational distribution of optical power across the generated glass elements, effectively managing internal ray angles to minimize spherical aberrations and chromatic dispersion. Furthermore, the accompanying spot diagrams confirm exceptional ray convergence at the focal plane. The tightly concentrated focal spots indicate that the generated designs strictly satisfy the demanding imaging specifications. This superior optical performance directly validates the effectiveness of our LENSDESIGNER.

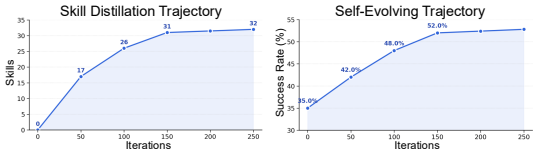

Figure 6: Trajectory of self-evolving.

Self-Evolving. Figure 6 tracks the accumulated skills and success rate across self-evolving iterations. The mean success rate shows a steady upward trend as skills accumulate. The agent distills operation sequences and failure-derived constraints from previous tasks, then reuses them in subsequent designs. This trend supports

the effectiveness of the training-free self-evolving process. Appendix G.4 specifies the three-run averaging and evaluation checkpoints.

4.3 Comparison with Human-Designed Patent Lenses

We compare LENSDESIGNER with nine lens prescriptions disclosed in patent documents, matching the queried FoV, f-number, and half-image-height y. These prescriptions provide human-designed reference systems for the selected specifications. Table 3 reports the Avg RMS of each patent design and the corresponding agent-generated design. LENSDESIGNER achieves a lower Avg RMS in six of the nine cases, with a mean of 9.850 μm versus 10.177 μm for the patent designs.

Table 3: Comparison with human-designed lens prescriptions from patent documents at matching FoV, f-number, and half-image-height. Avg RMS is reported in μm; lower values are better.

Patent	FoV (°)	f/#	y (mm)	Avg RMS (μm) ↓	
				Patent	LENSDESIGNER
US10203487B2	160.0	2.5	3.0	6.502	6.228
US6075658A	50.0	2.6	3.0	15.910	9.470
US20150168693A1	105.0	2.0	5.3	5.180	8.150
EP2073048A2	30.0	1.0	3.0	14.661	21.464
US7924510B2	60.0	1.4	2.0	4.556	4.411
US20250138284A1	40.0	2.8	4.3	2.202	11.747
US9488800B2	110.0	4.7	10.8	17.814	4.140
CN112394486B	65.0	1.9	4.3	20.119	18.774
US7173774B2	80.0	3.6	3.0	4.651	4.270
Average				**10.177**	**9.850**

Professional optical engineers assessed two selected cases. For the $160°$, $f/2.5$ query, LENSDESIGNER uses six air-spaced elements without cemented pairs, compared with ten elements including two cemented doublets in the patent, while attaining comparable Avg RMS (6.228 versus 6.502 μm). The patent also covers visible and near-infrared wavelengths, a broader spectral band than this query requires. For the $50°$, $f/2.6$ query, the agent uses six elements rather than five and places the stop at the rear rather than the front, yielding an Avg RMS of 9.470 versus 15.910 μm. These cases illustrate different choices of element count and stop placement for different specifications, using catalog glasses.

4.4 Ablation Study

To validate the effectiveness of each core module within LENSDESIGNER, we conduct a comprehensive ablation study using `Gemini-2.5` [6], as summarized in Table 4. A separate fixed-seed comparison is reported in Appendix G.2.

Optics-Aware Retrieval: Removing `LensLib100K` entirely forces the agent to design from scratch, leading to the worst overall performance. Furthermore, replacing our optics-aware retrieval with a Vanilla RAG yields sub-optimal initial seeds, resulting in a severe degradation of the RMS spot radius, particularly on hard queries (189.6 μm vs. 48.8 μm). This demonstrates the necessity of matching candidates based on deterministic physical metrics rather than mere semantic proximity.

Tool Suite & Macro Orchestration: Ablating the Optical Tool Suite or Macro Orchestration significantly reduces the success rate and increases aberration errors. This indicates that relying solely on the open-ended reasoning of large language models is insufficient for optical design. Strict physical guardrails and systematic, deterministic workflows are essential for continuous convergence.

Self-Evolving Mechanism: We further ablate the components of the self-evolving module. Removing either the reflection mechanism (which analyzes errors) or the skill library (which stores persistent cross-task heuristics) leads to notable performance drops on medium and hard tasks. This confirms that dynamically distilling and transferring historical experience is critical for mastering complex optical specifications. These component ablations compare reflection and persistent skill memory with the backbone held fixed.

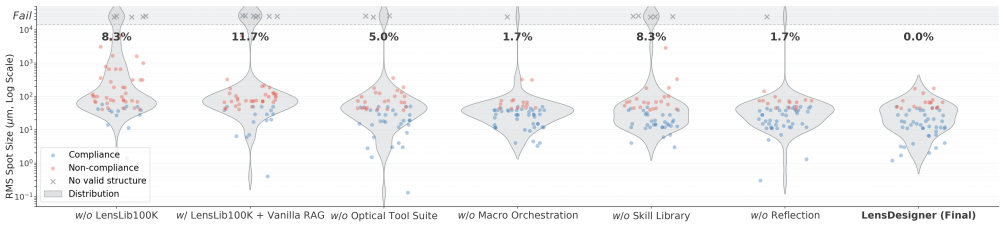

Figure 7: Ablation study on the core components of LENSDESIGNER. The violin plots illustrate the distribution of RMS spot radii across different model variants on the LensArena benchmark.

Table 4: Ablation study on the core components of LENSDESIGNER using the LensArena Benchmark. All variants are evaluated on the identical foundation model for fair comparison.

	Success Rate (%) ↑			Avg RMS (μm) ↓		
Method	Easy	Medium	Hard	Easy	Medium	Hard
w/o LensLib100K	25.0	39.3	26.7	89.6	194.4	293.4
w/ LensLib100K + Vanilla RAG	58.3	51.5	40.0	35.1	128.1	189.6
w/o Optical Tool Suite	66.7	63.6	26.7	37.1	54.7	68.9
w/o Macro Orchestration	80.0	69.7	33.3	21.8	34.2	87.9
[Self-Envolving]						
w/o Reflection	66.7	60.6	46.7	26.1	31.2	81.4
w/o Skill Library	66.7	72.7	46.7	23.2	31.2	81.4
LENSDESIGNER (Final)	91.7	70.0	53.2	19.2	33.4	48.8

5 Conclusion

In this paper, we introduced LENSDESIGNER, the first large language model-driven agent framework for automated classical imaging lens design. To comprehensively evaluate performance in this domain, we also proposed LensArena, the first large-scale and systematic benchmark tailored for optical system design. Extensive experiments demonstrate that our approach significantly outperforms existing optimization-based and deep-learning-based methods in both success rate and design efficiency, particularly on complex tasks with extreme optical parameters. We hope this work inspires broader research in AI for optics, uniting physical principles, agentic reasoning, and accumulated experience to advance optical design and scientific discovery.

