Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-025-02619-z

Reinforcement learning-based laser cutting machine parameter optimization

Khanh Quan Pham[1] · Majid Kundroo[1] · Geunwoo Ban[2] · Seongho Bae[2] · Taehong Kim[1]

Received: 22 January 2025 / Accepted: 1 May 2025

Abstract
Achieving high accuracy in laser-based cutting of optical films requires careful tuning of parameters such as focal length and laser power beam, adjusted according to the specific properties of each film type. Trial-and-error based traditional methods are used to find the most suitable cutting parameters for various films, but they are slow and inaccurate. To address this issue, this paper presents the Reinforcement Learning for Laser Cutting (RL^2C) algorithm, which uses Q-learning with an epsilon-greedy policy to dynamically optimize cutting parameters, significantly reducing taper size and film wastage. Additionally, RL^2C incorporates a dynamic environment space adaptability mechanism to allow it to adapt to new states encountered during the learning process over multiple batches of experiments. Experimental results demonstrate that RL^2C requires fewer steps and less time to find optimal cutting parameters compared to various RL-based optimization methods. Specifically, RL^2C reduces the number of optimization steps by up to 12.5% and processing time by up to 81.8% compared to existing methods. This study demonstrates the potential of RL in industrial laser-cutting processes by improving cut quality, reducing time and film wastage, and minimizing manual interventions.

Keywords Reinforcement learning · Q-learning · Parameter optimization · Dynamic environment space

Introduction

Laser cutting is one of the key enabling technologies in smart manufacturing, which has gained prominence due to its accuracy and efficiency in manufacturing parts for different applications across industries like electronics, automotive aerospace industries, and others. This technology is capable of cutting many materials, such as metals, plastics, woods, fabrics, polymers, composites, etc., with less wastage and high accuracy; that is why it is essential in those sectors where high accuracy and complex cutting operations are required (Genna et al., 2020; Singh et al., 2022). The overall quality of laser cutting depends mainly on key factors such as laser power, pulse frequency, cutting speed, and focus position (Ghany & Newishy, 2005). Especially, the process of selecting the optimal cutting parameters, particularly laser power and focal length, becomes even more challenging due to the growing variety of materials and cutting requirements. For instance, optical films, which can be used in foldable screens and automotive displays, require fine-tuning of parameters to obtain the optimal cutting quality with the minimum number of defects and unnecessary wastage (Chan et al., 2016; Li et al., 2020).

The conventional methods used in the process of identifying the best laser cutting parameters are manual trial-and-error methods and simulation models, which are both time-consuming and inaccurate. These simulation models often use numerical techniques such as finite element analysis (FEA) and computational fluid dynamics (CFD) for estimating thermo-mechanical properties during cutting operations

✉ Taehong Kim
taehongkim@cbnu.ac.kr

Khanh Quan Pham
khanhquan@cbnu.ac.kr

Majid Kundroo
kundroomajid@cbnu.ac.kr

Geunwoo Ban
gwban@npstech.co.kr

Seongho Bae
bsh@npstech.co.kr

[1] School of Information and Communication Engineering, Chungbuk National University, Cheongju 28644, Korea

[2] NPS CO., LTD., Cheongju 28371, Korea

Published online: 21 May 2025

(Marimuthu et al., 2013). Nevertheless, these methods do not consider the relationships of input and output of cutting parameters, especially in dynamic manufacturing environments (Peirovi et al., 2017). To overcome these challenges, some more sophisticated machine learning (ML) algorithms (Ren et al., 2023; Ürgün et al., 2024) have recently been used to automate the optimization process: artificial neural networks (ANNs), metaheuristic algorithms including genetic algorithms, or particle swarm optimization (PSO). Despite the promising nature of the aforementioned approaches, they have not proven effective in exploring huge parameter spaces, working with a wide range of materials, and managing complex industrial environments (Ürgün et al., 2024). In particular, ANN-based models have high overfitting issues while modeling new or unknown materials, and metaheuristic algorithms face difficulties in converging in real-time environments (Ren et al., 2023).

Among the ML approaches, the application of reinforcement learning (RL) (Mamond et al., 2025), especially Q-learning, offers a promising solution for automating the work of laser cutting. Unlike supervised learning or metaheuristic algorithms, RL enables an agent to learn from its interaction with the environment and improves its actions based on the rewards gained during the process. Specifically, Q-Learning can be employed to optimize laser cutting parameters, minimizing the need for manual intervention. The use of RL enables more flexible, continuous adaptation to new materials and conditions as they arise, offering a distinct advantage over static optimization techniques such as Bayesian optimization (Wahab et al., 2020) or PSO (Pramanik et al., 2022).

This paper introduces the RL for laser cutting (RL^{2}C) algorithm, a Q-learning-based method designed to optimize laser cutting parameters across three critical stages: focus, power, and taper. RL^{2}C enables the agent to update its knowledge over and over as it meets more materials and cutting conditions. This adaptive mechanism enhances the algorithm's efficiency and effectiveness, overcoming the shortcomings of the static optimization methods and enabling real-time decision-making in industrial settings. While the core methodology is based on standard tabular Q-Learning, our approach features innovations such as stage-wise optimization and dynamic state-space updates, enabling the algorithm to address the challenges posed by different optical films and their unique properties.

The key contributions of this paper are:

- **RL^{2}C algorithm** We propose a novel Q-Learning-based algorithm designed to optimize laser cutting parameters for optical films, integrating a dynamic environment space adaptability mechanism to accommodate real-time learning and the complexities of material characteristics.
- **Industrial application** This study highlights the practical application of the RL^{2}C algorithm in an industrial setting, specifically for laser cutting machines, showcasing its potential to enhance production efficiency and cutting quality.
- **Comparative analysis** We present a thorough comparison between RL^{2}C and baseline optimization techniques, such as RL-based Bayesian Optimization, RL-based PSO, and Random Search. The analysis demonstrates the effectiveness of our method in time and step efficiency across multiple optimization stages. It highlights the superior performance of RL^{2}C in achieving faster convergence and higher precision in parameter selection.

The rest of this paper is organized as follows: section "Related works" discusses the related works, providing an overview of existing optimization techniques in laser cutting, including ML and RL approaches. Section "Overview of laser cutting process" presents the overview of the laser cutting process, highlighting material film structures, dataset characteristics, and the optimization stages. Section "Proposed method" introduces the proposed RL^{2}C algorithm, detailing its structure, dynamic adaptability mechanism, and implementation steps. Section "Experimental results" presents experimental results, including a comparative analysis of RL^{2}C with baseline methods in terms of step and time efficiency. Finally, section "Conclusion" concludes the study, summarizing key findings and discussing potential future research directions.

Related works

Laser cutting is a widely adopted manufacturing process, but determining the optimal machine parameters remains a significant challenge due to the complex interactions between parameters such as focal length, power, cutting speed, and the material properties of the workpiece (Behbahani et al., 2022). Traditional approaches, including trial-and-error experiments and simulation-based methods, have notable limitations. These methods are often time-consuming, computationally expensive, and lack the ability to model nonlinear relationships or adapt to dynamic changes in the cutting environment (Xu et al., 2014).