Limitations and Future Work. Currently limited to spherical lenses, LENSDESIGNER faces challenges in scaling to aspheric or freeform surfaces and suffers from latency due to external ray-tracing software. Future efforts will extend its capabilities to non-spherical and complex folded systems to further automate optical engineering.

Acknowledgments. We thank the reviewers for their valuable feedback. We are also grateful to Shuhang Xie of the College of Optical Science and Engineering, Zhejiang University for helpful discussions on lens design quality assessment and assistance with collecting and comparing optical design patents. This research is part of the LensMind project at the College of Optical Science and Engineering, Zhejiang University, and the OptiBrain project at INSAIT. This work was partially supported by the Ministry of Education and Science of Bulgaria, as part of the Bulgarian National Roadmap for Research Infrastructure (support for INSAIT) and the European Union's Horizon Europe – the Framework Programme for Research and Innovation, under grant agreement 101168521.

Competing Interests. The authors declare no competing interests.

References

[1] Emre Can Acikgoz, Cheng Qian, Jonas Hübotter, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. Tool-r0: Self-evolving llm agents for tool-learning from zero data, 2026.

[2] Ansys, Inc. *Ansys Zemax OpticStudio*. Canonsburg, PA, USA, 2026. Available from: `https://www.ansys.com/products/optics/ansys-zemax-opticstudio`.

[3] Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024.

[4] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

[5] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[7] OpenClaw Contributors. Openclaw: Personal ai assistant. `https://github.com/openclaw/openclaw`, 2026.

[8] Geoffroi Côté, Jean-François Lalonde, and Simon Thibault. Extrapolating from lens design databases using deep learning. *Optics Express*, 27(20):28279–28292, 2019.

[9] Geoffroi Côté, Jean-François Lalonde, and Simon Thibault. Deep learning-enabled framework for automatic lens design starting point generation. *Optics Express*, 29(3):3841–3854, 2021.

[10] Geoffroi Côté, Fahim Mannan, Simon Thibault, Jean-François Lalonde, and Felix Heide. The differentiable lens: Compound lens search over glass surfaces and materials for object detection. In *Proc. CVPR*, pages 20803–20812, 2023.

[11] Rishi Bommasani et al. On the opportunities and risks of foundation models, 2022.

[12] Yao Gao, Qi Jiang, Shaohua Gao, Lei Sun, Kailun Yang, and Kaiwei Wang. Exploring quasi-global solutions to compound lens based computational imaging systems. *IEEE Transactions on Computational Imaging*, 2025.

[13] Yao Gao, Lei Sun, Shaohua Gao, Qi Jiang, Kailun Yang, Weijian Hu, Xiaolong Qian, Wenyong Li, Luc Van Gool, and Kaiwei Wang. Neuro-inspired automated lens design. *arXiv preprint arXiv:2510.09979*, 2025.

[14] Gemini. Gemini 3 flash. `https://deepmind.google/models/gemini/flash/`, 2025.

[15] Yuyu Geng, Lei Sun, Yao Gao, Xinxin Hu, Zhonghua Yi, Xiaolong Qian, Weijian Hu, Jian Bai, and Kaiwei Wang. Optiagent: A physics-driven agentic framework for automated optical design, 2026.

[16] Dabo Guo, Liang Yin, and Guang Yuan. New automatic optical design method based on combination of particle swarm optimization and least squares. *Optics Express*, 27(12):17027–17040, 2019.

[17] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia tools and applications*, 80(5):8091–8126, 2021.

[18] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[19] Xiaobing Liu, Xingxiang Zhang, Tianjiao Fu, Kaizhi Wang, Fukun Sun, Tongzheng Bai, and Duo Wang. Global information selectively guided gradient descent for ab initio optical design. *Optics & Laser Technology*, 184:112497, 2025.

[20] Joseph Meiron. Damped least-squares method for automatic lens design. *Journal of the Optical Society of America*, 55(9):1105–1109, 1965.

[21] Moonshot AI. Kimi chat (version 2.4) [large language model], 2024. Accessed: 2024-05-07.

[22] OpenAI. Chatgpt 5.2. `https://openai.com/index/introducing-gpt-5-2/`, 2025.

[23] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186, 2024.

[24] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[25] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[26] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[27] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[28] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024.

[29] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

[30] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huang, and Tao Gui. The rise and potential of large language model based agents: A survey, 2023.

[31] Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning, 2026.

[32] Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning, 2025.

[33] Ming Xu. Text2vec: Text to vector toolkit. `https://github.com/shibing624/text2vec`, 2023.

[34] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[35] John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652, 2024.

[36] Tong Yang, Guo-Fan Jin, and Jun Zhu. Automated design of freeform imaging systems. *Light: Science & Applications*, 6(10):e17081–e17081, 2017.

[37] Xinge Yang, Qiang Fu, and Wolfgang Heidrich. Curriculum learning for ab initio deep learned refractive optics. *Nature Communications*, 15(1):6572, 2024.

[38] Benqi Zhang, Guofan Jin, and Jun Zhu. Towards automatic freeform optics design: coarse and fine search of the three-mirror solution space. *Light: Science & Applications*, 10(1):65, 2021.

[39] Jiajun Zhang, Zhaofeng Cen, and Xiaotong Li. Automated design of machine vision lens based on the combination of particle swarm optimization and damped least squares. In *Optical Design and Testing X*, volume 11548, pages 261–272. SPIE, 2020.

[40] Yuntong Zhang, Haifeng Ruan, Zhiyu Fan, and Abhik Roychoudhury. Autocoderover: Autonomous program improvement. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 1592–1604, 2024.

Appendix

A Detailed Optical Design Preliminaries

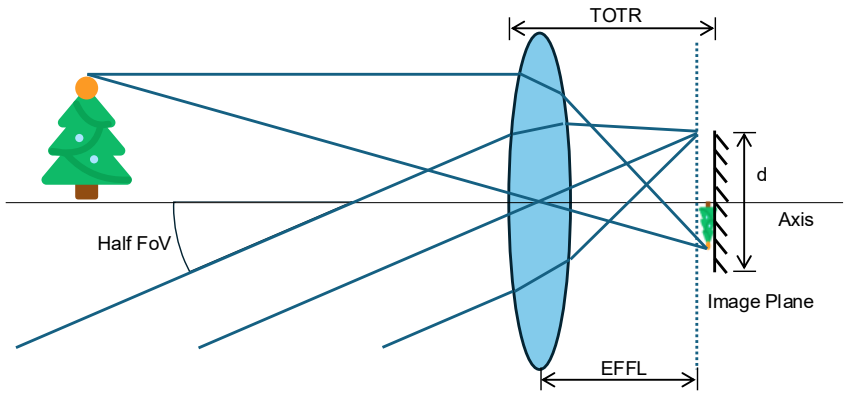

Figure 8: A simple example of a lens system.

This section provides a comprehensive exposition of the fundamental optical principles, the physical imaging process, and the quantitative metrics essential for evaluating lens configurations within our proposed framework. A simple example of lens system is shown in Fig. 8.

A.1 The Physical Imaging Process and Essential Components

The formation of an optical image is a deterministic physical process governed by the laws of refraction. In a standard computational environment, this process is modeled via sequential ray tracing. Light rays propagate from a source target, traverse multiple refractive media, and ultimately converge to form a visible representation. A complete spherical lens system necessitates the following fundamental structural elements:

- **Object Plane:** The physical or virtual surface where the target scene originates. Light rays radiate from discrete coordinate points on this plane toward the optical system.
- **Optical Elements:** A sequence of spherical lenses manufactured from specific glass materials. Each individual element is mathematically defined by its front and rear surface curvatures, central geometric thickness, and distinct material properties, specifically the refractive index and Abbe number. These elements sequentially refract incoming light bundles to precisely manipulate their trajectory.
- **Stops and Pupils:** The aperture stop is a physical boundary that restricts the diameter of the light bundle traversing the system. It governs the overall illumination brightness and depth of field. The effective optical images of this aperture stop, as formed by the lenses preceding and following it, are formally known as the entrance pupil and exit pupil, respectively.
- **Image Plane:** The terminal spatial surface where the refracted light rays intersect to form the final focused picture. In modern digital optics, this physical plane strictly coincides with the location of the electronic image sensor.

A.2 Detailed Explanation of Optical Parameters

To rigorously constrain the design parameter space and quantitatively evaluate optimization outcomes, we rely on a specific set of optical metrics. A detailed theoretical explanation of each metric is provided below:

- **Field of View (FoV):** This metric quantifies the maximum angular extent of the external environment that the optical system can project onto the sensor. It directly dictates the observational breadth of the lens. A larger FoV corresponds to wide angle photography, whereas a narrow FoV is characteristic of telephoto systems specifically designed for distant target observation.

- **F number (f/#):** Defined as the dimensionless ratio of the Effective Focal Length to the physical diameter of the entrance pupil. It serves as a universal indicator of the optical aperture size and the overall light gathering capability. A smaller F number indicates a comparatively larger aperture, allowing substantially more light to reach the sensor, which is highly beneficial for low illumination environments.
- **Optical Format (d) and Semi Image Height (y):** The optical format refers to the physical diagonal dimension of the active image sensor. To prevent mechanical vignetting and ensure full sensor coverage, the optical system must project an image circle that completely encompasses this sensor. The semi-image height y is exactly half of the optical format ($y = d/2$), representing the maximum radial coordinate on the image plane that requires sharp optical resolution.
- **Effective Focal Length (EFFL):** The physical distance measured from the rear principal plane of the optical system to the focal point on the image plane. It is the primary determinant of optical magnification. For a fixed sensor size, the EFFL mathematically restricts the Field of View, establishing a fundamental trade-off in optical geometry.
- **Total Track (TOTR):** This absolute metric measures the physical length of the entire optical assembly, calculated along the central optical axis from the front vertex of the first lens element to the final image plane. It is a critical geometric constraint for compact device packaging, such as integrating camera modules into thin mobile devices.
- **Root Mean Square (RMS) Spot Radius:** The most critical quantitative metric for evaluating optical aberrations. When light rays from a single infinitesimal point on the object plane pass through a non ideal lens system, they do not converge perfectly to a single identical point on the image plane. Instead, they scatter mathematically to form a microscopic blur circle known as the focal spot. The RMS computes the standard deviation of the spatial distribution of these intersecting rays. A smaller RMS spot radius directly correlates to higher imaging sharpness, minimal structural aberrations, and superior spatial resolution.

B Detailed Related Work

Automatic Lens Design. As previously noted, optical lens design is an empirical task highly dependent on human expertise. While modern software tools like Zemax and CODE V provide powerful simulation environments, they function primarily as calculators for intensive numerical optimization rather than autonomous decision makers. To alleviate the heavy reliance on manual labor and expert intuition, researchers have proposed various automatic lens design algorithms. These approaches primarily include methods driven by continuous optimization such as QGSO [16, 39, 12, 19, 13, 36, 38], neural network models including DeepLens [8, 9, 37], and database retrieval techniques like LensNet [9]. However, these existing solutions still present significant practical limitations, which often demand meticulous manual intervention and parameter tuning, frequently requiring several days to converge on a viable solution. Furthermore, pure retrieval techniques are inherently constrained and cannot handle novel structural requirements outside their existing libraries. In contrast, our proposed framework aims to rapidly generate compliant lens configurations through simple text interaction, entirely circumventing the need for complex parameter adjustments, specialized domain expertise, and prolonged optimization periods.

LLMs and LLM-based Agents. Large language models are increasingly recognized as the primary foundation for achieving artificial general intelligence [29, 5, 11]. Moving past traditional capacity expansion principles, the latest generative architectures, such as LLaMA [26], DeepSeek [18], Qwen3 [34], Claude Sonnet [3], and GPT 5 [25], clearly demonstrate their capability to serve as the central reasoning mechanisms for independent automated entities [30, 28]. By encapsulating cognitive engines within interactive frameworks, intelligent agents autonomously perceive contexts, formulate plans, and execute complex actions to significantly elevate productivity. For instance, OpenClaw [7] leverages tool utilization and environmental feedback for precise robotic manipulation. Similarly, these cognitive agents are revolutionizing other specialized disciplines, ranging from ChemCrow [4] designing chemical synthesis pathways to software agents [35, 40, 23] resolving intricate programming anomalies. Despite these interdisciplinary triumphs, deploying such autonomous entities within the highly empirical and numerically demanding domain of optical lens design remains largely unexplored [15].