ML techniques (Liu et al., 2023), particularly ANNs, have emerged as powerful tools for optimizing laser cutting parameters. ANNs are capable of learning complex, non-linear relationships between input parameters and output quality metrics, such as line width, brightness, and taper size. Early studies by Chen et al. (2005) and Dixin et al. (2006) used ANNs to quantitatively describe the relationship between cutting quality and parameters in non-vertical laser cutting scenarios. Building on this foundation, Tsai et al. (2008) integrated ANNs with genetic algorithms, enabling the automatic discovery of optimal parameter settings and improving cutting quality. Beyond ANNs, hybrid ML approaches (Mi et al., 2023; Park et al., 2022; Wang et al., 2025) have also been explored. Tercan et al. (2017) proposed a clustering and classification framework for laser cutting planning. By first clustering simulation data into performance-based groups and applying classification trees, they identified parameter regions that maximized cutting efficiency. However, these methods often suffer from computational overhead and reduced scalability when applied to high-dimensional parameter spaces or real-time scenarios.

In addition to ML, metaheuristic optimization methods have been applied to parameter optimization for laser cutting. Techniques such as genetic algorithms, PSO, whale optimization algorithm (WOA), and ant lion optimization (ALO) metaheuristic methods have been utilized to explore the parameter space and identify optimal configurations. Ürgün et al. (2024) demonstrated the effectiveness of these algorithms in optimizing laser cutter parameters. While metaheuristic methods provide robust solutions for static problems, they struggle to adapt dynamically to changing environments or materials, limiting their practicality in real-world applications.

Recent research has shifted toward RL as a more flexible and scalable solution for parameter optimization. RL distinguishes itself from supervised learning (SL) or unsupervised learning (USL) by training through dynamic interactions with the environment rather than relying on static datasets. This makes RL more flexible, as it can adjust its reward function to suit specific tasks and optimize parameters more effectively. Kuprikov et al. (2022) employed deep RL to optimize laser cutting parameters, showcasing its ability to achieve higher precision compared to static optimization methods. Similarly, Behbahani et al. (2022) applied ML techniques, including RL, to optimize the machining process of alumina ceramics. Besides, Chang et al. (2024) explores the optimization of laser annealing parameters using Bayesian RL (BRL). This study leverages fixed and variable prior knowledge derived from experimental data and technology computer-aided design (TCAD) simulations to guide the RL agent in selecting optimal parameters, such as laser power, repetition rate, and processing temperature. Zhang et al. (2023) proposed a Q-learning-based multi-objective PSO (QL-MoPSO) framework for distributed flow-shop scheduling problems. Their hybrid approach balanced exploration and exploitation capabilities, achieving faster convergence and improved diversity in solutions.

However, existing RL approaches often assume deterministic state transitions and static learning mechanisms. This means that if a state is not included in the predefined state space during training, the RL agent may struggle to generalize and adapt effectively when encountering unseen conditions. Building on this foundation, our proposed RL^{2}C algorithm addresses these limitations by introducing a dynamic environment space adaptability mechanism that continuously adapts to new material conditions. Unlike static RL models, RL^{2}C updates the Q-table in real time for each environment, allowing the agent to effectively explore and exploit the parameter space across diverse materials and cutting conditions.

Overview of laser cutting process

This section presents a comprehensive overview of the material film structure, the configuration of the input dataset used for training, the measurement techniques used to evaluate the cutting quality, and the formulation of optimization problems.

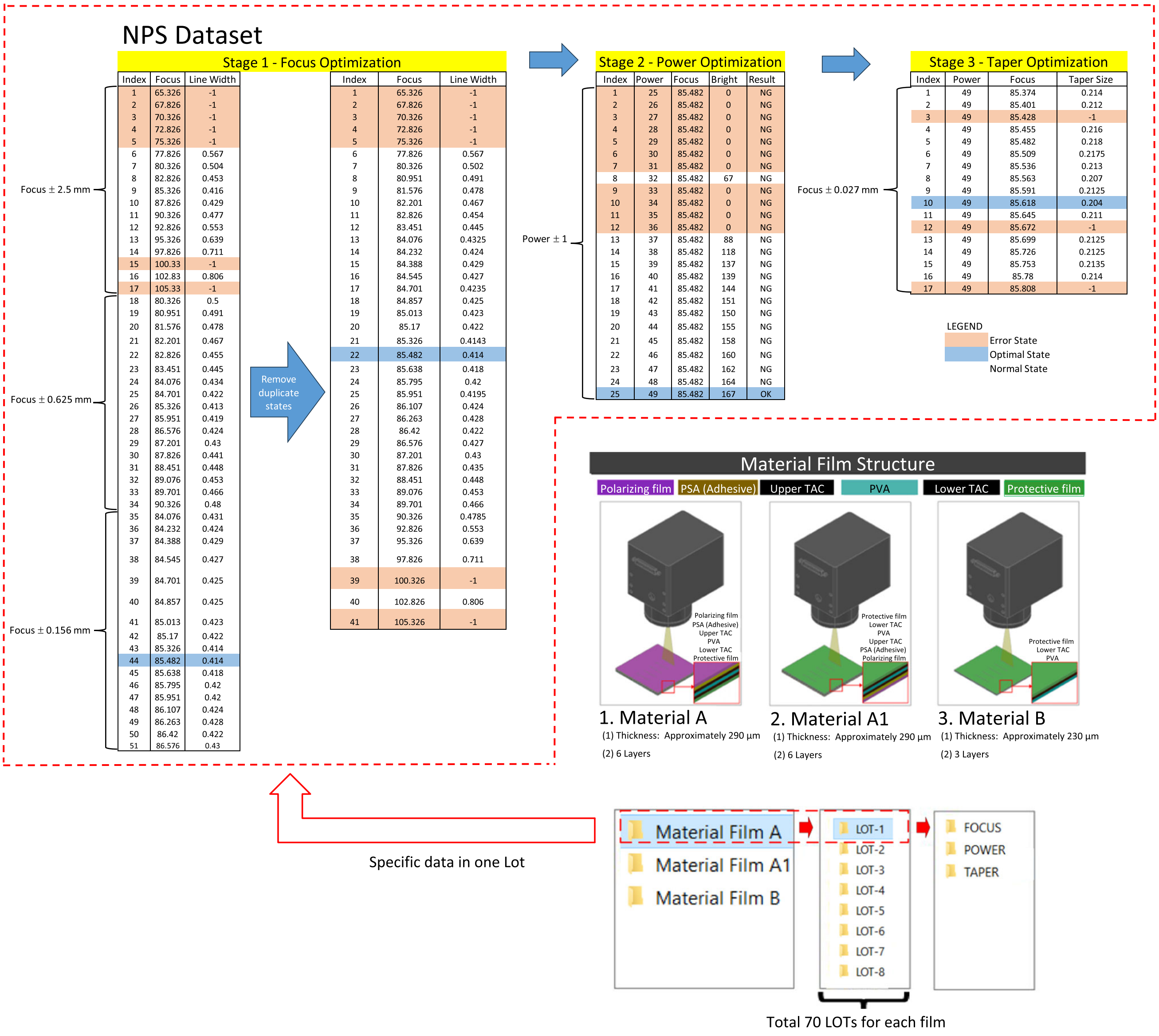
Material film structure and input dataset configuration

Figure 1 illustrates the structure of the material films and the corresponding input dataset configuration used for training the RL^{2}C algorithm.

The bottom part of Fig. 1 presents the data characteristics grouped according to the material. There are three types of material films: Material A, Material A1, and Material B, with each material film having 70 lots for data collection. Each lot represents a unique set of experimental conditions applied to that specific material film. Within each lot, the data are further categorized into three subfolders: **focus**, **power**, and **taper** which correspond to the three optimization stages. These subfolders store the raw measurement and parameter-tuning data associated with each stage of the laser-cutting process.