C Detailed Optics-Aware Retrieval Mechanism

This section elaborates on our Optics-Aware Retrieval module. Rather than merely returning similar documents, our retrieval framework operates as a physical performance driven selector and an Out of Distribution router. We bypass conventional semantic matching by incorporating three domain specific innovations.

More Details about LensLib100K Construction We use the enhanced QGSO [12] automatic optical design algorithm. QGSO first define a $Population$ composed of all lens structures to be optimized, where each structure is coined an $Individual$. Without any empirical preference, the $Individual$ is the normalized lens parameters vector P from random initialization, and m $Individuals$ are initialized to constitute the $Population$ for enriching diversity. Then, a hybrid global and local optimization strategy is proposed to find multiple promising high-quality structures $\hat{\mathrm{P}}$ for P, coined the $Parent$, which minimizes the Merit Function. In addition, $Parent$ is mutated for more possible structures, the mutated $Parent$ is then mixed with the origin $Parent$ to serve as the start points of the next generation. Finally, the hybrid optimization, selection, and mutation processes are re-conducted for each generation, where the $Parent$ is outputted as the diverse lens structures when the generation meets the set number.

Simulator in the Loop Labeling Standard retrieval augmented generation relies on semantic text embeddings. Conversely, we perform physical characterization on every lens before adding it to the database. Utilizing an offline paraxial ray tracer, we evaluate each structure across multiple field coordinates to precompute essential physical metrics, including the Effective Focal Length, total track length, paraxial root mean square spot radius, and actual aperture radius. Consequently, candidate selection is solidly grounded in deterministic physical states rather than heuristic language estimations.

Asymmetric Optical Prior Distance Equation 1 defines the distance used for the first retrieval stage. The FoV weight is 10.0. The f-number weight is 1.0 when the candidate has a larger f-number than the target, and 0.5 otherwise. Enlarging the aperture admits additional marginal rays and can increase aberrations; stopping down does not require extending the seed's aperture. The paraxial RMS has weight 0.01 and serves as a tiebreaker. These fixed weights guide initial candidate selection before LLM reranking.

Two Stage Retrieval and Active Routing Our framework utilizes a widely adopted indexing database for initial screening, but pure numerical distance does not consistently guarantee the easiest trajectory. Therefore, we implement a two stage pipeline. First, the distance formula extracts the top twenty candidates. Next, a lightweight language model evaluates these candidates using structural heuristics to select the final top five starting structures. Furthermore, our retrieval module acts as an active router. If the target specifications represent an Out of Distribution query (specifically defined as a FoV deviation exceeding 10° or an f/# deviation exceeding 0.5), the system explicitly attaches a Standard Operating Procedure to the prompt. This proactive routing instructs the downstream agent on the exact sequential recovery path required, significantly enhancing robustness in sparsely populated optimization spaces.

D Detailed Methodology

This section provides a comprehensive technical breakdown of the autonomous optical design framework, detailing the exact physical equations, algorithmic guardrails, and prompt engineering strategies utilized to ensure physical validity and deterministic convergence.

D.1 Optical Tool Suite Implementation

The core of our framework relies on an optical tool suite. Unlike generic numerical editors, each tool strictly embeds optical engineering principles and boundary conditions.

Deterministic Focal Length Alignment (`align_effl`). When aligning the effective focal length (EFFL) of a retrieved candidate to the user specification, we avoid using optimization search. Instead,

Table 5: Physical guardrails and operational definitions used by LENSDESIGNER.

Term or guardrail	Definition and implementation
Stop Surface Recognition	Rejects semi-diameter edits on non-stop surfaces.
Aperture Scaling Limits	Rejects semi-diameter scaling outside the $[0.5, 3.0]$ ratio.
Dynamic F-number Sync	Synchronizes system F-number automatically after aperture edits.
Material Whitelist	Rejects non-catalog glasses (restricted to verified H- and D- types).
Loop Prevention / sliding-window circuit breaker	Tracks 10 recent tool calls; forces a strategy change after 3 identical pairs of a tool and its normalized input.
Open-ended reasoning	A free-form agent decision loop without a prescribed sequence of design stages.
SOP / logical design sequence	Four fixed stages: retrieval, focal-length alignment, structural modification and local optimization, and final verification.
Macroscopic alignment	The `align_effl` tool rescales lens geometry to match target EFFL.
Paraxial pre-filter	The `check_spec` tool rejects relative EFFL error $> 2\%$ or FoV error $> 50\%$.
Experience accumulation / self-evolving	Record actions and their metric changes, distill success and failure trajectories, and save reusable skills (Appendix D.3).
Training-free capability evolution	Update the persistent skill library without updating LLM weights.
Valid-structure proportion	Fraction of queries producing structures that Zemax can ray-trace.
Upward success-rate trend	Observed success-rate changes across self-evolving checkpoints (Fig. 6).

we apply a deterministic geometric scaling. All surface radii, thicknesses, and semi-diameters are synchronously multiplied by a scale factor defined as scale $=$ $\text{EFFL}_{\text{target}}/\text{EFFL}_{\text{current}}$. This transformation precisely scales the EFFL linearly while preserving the inherent relative aberration profiles. To prevent unphysical expansions, we enforce a strict 300% scaling upper limit. Crucially, the updated EFFL is immediately written back to the internal model state to prevent downstream evaluators from targeting outdated specifications.

Guarded Single-Surface Modification (`modify_lens`). To facilitate localized adjustments, this tool alters single parameters while being constrained by five distinct physical guardrails to prevent fatal execution errors, as summarized in Table 5.

Optical Power-Conserving Split (`split_lens`). To mitigate severe higher-order spherical aberrations induced by excessive local curvature, we introduce a tool that splits a single positive lens into two separate elements. This operation rigorously conserves the original optical power ϕ_{orig} derived from the thin-lens approximation:

$$\phi_{\text{orig}} = \frac{n_d - 1}{R_{\text{orig}}}, \quad \phi_1 = r \cdot \phi_{\text{orig}}, \quad \phi_2 = (1 - r) \cdot \phi_{\text{orig}}, \quad R_i = \frac{n_d - 1}{\phi_i} \tag{4}$$

The power distribution ratio r is bounded within $[0.2, 0.8]$ (defaulting to 0.5), ensuring a balanced division of the refractive workload.

Penalized Local Optimization (`local_optimize`). We implement a custom finite-difference gradient descent algorithm to provide rapid, localized fine-tuning. A critical innovation here is the integration of an EFFL soft penalty directly into the objective function. This ensures that the optimizer does not collapse the macroscopic focal length to artificially minimize the root-mean-square (RMS) spot radius:

$$\widetilde{\text{RMS}}(\mathbf{R}) = \text{RMS}(\mathbf{R}) + \beta \cdot \text{RMS}_0 \cdot \max\left(0, \frac{|\text{EFFL} - \text{EFFL}_0|}{|\text{EFFL}_0|} - \tau\right) \tag{5}$$

We empirically set the penalty weight $\beta = 0.5$ and the tolerance threshold $\tau = 0.03$. The gradient is estimated using central differences with an adaptive epsilon. To ensure geometric stability, if the updated curvature flips its sign (transitioning a convex surface to concave), the step is rejected, and the radius is clamped to 95% of its previous value. Optimization terminates early if the RMS reaches a noise floor of $3\,\mu m$. The `random_restart` tool complements this by applying controlled random perturbations uniformly sampled up to 15% to escape local minima.

Ground-Truth Verification (`zemax_optimize`). The `zemax_optimize` tool calls Zemax OpticStudio's built-in DLS optimizer for final numerical refinement. Zemax ray tracing evaluates the merit-function operands during optimization. We keep glass center thicknesses fixed to avoid collapse toward near-zero thickness and configure the constraints in Table 6. Field-dependent RMS weights emphasize the outer fields, EFFL and f-number terms constrain the requested specifications, and a unilateral TOTR penalty applies only above the length limit. DLS itself is an existing optimization algorithm rather than a contribution of this framework.

Table 6: Custom Merit Function configuration for Zemax DLS optimization.

Operand	Target	Weight	Physical Justification
RSCE (Per Field)	0	4 / 5 / 6	Escalating weights for challenging marginal fields.
EFFL / WFNO	User Spec	1.0 / 2.0	Weak anchoring to prevent macroscopic drift.
OPLT (Max TOTR)	Max Limit	2.0	Unilateral penalty applied only when exceeding limits.
MNCG	$2.0\,mm$	50.0	Hard constraint for minimum manufacturable glass thickness.

A crucial engineering component of this tool is the explicit state synchronization mechanism. Upon completion of the DLS optimization, the finalized surface geometries are immediately written back to the agent's internal memory state. This prevents the severe desynchronization bug where the agent evaluates subsequent actions using an outdated mathematical model while the external software holds the optimized state.

D.2 Macro Orchestration and Workflow

To navigate the expansive optical design space effectively, we replace generic open-ended reasoning paradigms with a deterministic Standard Operating Procedure (SOP) embedded in the system prompt. The agent must sequentially execute retrieval, macroscopic alignment, microscopic optimization, and final verification.

To optimize token usage and computational overhead, we introduce a two-tier evaluation mechanism. The `check_spec` tool acts as a rapid paraxial filter. It rejects candidates with relative EFFL error above 2% or relative FoV error above 50% before full ray tracing. It uniquely employs asymmetric validation for the F-number, instantly passing designs with smaller-than-target F-numbers since restricting the aperture is a trivial subsequent operation.

To mitigate infinite loops, a sliding window circuit breaker tracks the most recent ten tool actions. If any identical tool and normalized input combination repeats three times within this window, the system triggers a forced strategy shift. Furthermore, to accommodate context length limits, the prompt solely indexes available skills by name, dynamically loading the full verbose instructions only when the `get_skill_detail` function is actively invoked.

Refinement and Structural Changes. The agent first checks whether the retrieved seed matches the requested FoV, f-number, and structural arrangement. When these already match and the remaining differences concern scale, curvatures, or thicknesses, it uses focal-length alignment and local optimization. Structural changes through `split_lens` are reserved for residual higher-order aberrations that continuous refinement does not resolve.

For a six-element query with FoV 20°, $f/6.0$, half-image-height $10.8\,mm$, target EFFL $75.25\,mm$, and RMS below $0.010\,mm$, retrieval returns a structurally matching seed with EFFL $112.124\,mm$. The agent reuses a rescale-then-polish skill and applies `align_effl` with a scale factor of approximately 0.671, followed by `zemax_optimize`. The final EFFL is $75.127\,mm$ (relative error -0.163%), and Avg RMS is $0.006027\,mm$. The merit-function value decreases from 1.633 to 1.068. This trajectory follows the refinement path without changing the number of elements.

Unresolved Requests and Failure Reporting. Physical checks reject invalid modifications and restore the last valid state. Repeated tool calls trigger a strategy change through the circuit breaker. If the available candidates and optimization budget do not yield a compliant design, the framework reports the query as unresolved. A structure that cannot be ray-traced at the requested fields is classified as No valid structure; a traceable structure whose RMS exceeds the target is classified as Non-compliant. Both outcomes count as failures when computing Success Rate. An unresolved search identifies a limit of the available candidates and budget rather than proving that the specification is physically impossible.

For example, a six-element query with FoV $155°$, $f/4.5$, EFFL $0.67\,mm$, and an RMS target below $0.04\,mm$ retrieves only one distant candidate. During staged focal-length reduction, a modification produces an air-gap edge thickness of $-1.739\,mm$. The physical check rejects this geometry and rolls back the change. A subsequent attempt also fails to produce valid RMS measurements at all three fields, so the framework returns an explicit failure in the No valid structure category.

D.3 Knowledge-Grounded Self-Evolving Mechanics

Our training-free self-evolving process distills reusable design strategies from task trajectories without updating LLM weights.

Causal Trajectory Recording. Unlike conventional text-based trajectory logging, our memory module strictly tracks physical causality. Each record explicitly juxtaposes the geometric action against the resulting state changes in RMS and EFFL.

Heuristic Distillation Filters. To ensure only high-quality knowledge enters the skill library, we apply six specific heuristic filters before distillation. Notably, the system aggressively prioritizes trajectories that succeed on out-of-distribution (OOD) parameters where the database provides poor initial coverage. Furthermore, it explicitly distills negative experiences. For example, failing after a specific glass substitution teaches the agent valuable material boundary conditions, converting a failed session into a reusable constraint rule.