To clarify how diverse material films influence the learning behavior of RL^{2}C, the middle part of Fig. 1 outlines the physical layer structure of each material film. Material A and A1 consist of six layers, including a polarizing film, Pressure-Sensitive Adhesive (PSA), Upper Tri-Acetyl Cellulose (TAC), Polyvinyl Alcohol (PVA), Lower TAC, and a protective film, with a total thickness of approximately 290 µm. Material B, on the other hand, has a simpler structure, consisting of three layers, including a protective film, lower TAC, and PVA, with a total thickness of approximately 230 µm.

The upper part of Fig. 1 represents the sequential flow of the dataset across the three optimization stages—**Focus Optimization**, **Power Optimization**, and **Taper Optimization**. It provides a detailed example of data collected from a single lot, showcasing the step-by-step progression through each stage of the laser-cutting process. Each row in this upper section corresponds to a trial conducted under a spe-



NPS Dataset

Stage 1 - Focus Optimization

Index	Focus	Line Width
1	65.326	-1
2	67.826	-1
3	70.326	-1
4	72.826	-1
5	75.326	-1
6	77.826	0.567
7	80.326	0.504
8	82.826	0.453
9	85.326	0.416
10	87.826	0.429
11	90.326	0.477
12	92.826	0.553
13	95.326	0.639
14	97.826	0.711
15	100.33	-1
16	102.83	0.806
17	105.33	-1
18	80.326	0.5
19	80.951	0.491
20	81.576	0.478
21	82.201	0.467
22	82.826	0.455
23	83.451	0.445
24	84.076	0.434
25	84.701	0.422
26	85.326	0.413
27	85.951	0.419
28	86.576	0.424
29	87.201	0.43
30	87.826	0.441
31	88.451	0.448
32	89.076	0.453
33	89.701	0.466
34	90.326	0.48
35	84.076	0.431
36	84.232	0.424
37	84.388	0.429
38	84.545	0.427
39	84.701	0.425
40	84.857	0.425
41	85.013	0.423
42	85.17	0.422
43	85.326	0.414
44	85.482	0.414
45	85.638	0.418
46	85.795	0.42
47	85.951	0.42
48	86.107	0.424
49	86.263	0.428
50	86.42	0.422
51	86.576	0.43

Index	Focus	Line Width
1	65.326	-1
2	67.826	-1
3	70.326	-1
4	72.826	-1
5	75.326	-1
6	77.826	0.567
7	80.326	0.502
8	80.951	0.491
9	81.576	0.478
10	82.201	0.467
11	82.826	0.454
12	83.451	0.445
13	84.076	0.4325
14	84.232	0.424
15	84.388	0.429
16	84.545	0.427
17	84.701	0.4235
18	84.857	0.425
19	85.013	0.423
20	85.17	0.422
21	85.326	0.4143
22	85.482	0.414
23	85.638	0.418
24	85.795	0.42
25	85.951	0.4195
26	86.107	0.424
27	86.263	0.428
28	86.42	0.422
29	86.576	0.427
30	87.201	0.43
31	87.826	0.435
32	88.451	0.448
33	89.076	0.453
34	89.701	0.466
35	90.326	0.4785
36	92.826	0.553
37	95.326	0.639
38	97.826	0.711
39	100.326	-1
40	102.826	0.806
41	105.326	-1

Stage 2 - Power Optimization

Index	Power	Focus	Bright	Result
1	25	85.482	0	NG
2	26	85.482	0	NG
3	27	85.482	0	NG
4	28	85.482	0	NG
5	29	85.482	0	NG
6	30	85.482	0	NG
7	31	85.482	0	NG
8	32	85.482	67	NG
9	33	85.482	0	NG
10	34	85.482	0	NG
11	35	85.482	0	NG
12	36	85.482	0	NG
13	37	85.482	88	NG
14	38	85.482	118	NG
15	39	85.482	137	NG
16	40	85.482	139	NG
17	41	85.482	144	NG
18	42	85.482	151	NG
19	43	85.482	150	NG
20	44	85.482	155	NG
21	45	85.482	158	NG
22	46	85.482	160	NG
23	47	85.482	162	NG
24	48	85.482	164	NG
25	49	85.482	167	OK

Stage 3 - Taper Optimization

Index	Power	Focus	Taper Size
1	49	85.374	0.214
2	49	85.401	0.212
3	49	85.428	-1
4	49	85.455	0.216
5	49	85.482	0.218
6	49	85.509	0.2175
7	49	85.536	0.213
8	49	85.563	0.207
9	49	85.591	0.2125
10	49	85.618	0.204
11	49	85.645	0.211
12	49	85.672	-1
13	49	85.699	0.2125
14	49	85.726	0.2125
15	49	85.753	0.2135
16	49	85.78	0.214
17	49	85.808	-1

LEGEND
🟧 Error State
🟦 Optimal State
Normal State

Material Film Structure

1. Material A
(1) Thickness: Approximately 290 μm
(2) 6 Layers

2. Material A1
(1) Thickness: Approximately 290 μm
(2) 6 Layers

3. Material B
(1) Thickness: Approximately 230 μm
(2) 3 Layers

Specific data in one Lot

Total 70 LOTs for each film

Fig. 1 Material film structure and input dataset configuration for RL^2C training

cific parameter setting, while the columns capture both the control variables (e.g., focus value or power value) and the resulting measurements (e.g., line width, brightness, or taper size) for that stage. It is important to note that the upper part of Fig. 1 illustrates the configuration and results from one representative lot only; the same structure applies to all 70 lots per material film, but the specific parameter values and measurements differ depending on the material characteristics and experimental variations. These configurations are directly used in the learning process of RL^2C. Each row in the dataset serves as a state-action-reward tuple, which forms the basis for Q-learning updates across episodes. Based on this data structure, the specific procedures and data handling approaches for each optimization stage are detailed below, beginning with Stage 1.

In Stage 1, Focus Optimization aims to identify the focus value that minimizes the line width. By adjusting the focus value with specific variation sizes (±2.5, ±0.625, ±0.156 mm), the process begins with selecting a range of focus values and testing them iteratively. Then we measure and record the line width for each focus value setting. During this process, each focus value is categorized into states: error state for invalid or failed attempts (e.g., line width = −1), normal state for acceptable but suboptimal results, and optimal state for the minimum line width value. When duplicate focus values are present, their corresponding line width values are computed as the mean to ensure the data are accurate and reliable. For instance, at the end of Stage 1, the optimal focus setting identified is 85.482, resulting in a minimum line width of 0.414.

In Stage 2, by using the optimal focus value obtained from Stage 1 as a fixed input parameter, Power Optimization aims to determine the power value that ensures the laser cut meets a specific brightness threshold. The process involves adjusting the power incrementally (e.g., $\pm 1\%$) and measuring the resulting brightness for each setting. Each power value is tested while keeping the focus constant at the optimal setting identified in Stage 1. We measure and record the brightness, and the power value that achieves the desired brightness is identified as the optimal state (e.g., result = OK). Error states whose brightness equals zero and the result is marked as "NG" (not good) (e.g., brightness = 0 and result = NG), while normal states are the remaining ones (e.g., brightness $\neq 0$ and result = NG). The optimal power value of 49 identified in Stage 2 will be used in the final stage of the optimization process.