Optics-Aware Distillation Prompt. To prevent the skill library from being flooded with fundamental optical textbook principles, we employ a Optics-Aware distillation strategy. The distilling language model is explicitly provided with a baseline of common optical heuristics. It is strictly instructed to reject standard practices and extract only insights that transcend this baseline, such as specific non-intuitive operation sequences or extreme parametric combinations that yielded unexpected success.

Novelty Assessment and Persistence. Extracted skills undergo rigorous deduplication. A skill is rejected if its Bigram-Jaccard similarity score compared to existing skills exceeds 0.80. However, an identical sequence of tools is still classified as a novel skill if it employs extreme numerical parameters (such as an EFFL scaling factor exceeding 2.0 or a single-step RMS improvement exceeding 80%). These validated, highly specific skills are serialized into a persistent JSON library and injected directly into the active prompt context for all subsequent tasks, establishing a continuous and interpretable evolution cycle.

D.4 General Implementation Details

The complete framework is instantiated utilizing the LangChain architecture, encompassing standard modular components for tool execution and retrieval. The reasoning trajectory is managed using a sliding window scratchpad that retains only the ten most recent steps to prevent context overflow. Communication with the Zemax OpticStudio environment is facilitated through a robust, cross-process Flask bridge, allowing the Python-based agent to seamlessly orchestrate the proprietary physical engine.

Our hardware setup includes an NVIDIA 4090 GPU (24GB VRAM) paired with an Intel Core i9-14900K CPU and 64GB of RAM. While we access most LLMs via their official APIs, the Qwen models are deployed locally and run exclusively on our GPU.

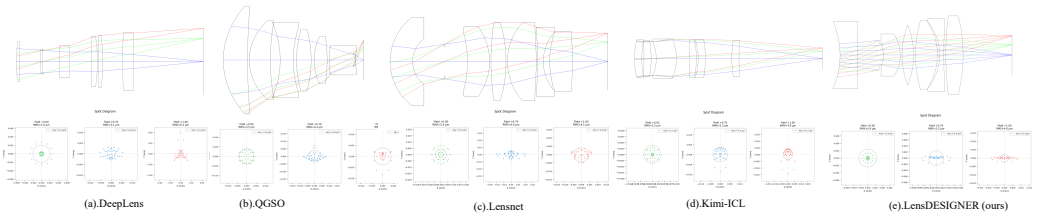

Figure 9: Additional visualized results of lens layouts and spot diagrams from the `LensArena` benchmark. LENSDESIGNER demonstrates robust performance across a diverse range of design specifications, consistently producing optimized optical structures with superior imaging convergence compared to baseline methods.

E More Qualitative Results

Here we show more qualitative results from our LensArena benchmark in Fig. 9.

F More Experimental Setup Details

F.1 More Implementation Details

We use a locally served *Qwen3-32b* [34] (deployed via vLLM with an OpenAI-compatible endpoint and the `enable_thinking=False` chat-template flag) as the primary tool-calling agent, and *Gemini-3-flash-preview* [14] for both the optics-aware retrieval reranker (Sec. 3.3) and the self-evolve trajectory distiller (Sec. 3.4). Lens-spec embeddings are produced by the open-source *text2vec-base* model [33] on a locally persisted FAISS index. We set all LLM temperatures to 0 to ensure reproducibility of tool selection, retrieval reranking, and skill distillation, with `max_tokens` of 512 for the agent loop, 1,000 for the reranker, and 8,000 for the distiller. Our optical simulator is a two-tier setup that couples an in-house Python paraxial ray-tracer—used for offline RAG labeling, online `check_spec` verification, and the `local_optimize` finite-difference gradient descent—with *Ansys Zemax OpticStudio* [2] accessed through the ZOS-API .NET bindings (`pythonnet`) and wrapped behind a Flask HTTP bridge, with Zemax serving as the sole source of truth for final RMS / EFFL / F-number / merit-function values; all glass materials are restricted to the CDGM library, and Zemax local optimization runs the Damped-Least-Squares solver under our customized merit function. The ReAct controller is built on LangChain's `create_react_agent` with a 10-step sliding-window scratchpad and a maximum of 25 reasoning iterations per query.

G Additional Experimental Results

G.1 Ablation Study with Gemini-3.1-flash

The ablation study in Sec. 4.4 uses `Gemini-2.5` to reduce experimental cost. To assess whether the component contributions extend to another backbone, we repeat the same ablation variants using `Gemini-3.1-flash`. Table 7 reports Success Rate and Avg RMS across the three LensArena difficulty levels. As in the main experiments, Avg RMS is computed only over successful queries.

Optics-Aware Retrieval: Removing LensLib100K reduces the Hard success rate from 56.7% to 30.0% and increases Avg RMS from 36.5 to 187.5 μm. Retaining the library with Vanilla RAG improves success to 43.3%, but Avg RMS remains 162.4 μm. These results support using physically matched initial structures rather than semantic similarity alone.

Tool Suite & Macro Orchestration: Removing the Optical Tool Suite or Macro Orchestration lowers the Hard success rate to 26.7% and 36.7%, respectively. Their corresponding Avg RMS values increase to 59.9 and 65.7 μm. Both physical constraints on individual actions and an ordered design workflow contribute to solving challenging queries.

Self-Evolving Mechanism: Removing either reflection or the skill library reduces the Hard success rate to 50.0%, while Avg RMS increases to 67.8 and 63.6 μm, respectively. On Medium queries,

Table 7: Ablation study on LensArena using `Gemini-3.1-flash`. All variants use the same backbone and evaluation metrics as the full pipeline.

	Success Rate (%) ↑			Avg RMS (μm) ↓		
Method	Easy	Medium	Hard	Easy	Medium	Hard
w/o LensLib100K	29.2	39.4	30.0	77.5	96.2	187.5
w/ LensLib100K + Vanilla RAG	62.5	51.5	43.3	35.2	88.4	162.4
w/o Optical Tool Suite	70.8	63.6	26.7	33.8	42.5	59.9
w/o Macro Orchestration	79.2	71.2	36.7	18.3	42.7	65.7
[Self-Evolving]						
w/o Reflection	70.8	60.6	50.0	15.8	29.1	67.8
w/o Skill Library	70.8	72.7	50.0	14.4	25.7	63.6
LENSDESIGNER (Final)	83.3	74.2	56.7	11.3	26.8	36.5

removing the skill library yields a slightly lower successful-query Avg RMS (25.7 versus 26.8 μm), alongside a lower success rate (72.7% versus 74.2%). Since these averages cover different successful subsets, the two metrics should be read together. The full pipeline achieves the highest success rate at every difficulty level, supporting the benefits of experience accumulation with a second backbone.