In Stage 3, Taper Optimization focuses on minimizing the taper size. Using the optimal focus 85.482 mm and power value of 49 determined from Stages 1 and 2, the focus is fine-tuned within a narrow range (e.g., optimal focus from Stage 1 ± 0.027 mm) to minimize the taper size. Similar to the previous stages, the focus value is incrementally adjusted, and the resulting taper size is measured for each setting. Moreover, each focus value is also categorized into states: error state for taper size = -1, optimal state for the smallest taper size, and normal state for the remaining one. At the end of Stage 3, the optimal focus setting (85.618 mm) that minimizes the taper size is identified and used for production, completing the optimization process.

This three-stage structure forms a consistent learning workflow across all lots and materials. Importantly, the entire pipeline-from data collection to learning-is designed to operate in a fully autonomous manner, with minimal human intervention. Human involvement is limited to loading the film material into the laser cutting machine and specifying the number of lots to be processed. Once initiated, the machine automatically executes all cutting procedures sequentially across the three optimization stages-starting with focus, followed by power, and then taper optimization. For each parameter configuration, the system measures the corresponding cutting quality (e.g., line width, brightness, taper size) and stores the results in structured CSV files. These files are formatted in the same structure as shown in the upper part of Fig. 1, including both the parameter values and their measured outcomes.

Once the generated CSV files from all lots are ready and the environment is initialized, the remaining stages-from state exploration and action selection to Q-table updates and policy refinement-are fully automated. The RL^{2}C agent autonomously interacts with the environment, optimizes parameters across episodes, and dynamically adapts to new states using its dynamic Q-learning mechanism. This division of responsibilities ensures that human oversight remains minimal and is only required during the initial experimental setup. After that, the learning and decision-making processes operate in an autonomous and self-improving manner. This makes the RL^{2}C framework both scalable and practical for real-world manufacturing environments.

Laser marking and cutting quality measurements

Figure 2 presents examples of laser marking and cutting, highlighting key measurement points for assessing the quality of the laser cutting process. In Fig. 2a, the surface of the material film ① is displayed, with the line width ② being measured to evaluate focus optimization. This line width represents the thickness of the black line on the material surface, and it is a critical parameter for ensuring precise marking.

In Fig. 2b, several measurements are shown to optimize power and taper. The cut width ③ measures the thickness of the white region in the middle, representing the fully cut area. Taper 1 ④ and Taper 2 ⑤ refer to the black regions on either side of the cut. The narrower the black region, the better cutting quality. Since taper size can be asymmetric, the sum of Taper 1 and Taper 2 is averaged to determine the overall cut quality.

Several output parameters define the quality of the cut, which are line width, brightness, and taper size. Line width is related to the width of the cutting line made by the laser beam, and it determines the accuracy of the cutting. Brightness is the quality of the cut surface and the intensity of the dark spots left on the material after the cut. Taper size is the difference in the width of the cut between the top and bottom surfaces of the material, affecting the consistency of the cut. The above-mentioned output parameters are controlled to achieve high-quality cutting, although several input parameters like focus and power can affect them. Focus or focal length refers to the distance between the focusing lens and the material being cut, while power refers to the intensity of the laser beam.

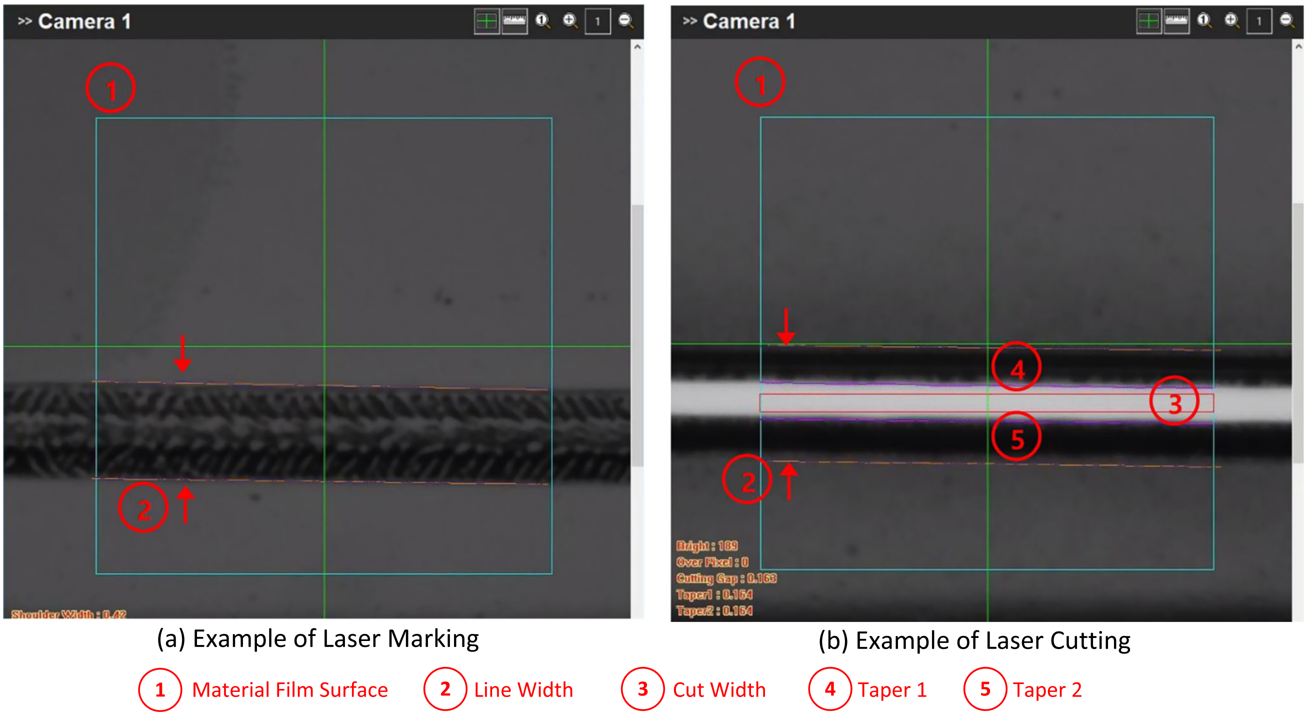
Problem formulation

The optimization of the laser cutting process involves three key stages, each targeting a specific parameter to improve cutting quality. The problem is formulated as a sequential optimization process, where each stage refines a parameter to achieve the best results under given constraints.

Stage 1: focus optimization

Focus Optimization aims to minimize the line width, L_w, which is a function of the focus value, F. The objective function for this stage is:

$$\min_{F} L_w(F) \tag{1}$$



Shoulder Width : 0.42

(a) Example of Laser Marking

Bright : 189
Over Pixel : 0
Cutting Gap : 0.163
Taper1 : 0.164
Taper2 : 0.164

(b) Example of Laser Cutting

Fig. 2 Examples of laser marking and cutting

where F represents the focus value. The constraints for this stage are:

- $F \in \{F_{\text{start}}, F_{\text{start}} \pm \Delta F, \ldots, F_{\text{end}}\}$, where $\Delta F = \pm 2.5$, ± 0.625, ± 0.156 mm.
- $L_w(F) > 0$ and $L_w(F) \neq -1$ (valid line width values).
- Duplicate focus values are removed, and their corresponding line width values are averaged.

Stage 2: power optimization

Power Optimization determines the laser power, P, that meets a specific brightness threshold B_{thresh}. The optimization objective is expressed as:

$$\min_{P} \, (B(P) - B_{\text{thresh}}) \; \textit{subject to} \; B(P) \geq B_{\text{thresh}} \tag{2}$$

The constraints for this stage include:

- $P \in \{P_{\text{start}}, P_{\text{start}} \pm \Delta P, \ldots, P_{\text{end}}\}$, where $\Delta P = \pm 1\,\%$.
- Only power values where $B(P) \geq B_{\text{thresh}}$ are valid.
- Result $(P) = \text{OK}$ for optimal states.

Stage 3: taper optimization

Taper Optimization minimizes the taper size, T_s, using the optimal focus and power values from the previous stages. The optimization objective is:

$$\min_{F} T_s(F_{\text{opt}}, P_{\text{opt}}) \tag{3}$$

with the constraints:

- $F \in \{F_{\text{opt}} \pm \Delta F, \ldots, F_{\text{opt}} + n \cdot \Delta F\}$, where $\Delta F = \pm 0.027$ mm.
- $T_s(F_{\text{opt}}, P_{\text{opt}}) > 0$ and $T_s(F_{\text{opt}}, P_{\text{opt}}) \neq -1$ (valid taper size values).

Overall objective

The overall optimization can be formulated as a multi-stage process, where the combined objective is:

$$\min_{F,P} \left\{ L_w(F), (B(P) - B_{\text{thresh}}), T_s(F_{\text{opt}}, P_{\text{opt}}) \right\} \tag{4}$$

subject to the constraints defined for each stage.

The three optimization stages above are designed to operate sequentially, with each stage building upon the outcome of the previous one. Specifically, Stage 2 takes the optimal focus value F_{opt} obtained from Stage 1 as a fixed input to determine the corresponding optimal power value P_{opt}. Stage 3 then uses both F_{opt} and P_{opt} as inputs to optimize the taper size. Importantly, Stage 3 does not explore the entire focus search space evaluated in Stage 1; instead, it performs a search within a narrow range centered around

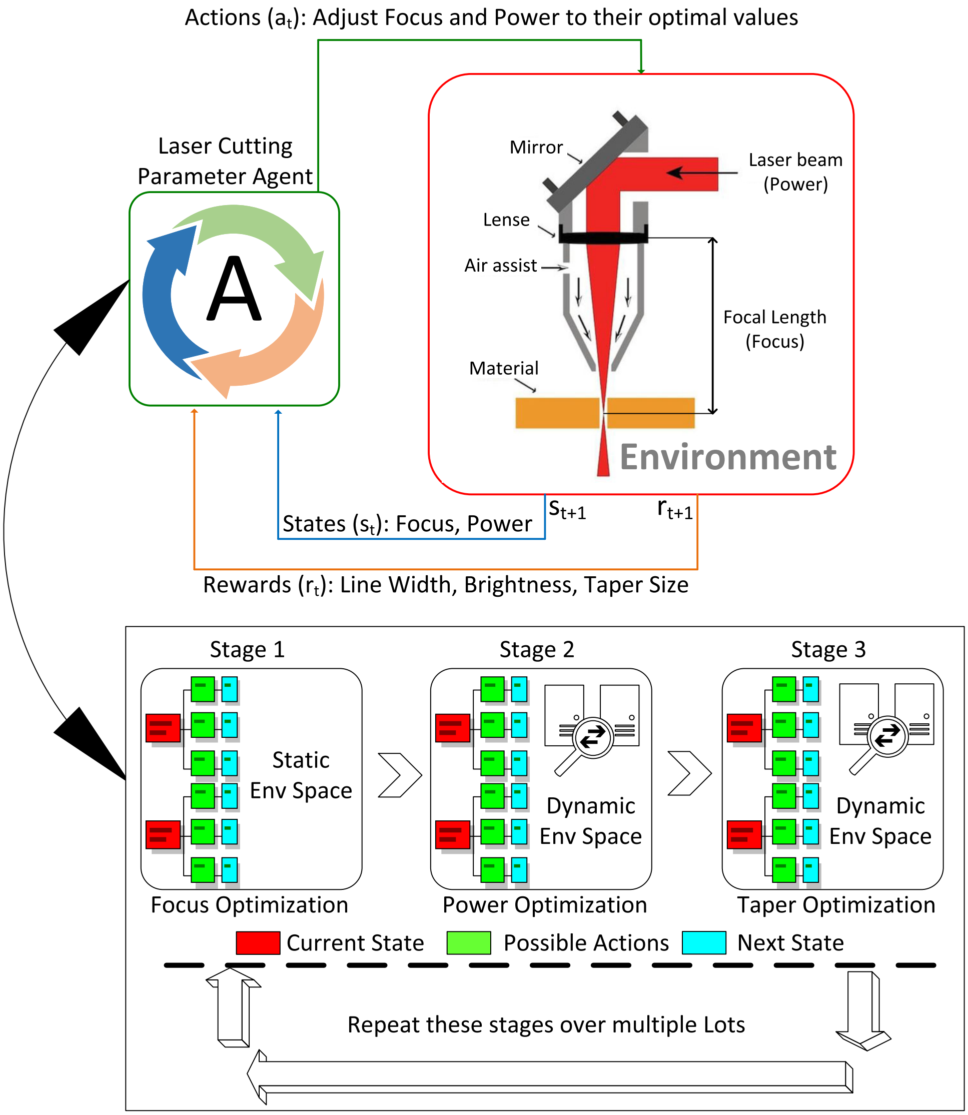


Fig. 3 Reinforcement learning framework for laser cutting (RL^{2}C) parameter optimization

F_{opt} (e.g.,$F_{\text{opt}} \pm \Delta F$). This design makes Stage 3 inherently dependent on the results of both Stage 1 and Stage 2, while also effectively refining the parameter found in Stage 1 to achieve the best possible final quality.

Proposed method

Laser cutting parameter optimization agent

Laser-cutting machines are precision-cutting instruments that utilize high-power laser beams on the material that is to be cut. The quality of the cut is primarily determined by output parameters such as line width, brightness, and taper size. These parameters must be optimized to ensure optimal cutting quality. Among the key factors influencing these outputs are the input parameters, including focus and power, which directly control the cutting process.

Figure 3 presents the architecture of the RL^{2}C framework, which is trained using the structured dataset illustrated in Fig. 1. In this framework, the laser-cutting parameter agent interacts with an environment that simulates the laser-cutting process. The environment receives focus and power values as actions (a_t) and returns the resulting cutting quality metrics: line width, brightness, and taper size as rewards (r_t), along with the next state (s_{t+1}). This process is structured into three stages: Stage 1 (Focus Optimization), Stage 2 (Power Optimization), and Stage 3 (Taper Optimization).

In Stage 1, the agent operates within a static environment space, where the state space remains consistent across all lots. This stage focuses on optimizing the focus parameter under stable conditions, allowing the agent to refine the focus settings effectively.

From Stage 2 to Stage 3, the environment turns into a dynamic one, meaning that the state space varies over lots, presenting new challenges for the agent. In Stage 2, the dynamic environment arises because the optimal power setting required to achieve the desired brightness depends on material-specific characteristics, such as variations in composition and thickness across lots. The agent must adjust the power parameter to account for these differences dynamically, ensuring that the brightness threshold is satisfied across lots.