G.2 Agent Contribution beyond Retrieval

We hold the retrieved initial structure fixed for each query and compare direct seed optimization without an LLM against the full agent. Seed + Local DLS and Seed + Zemax hand the same seed directly to their respective optimizers. The agent comparisons use `Gemini-3.0-flash` and `ChatGPT-5.2`, separately from the `Gemini-2.5` component ablations in Sec. 4.4. Table 8 reports the results by difficulty and overall.

Table 8: Agent contribution beyond retrieval on LensArena, using identical initial seeds for each query. Overall success rates include all queries; Avg RMS averages successful queries.

	Success Rate (%) ↑				Avg RMS (μm) ↓			
Method	Easy	Medium	Hard	Overall	Easy	Medium	Hard	Overall
Seed + Local DLS	75.0	71.2	53.3	67.5	13.60	32.40	44.50	30.61
Seed + Zemax	79.2	72.7	50.0	68.3	14.20	30.50	48.00	29.92
LENSDESIGNER `Gemini-3.0-flash`	87.5	83.3	80.0	83.3	3.7	7.5	12.7	8.04
LENSDESIGNER `ChatGPT-5.2`	87.5	83.3	66.7	80.0	6.5	6.9	14.8	8.80

Agent Contribution beyond Retrieval: Seed + Local DLS and Seed + Zemax perform local optimization from the retrieved seed without LLM-controlled tool use. Their overall success rates are 67.5% and 68.3%, with Avg RMS values of 30.61 and 29.92 μm, respectively. From the same initial seeds, LENSDESIGNER achieves 83.3% success and 8.04 μm Avg RMS with `Gemini-3.0-flash`, and 80.0% success and 8.80 μm with `ChatGPT-5.2`. Relative to Seed + Zemax, the success-rate gains are 15.0 and 11.7 percentage points. Overall success rates include all queries, while overall Avg RMS averages the successful queries. These comparisons show that the agent improves results beyond retrieval followed by standalone optimization.

G.3 Effect of Reasoning Mode

The default implementation disables thinking to limit latency and cost. To examine the effect on design quality, we compare LENSDESIGNER with thinking disabled and enabled using `Gemini-3.6-flash` on LensArena. Table 9 reports Success Rate and Avg RMS across the three difficulty levels.

Enabling thinking improves both reported metrics at every difficulty level. Success rates increase by 4.1, 9.1, and 6.7 percentage points on Easy, Medium, and Hard queries, respectively. Avg RMS decreases from 9.8 to 7.3 μm, 27.7 to 19.9 μm, and 32.1 to 25.6 μm. The larger success-rate gains on Medium and Hard queries indicate that additional reasoning can improve the design agent's performance on more demanding specifications.

Table 9: Effect of reasoning mode on LensArena with `Gemini-3.6-flash`. Avg RMS is computed over successful queries.

	Success Rate (%) ↑			Avg RMS (μm) ↓		
Reasoning mode	Easy	Medium	Hard	Easy	Medium	Hard
Thinking disabled	91.7	77.3	60.0	9.8	27.7	32.1
Thinking enabled	95.8	86.4	66.7	7.3	19.9	25.6

G.4 Self-Evolving Evaluation Protocol

Figure 6 reports the mean success rate across three independent runs with different random seeds. Each self-evolving iteration processes a batch of 20 queries. We record success rates every 50 iterations. Averaging across the three runs summarizes the evolution of design performance as the agent accumulates experience.

H Data Examples from LensArena

Here we show some examples from our LensArena benchmark.

Linting Error Message


```
{
  "easy": {
    "data": [
      {
        "id": 1,
        "fov": 30,
        "fnum": 4.5,
        "y": 10.8,
        "effl": 40.31,
        "rms_target": 0.048,
        "distortion_pct": 5,
        "difficulty": "easy",
        "design": "Please design an F/4.5 lens with a FOV of 30 degrees, half
        ↪  image height y=10.8mm, target RMS < 0.048mm, and distortion \leq
        ↪  5%."
      },
      {
        "id": 11,
        "fov": 25,
        "fnum": 5.9,
        "y": 2.0,
        "effl": 9.02,
        "rms_target": 0.056,
        "distortion_pct": 5,
        "difficulty": "easy",
        "design": "Could you design a lens for me with FOV=25 degrees and
        ↪  F/5.9, featuring a half image height y=2.0mm, target RMS < 0.056mm,
        ↪  and distortion \leq 5%?"
      }
    ]
  },

...

  "medium": {
    "data": [
      {
        "id": 16,
        "fov": 70,
```

```
        "fnum": 2.6,
        "y": 5.3,
        "effl": 7.57,
        "rms_target": 0.057,
        "distortion_pct": 5,
        "difficulty": "medium",
        "design": "I need you to design a lens with a 70-degree FOV and F/2.6.
        ↪  The half image height should be y=5.3mm, target RMS < 0.057mm, and
        ↪  distortion \leq 5%."
      },
      {
        "id": 17,
        "fov": 105,
        "fnum": 4.5,
        "y": 3.0,
        "effl": 2.3,
        "rms_target": 0.03,
        "distortion_pct": 5,
        "difficulty": "medium",
        "design": "Design an F/4.5 lens with a 105-degree FOV for me, where
        ↪  half image height y=3.0mm, target RMS < 0.03mm, and distortion \leq
        ↪  5%."
      }
    ]
  },

...

  "hard": {
    "data": [
      {
        "id": 10,
        "fov": 110,
        "fnum": 1.9,
        "y": 2.0,
        "effl": 1.4,
        "rms_target": 0.048,
        "distortion_pct": 5,
        "difficulty": "hard",
        "design": "Can you design a lens with FOV=110 degrees, F/1.9, half
        ↪  image height y=2.0mm, target RMS < 0.048mm, and distortion \leq
        ↪  5/%?"
      },
      {
        "id": 13,
        "fov": 10,
        "fnum": 2.0,
        "y": 3.6,
        "effl": 41.15,
        "rms_target": 0.038,
        "distortion_pct": 5,
        "difficulty": "hard",
        "design": "I would like you to design a lens featuring a 10-degree FOV
        ↪  and F/2.0, with a half image height y=3.6mm, target RMS < 0.038mm,
        ↪  and distortion \leq 5%."
      }
    ]
  }
}
```

Figure 10: Some examples from our LensAreana benchmark. Queries are divided into three difficulty levels. Most challenging queries (e.g. ultra-wide-angle lenses and large-aperture lenses) are included. For each query, there is an RMS target to determine whether the result is successful or not.