Algorithm 1 RL^{2}C algorithm

```
Input: $\alpha = 0.9$, $\gamma = 0.9$, $\varepsilon_{max} = 1.0$, $\varepsilon_{min} = 0.1$, $decay\_rate = 0.05$.
Output: Optimal state $s^*$, optimal policy $\pi^*$, optimal Q-table $Q^*$.
procedure RL AGENT
    Initialize Q-Table $Q$ for all state-action pairs.
    for each $material$ do
        for each $lot$ do
            $\varepsilon \leftarrow \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min})e^{-decay\_rate*lot}$
            for each $episode$ do
                $s', r \leftarrow$ Epsilon_Greedy_Policy($\varepsilon, s$)
                Update the Q-values for current state-action pairs using the Bellman Equation in Eq. 5.
                $s \leftarrow s'$; break if $s' == s^*$ or $s' ==$ Error state
            end for
            Update optimal policy $\pi^*(s) \leftarrow \arg\max_a(Q[s, a])$.
            $Q^* \leftarrow$ UPDATE_STATIC_Q_TABLE($Q$) ▷ Stage 1.
            $Q^* \leftarrow$ UPDATE_DYNAMIC_Q_TABLE($Q$) ▷ Stage 2 and 3.
        end for
    end for
    return $\pi^*, Q^*$
end procedure
procedure EPSILON_GREEDY_POLICY($\varepsilon, s$)
    if $random\_uniform(0, 1) < \varepsilon$ then
        $a \leftarrow random\_action()$
    else
        $a \leftarrow \arg\max_a(Q[s, a])$
    end if
    $s' \leftarrow s + a$
    $r \leftarrow$ Reward_Function($s, a, s', s^*$) ▷ Eq. 6 for Stage 1, Eq. 7 for Stage 2, and Eq. 8 for Stage 3.
    return $s', r$
end procedure
procedure UPDATE_STATIC_Q_TABLE($Q$)
    A new static Q-table is updated after each lot using Eq. 5.
end procedure
procedure UPDATE_DYNAMIC_Q_TABLE($Q$)
    if new states exist then
        for each new state do
            Compute possible next states for new states.
            Add more rows in $Q$ for new states.
        end for
        for each existing state that can reach new states do
            Update $Q$ values where transitions are possible.
        end for
    end if
    return $Q^*$
end procedure
```

Similarly, Stage 3 also operates in a dynamic environment, as taper size is influenced by both material properties and the cumulative effects of focus and power parameters selected in the earlier stages. This stage requires the agent to fine-tune the focus setting chosen in Stage 1 while using power adjustments made in Stage 2, as well as material-specific variations in taper behavior. The evolving state space in these stages reflects real-world variability, pushing the agent to adapt its actions dynamically for each lot.

Throughout these stages, the agent learns optimal parameter settings across multiple lots, adjusting its actions based on both static and dynamic environments. This dynamic adaptability is key to achieving high-quality laser cutting in real-world applications, where material properties and conditions can vary significantly across production lots.

RL^{2}C algorithm

The RL^{2}C in Algorithm 1 is designed to optimize laser cutting parameters such as focus and power. It uses Q-learning to train an agent to identify the best setting for high-quality cutting for various material films and lots. In this subsection, the details of the algorithm implementation are discussed, including input and output parameters, the iterative learning process, and its capability to adjust to the static and dynamic changes in the environment during various phases of the optimization.

The RL agent takes as input a set of hyperparameters (Kundroo & Kim, 2023): the learning rate α, the discount factor γ, the exploration-exploitation rate ϵ, and the rate of decay of the exploration parameter. The output of the algorithm is the optimal state s^*, the optimal policy π^*, and the optimal Q-table Q^*, which contains the best laser cutting settings with the corresponding chain of actions required to achieve them.

The RL^{2}C algorithm begins with the initialization of the environment, which determines the states and actions of the laser-cutting process. At this time, the Q-table is populated with zero values or small random positive values for each state-action pair. The agent also initializes its policy π^*, which will guide its actions based on the Q-values. The agent is now ready to explore and interact with the environment.

Each Q-table is structured as a key-value map, where each key is a state-action pair, and the value is the calculated Q-value. The state typically is current parameter values (e.g., focus or power), and the action represents discrete steps to increment or decrement these parameters. This design enables flexible learning across multiple lots, with the agent progressively refining its knowledge in each stage without interference from others. Once the environment and Q-table are set up, the agent starts its learning process when testing with many different materials. For each lot, the agent must decide whether to explore new actions or exploit the best-known actions based on its current knowledge. This decision is made using the ε-greedy policy, responsible for managing the trade-off between exploration and exploitation. At each step, the agent performs an action and obtains a response from the environment in the form of a reward function, which is calculated using line width, brightness, and taper size in Eqs. 6, 7, and 8, respectively.

After each action, the Q-values are updated using the Bellman equation:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') \right] \quad (5)$$

This equation adjusts the Q-value for the state-action pair based on both the immediate reward and the expected future rewards from the next state. After each episode, the agent updates its policy π^*, selecting actions that maximize its expected rewards for each state. The agent repeats this process until it reaches the optimal state s^*, encounters an error state, or gets the maximum number of steps to be performed in one episode. This process helps the agent better understand which of the possible optimal actions will result in the highest reward.

The RL^{2}C algorithm is developed to work for both static and dynamic environments, which represent the different conditions of the laser cutting process. One of the significant characteristics of this algorithm is the Update_Dynamic_Q_table() mechanism with which the agent learns to modify the Q-table in response to new states during training across multiple lots. As new states arise due to changes in material properties or laser cutting conditions, the agent incorporates them into the Q-table by adding new rows and updating transitions from existing states. This ensures that the agent can continue to explore and exploit these newly discovered states, maintaining optimal performance in an evolving environment. By doing so, the Q-table is continuously updated to account for these environmental changes, making the RL^{2}C algorithm robust and adaptable to the dynamic nature of real-world manufacturing scenarios.

To support stage-wise learning and maintain separation between optimization objectives, RL^{2}C maintains a separate Q-table for each stage: one for focus optimization (Stage 1), one for power optimization (Stage 2), and one for taper optimization (Stage 3). The Q-table in Stage 1 operates in a static environment, where the state space is predefined and fixed. In contrast, the Q-tables for Stage 2 and Stage 3 are dynamic; their state spaces evolve based on new input–output observations from each lot, allowing the agent to adapt to changes in material characteristics or cutting conditions.

Once the learning process is done at all of the stages, it returns to the optimal policy π^* and the final Q-table Q^*. These represent the agent's learned strategy for achieving the best laser-cutting results, as well as the expected rewards for each state-action pair. At the end of the process, the agent enhances the knowledge about the environment and identifies the optimal actions for every step taken in all lots.

Detailed design of each stage using RL^{2}C

This subsection describes how to apply the RL^{2}C algorithm in each state space, action space, and reward function.

Stage 1: focus optimization

- **State space** The state space is defined by the current focal length of the laser. Each state represents the system's configuration at a specific moment during the optimization process, with focus adjustments made across three phases. As shown in Fig. 1, in **Phase 1**, the focal length starts at an initial value of 65.326 mm and is adjusted by 2.5 mm increments over 17 iterations. After completing 17 steps, the focus value that satisfies the minimum line width among the results is selected. In **Phase 2**, based on the minimum line width value selected in **Phase 1** scale, optimization is performed again by setting that focus value as the midpoint and incrementing or decrementing by ±0.625 mm in 17 steps. Similarly, the focus value that satisfies the minimum line width is selected. In **Phase 3**, continuing from the last result in **Phase 2** scale, the final optimization is performed in **Phase 3** scale with 17-step adjustments with ±0.156 mm. This state space allows the agent to explore how precise focus adjustments impact the system's performance, as it narrows down the optimal focus setting within these specific focus ranges.
- **Action space** The available actions include both positive and negative adjustments in millimeters: [±2.5, ±0.625, ±0.156,± 0.157]. These actions allow the agent to either increase or decrease the focal length incrementally, depending on the feedback it receives from the environment.
- In our approach, each state that the agent encounters can be classified into one of three categories: **Optimal State**, **Error State**, or **Normal State**, based on the current focal length and the corresponding line width result. The **Optimal State** is the state where the focal length produces the narrowest line width, indicating the best possible performance for the laser marking process. These states are represented by the blue points in the 3D parameter space visualization in Fig. 4, which are concentrated near the optimal focal length range (e.g., around 85.5 mm). The **Error State** occurs when the linewidth cannot be measured properly, typically caused by an incorrect focal length that prevents proper marking (e.g., line width = −1.0). In the 3D parameter space, these states are depicted as red points, scattered outside the optimal range, highlighting regions where the cutting process fails. Lastly, **Normal State** refers to any state where the line width is measurable but not optimal. These states are represented by gray points in the visualization, forming a transition zone between error and

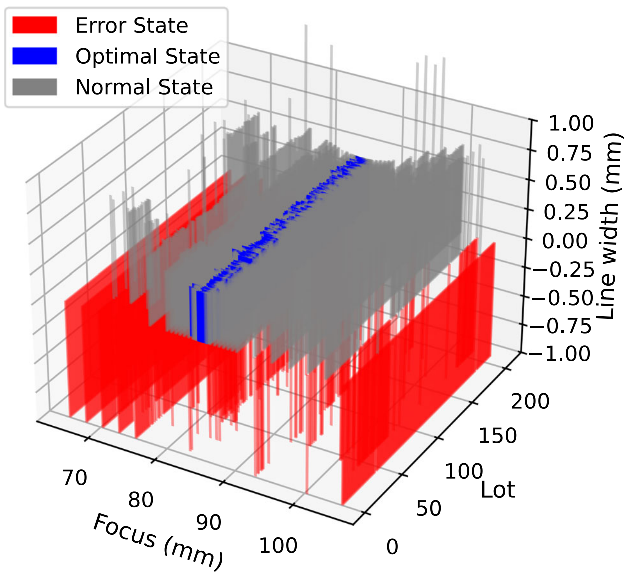


Fig. 4 3D parameter space visualization for Stage 1—Focus Optimization

optimal states. In these states, the agent receives negative feedback, as the focal length does not produce the narrowest possible line, but the process is still functional.

- **Reward function** The reward function guides the agent in selecting the best focal length while penalizing inefficient actions. The reward is defined as:

$$R1 = \begin{cases} -200, \textit{ if Error}, \\ \frac{100}{\textit{line width}} - (step \times 0.1), \textit{ if Optimal}, \\ -\frac{1}{\textit{line width}} - (step \times 0.1), \textit{ if Normal}. \end{cases} \tag{6}$$

In Eq. 6, lower line width values yield higher rewards, incentivizing more precise cuts. For *error states* (e.g., line width = − 1), a fixed penalty of − 200 is assigned to strongly discourage invalid settings. For *optimal states*, defined as the minimum line width observed in the dataset (Fig. 1), the reward scales with $\frac{100}{\textit{line width}}$, typically producing values in the range of approximately 125 to 250 for line width values between 0.4 and 0.8 mm. The *normal states* yield a smaller reward, ranging from approximately 1.25 to 2.5, calculated as $\frac{1}{\textit{line width}}$. In both *optimal* and *normal states*, the reward is slightly penalized by the number of steps taken in the episode via the $(step \times 0.1)$ term, encouraging faster convergence.

Stage 2: power optimization

- **State space** This is defined by the current power setting and the analysis of the cutting result for the current lot. The initial power setting starts at 25%, and the power is incremented by 1% in each step. For each power setting, the system analyzes the cutting image to evaluate the cutting quality. A successful cut is determined based on two conditions: **(1)** no dark spots are present within the cutting area, and **(2)** the average brightness value within the cutting area is 167 or higher. The state space thus consists of the power setting at each step, the absence or presence of dark spots, and the average brightness of the cut.
- **Action space** [±10, ±9, ±8, ±7, ±6, ±5, ±4, ±3, ±2, ±1] %.
- In Stage 2, the **Error State** occurs when the cutting brightness is 0, resulting in a significant penalty. These states are represented by red points in the 3D parameter space visualization in Fig. 5, showing regions where the power settings fail to generate sufficient brightness for a valid cut. The **Optimal State** is reached when the cutting brightness matches the optimal value, indicating a successful cut without dark spots, and the average brightness value is higher or equal to a specific threshold. In the parameter space, these states are depicted as blue points, concentrated in a specific range of power values. And the **Normal state** covers any non-optimal, measurable brightness values. These are represented by gray points in the visualization, forming a transition zone between error and optimal states.
- **Reward function** The reward function for Stage 2 is defined as:

$$R2 = \begin{cases} -5000, \textit{ if Error}, \\ (brightness \times 50) - (step \times 2), \textit{ if Optimal}, \\ -brightness - (step \times 2), \textit{ if Normal}. \end{cases} \tag{7}$$

In Eq. 7, the reward is primarily influenced by the brightness of the laser mark. According to the dataset in Fig. 1, brightness values range from 0 (*error states*) to a maximum of 167 in the optimal configuration. For *optimal states*, the reward can reach up to $167 \times 50 = 8350$ before applying the step penalty, making it highly rewarding. For *normal states*, brightness values are non-zero but insufficient to meet the threshold, resulting in negative rewards ranging roughly from − 1 to − 164. A strong fixed penalty of − 5000 is used for error states where brightness is zero. The step penalty term $(step \times 2)$ discourages prolonged exploration and encourages faster convergence.

Stage 3: taper optimization

- **State space** This is defined by the current focus and power settings, with better focus adjustments applied to optimize the taper value. Starting with the best focus