I Limitations and Broader Impact

I.1 Detailed Limitations

While LENSDESIGNER significantly advances autonomous optical engineering, several technical boundaries remain:

- **Geometrical Constraints:** Our current framework is primarily optimized for co-axial spherical systems. It does not yet natively support off-axis systems, tilted components, or non-imaging optics (e.g., light guides), which require higher-dimensional spatial reasoning.
- **Material Selection:** The agent currently selects from a predefined subset of common optical glasses. Expanding this to full industrial catalogs (e.g., Schott or Ohara) or incorporating temperature-dependent refractive index data (dn/dT) would significantly increase the search space complexity.
- **Computational Latency:** The system's performance is partially bottlenecked by the API call latency of commercial ray-tracing kernels. Developing a lightweight, differentiable proxy for initial structural screening could further accelerate the self-evolving process.

I.2 Broader Impact

I.2.1 Intended Uses

LENSDESIGNER aims to democratize high-end optical engineering through several key applications:

- **Accelerating R&D:** Reducing the design cycle for consumer electronics, medical imaging, and automotive sensors from weeks to minutes.
- **Educational Support:** Serving as an interactive pedagogical tool for students to explore the trade-offs between lens complexity and imaging performance.
- **Cross-Disciplinary Innovation:** Enabling researchers in fields such as robotics or biology to customize specialized imaging systems without requiring years of domain-specific training.

I.2.2 Potential Risks and Mitigations

The automation of physical design carries inherent responsibilities:

- **Dual-Use Concerns:** Advanced engineering tools could theoretically be applied to develop components for surveillance or specialized weaponry. We advocate for integrating safety alignment protocols within the foundation models to flag requests that violate security norms.
- **Over-reliance on Simulation:** AI-generated designs may not fully account for manufacturing tolerances (e.g., centering errors, mounting stresses). We emphasize that LENSDESIGNER is an *augmentative tool*; final designs must undergo rigorous optomechanical tolerance analysis by experts before fabrication.
- **Environmental Impact:** Large-scale agentic frameworks consume significant computational power. Our self-evolving mechanism mitigates this by distilling experience into a compact library, reducing the need for brute-force optimization in future tasks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction specify three main contributions: a large-scale lens library (LensLib100K), a self-evolving agentic framework (LENSDESIGNER), and a comprehensive benchmark (LensArena). Our extensive experiments in Section 4 and qualitative results in Figure 5 directly validate these claims by demonstrating significant performance gains and the generation of valid, high-quality lens structures.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 5 (and further detail them in Appendix I), specifically addressing the current constraint to spherical lens systems and the computational latency associated with external optical simulation kernels.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper focuses on the design of an autonomous agentic framework and its empirical validation. It does not introduce new mathematical theorems or formal theoretical proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide comprehensive implementation details in Section 4 and the Appendix, including the agent's prompt templates, model configurations, and hyperparameters for the QGSO optimization. Furthermore, we have included the core code and a sample of the LensArena benchmark in the supplementary material to ensure full reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided the core algorithm implementation and the evaluation benchmark in the supplemental material. All source code, the full `LensLib100K` dataset, and pre-trained agent checkpoints will be made publicly available on GitHub upon the paper's acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide comprehensive experimental settings in Section 4 and the Appendix. This includes the LLM backbone configurations, the specific hyperparameters for the QGSO optimization algorithm, and the precise design specifications for the LensArena benchmark.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Due to the substantial computational cost and latency of high-fidelity ray-tracing simulations (Zemax) within the self-evolving loop, some large-scale experiments were conducted once. However, we validated the consistency of our results across various design specifications in `LensArena` to ensure our conclusions are not based on outliers.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Detailed information regarding the compute workers (CPUs and GPUs), memory specifications is provided in Appendix D.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research strictly adheres to the NeurIPS Code of Ethics. The data used for `LensLib100K` was synthesized based on well-established optical principles or sourced from publicly available, open-access lens databases without any privacy or copyright infringements. We have also provided a Broader Impact statement in the Supplementary Material to address the dual-use potential of automated lens design and emphasize the importance of human-in-the-loop validation for safety-critical optical systems.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We did discuss them in Appendix I

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: While our work focuses on general optical design, we recognize the potential for dual-use applications. To ensure responsible release, we have included a comprehensive Broader Impact statement in the Appendix that outlines the intended use cases and identifies potential risks. Furthermore, our released dataset and code are hosted with clear usage guidelines, and we advocate for the integration of safety alignment protocols in the underlying LLM backbones to prevent the generation of designs that violate security or ethical norms.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this work are properly cited and used in accordance with their respective licenses. This includes the foundation models (e.g., GPT-5.2, Qwen-3) accessed via official APIs, the Zemax OpticStudio software used under a valid professional license, and the lens databases (e.g., LensNet) that informed our `LensLib100K` construction. We have detailed the versions and licensing terms for these assets in the "Implementation Details" section of the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The new assets (code and datasets) were not included in this submission's supplemental zip file to ensure a streamlined review process. However, the conceptual documentation and implementation details are provided in Section 4 and the Appendix. Full access to these assets, including comprehensive README files and environment specifications, will be provided via a public repository upon publication.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our research does not involve crowdsourcing or research with human subjects. All data used in the `LensLib100K` and `LensArena` datasets were generated through automated physical simulations and existing open-source optical libraries.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This research is strictly limited to computational modeling and automated optical design. It does not involve human subjects, clinical trials, or the collection of personal data; therefore, Institutional Review Board (IRB) approval is not required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Large Language Models (LLMs), including GPT-5.2 and Qwen-3, are central to our proposed LENSDESIGNER framework. They function as reasoning agents that interpret optical specifications, suggest structural modifications, and drive the self-evolving curriculum. The specific roles, prompt templates, and integration of these models within our agentic workflow are thoroughly described in Section 3 and the Appendix.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.