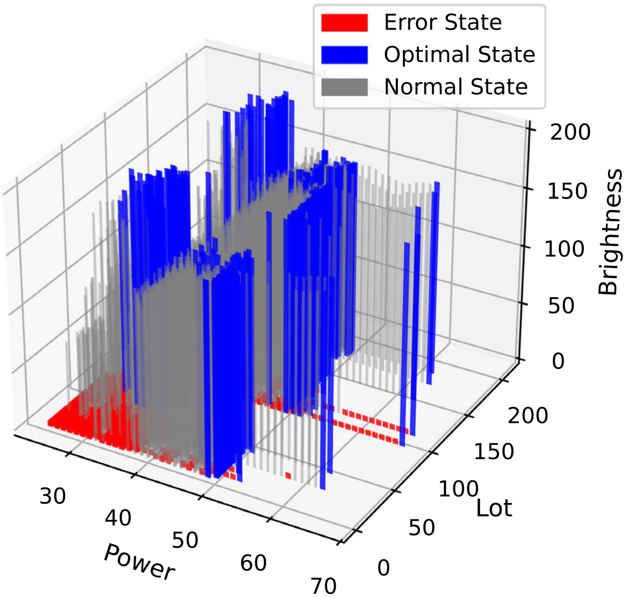


Fig. 5 3D parameter space visualization for Stage 2—Power Optimization

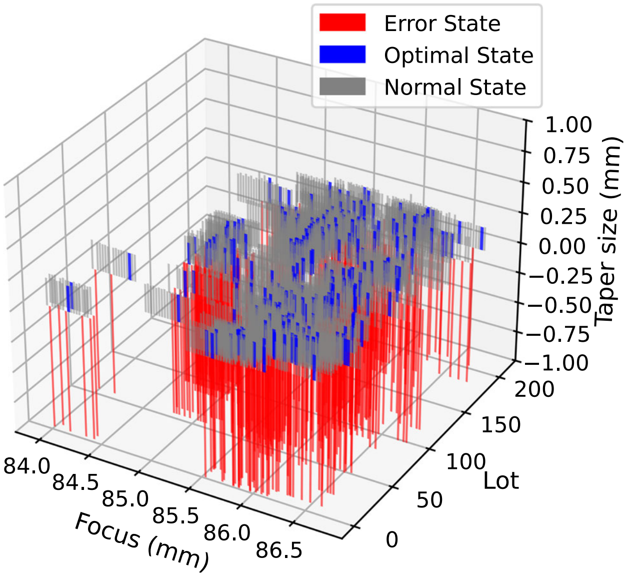


Fig. 6 3D parameter space visualization for Stage 3—Taper Optimization

and power values from the previous stages, the focus is adjusted incrementally by 0.027 mm over 17 iterations. The system evaluates the taper quality by calculating the average of **(Taper1 + Taper2)/2**. A successful cut is determined when the taper result meets the same cutting conditions as in Stage 1: the system returns an "OK" if the taper values are acceptable, indicating a successful cut. The state space consists of the focus adjustments, the calculated taper value, and the result status ("OK" or "NG").

- **Action space** [$\pm 0.109, \pm 0.108, \pm 0.056, \pm 0.055, \pm 0.054, \pm 0.028, \pm 0.027$] mm.
- In Stage 3, the **Error State** occurs when the taper result is invalid (i.e., taper value of -1.0), leading to a penalty. These states are represented by red points in the 3D parameter space visualization in Fig. 6, highlighting regions where the focus adjustments fail to produce valid taper measurements. The **Optimal State** is achieved when the taper result produces the minimum $(Taper1 + Taper2)/2$ value. These states are shown as blue points in the 3D visualization, clustered around specific focus values. The **Normal State** refers to any valid taper value that is not optimal, where the agent continues to adjust focus to reach the optimal taper. These states are depicted as gray points in the visualization, forming a transitional zone between error and optimal states.
- **Reward function** The reward function for Stage 3 is defined as:

$$R3 = \begin{cases} -200, \mathit{if\ Error}, \\ \left(\frac{100}{taper}\right) - (step \times 0.1), \mathit{if\ Optimal}, \\ -\left(\frac{1}{taper}\right) - (step \times 0.1), \mathit{if\ Normal}. \end{cases} \quad (8)$$

Equation 8 is designed to minimize taper size in Stage 3, where the agent performs a focused search around the optimal focus value previously found in Stage 1. In the example shown in Fig. 1, the taper values range from 0.204 mm (*optimal states*) to 0.218 mm (*normal states*), resulting in a maximum theoretical reward of $\frac{100}{0.204} \approx 490$ for the best-performing configuration. In contrast, *normal states* result in small negative rewards; for example, taper = 0.214 yields $-\frac{1}{0.214} \approx -4.67$. *Error states* (e.g., taper = – 1) are assigned a fixed penalty of – 200. However, in other lots where the optimal focus range is different, the taper values and the reward bounding may shift accordingly.

Experimental results

Experimental setup

This subsection explains the experimental setup for our evaluation. We have a total of 210 lots, with 90 for training and 120 for testing. The agent is trained on each material sequentially, starting with Material A, followed by A1, and then B, using the ε-greedy policy across these lots. For each lot, the agent runs 100 episodes. In each episode, the agent is allowed a maximum of 100 steps to find the shortest path to the optimal state. During the training phase, the ε value starts at 1.0 and gradually decreases over the training lots, reaching 0.1. This means that the ε value remains constant within each lot and only reduces when moving to the next lot, encouraging more exploration in the early lots and more exploitation of learned knowledge in later lots. Once the training is complete, the agent is evaluated on the remaining 120 testing lots with a fixed ε of 0.1, ensuring that the agent primarily relies on the learned policies while allowing limited exploration. This setup provides a comprehensive evaluation of the agent's performance across different materials, ensuring effectiveness for optimizing focus, power, and taper settings.

Baselines

To evaluate the performance of the proposed RL^{2}C algorithm, we compare it with three baseline methods: **Random Search**, **RL-based Bayesian Optimization** and **RL-based PSO**. These methods provide straightforward and probabilistic approaches for optimization, serving as benchmarks for assessing the effectiveness of our proposed method.

Random search Random Search is a stochastic optimization technique in which the agent selects actions uniformly at random from the defined action space without employing any structured strategy or policy. At each decision step, the agent randomly selects actions in a purely probabilistic manner, with no consideration of previous experiences or rewards, providing a straightforward baseline method.

RL-based Bayesian optimization (Chang et al., 2024) RL-based Bayesian Optimization integrates RL principles with Bayesian search strategies, leveraging a probabilistic model to guide the action selection process. The agent considers only a few actions to take according to their experience and then chooses the action that would give the maximum expected cumulative reward using the Gaussian process.

RL-based PSO (Zhang et al., 2023) Inspired by social behaviors in nature like birds flocking or fish schooling, PSO consists of a swarm of candidate solutions, or particles, in the action space. The agent's actions are adapted according to its own experiences and the results of other particles in the neighborhood to optimize decision-making.

Main results

Step efficiency analysis over specific lots

Figure 7 illustrates the number of steps required to find the optimal state of the proposed method across different lot positions (first, medium, and final) during the training phase for three distinct materials (A, A1, and B). The figures show the number of steps to reach the optimal state (blue bars) and the number of steps to reach the error state (red bars) over 100 episodes in each lot.

For material A in Fig. 7a, the result presents a very high number of steps across episodes. For instance, there are some episodes that take up to 100 steps to find the optimal state, and there are also episodes where the agent reaches an error state during the learning process. This suggests that the agent needs to perform many possible actions in the exploration phase to gain a better understanding of the environment. During this period, the agent may not be able to find the optimal state or may encounter error states. However, after obtaining sufficient knowledge about the current environment, the number of steps has been quickly reduced to under 40 steps in Fig. 7b and under 20 steps in Fig. 7c. This reduction in the number of steps indicates that the agent has successfully learned the optimal policies and can exploit them more efficiently, resulting in a faster convergence towards the optimal state.

For material A1 in Fig. 7d, the agent exhibits a more consistent and stable convergence rate, with fewer spikes and lower variability in the number of steps across episodes. This behavior is particularly evident in the medium and final lots, as shown in Fig. 7e and f. Similarly, for material B in Fig. 7g–i, the figures demonstrate an immediate convergence rate, with the agent requiring a very low number of steps to reach the optimal state across episodes. This trend is consistent across all three lots, implying that the agent has successfully learned the optimal policies and can exploit them efficiently for most of the lots in Material B.

Step efficiency analysis across different stages and materials

Figure 8 illustrates the average number of steps required by various optimization methods to reach the optimal state across three stages: Focus Optimization, Power Optimization, and Taper Optimization. The results are reported for three different materials (Material A, Material A1, and Material B) during both the training and testing phases. The comparison includes the proposed RL^{2}C method and three baseline methods: Random Search, RL-based Bayesian Optimization, and RL-based PSO.

As shown in Fig. 8a, during the focus optimization stage, the RL^{2}C method consistently requires fewer steps to reach the optimal state compared to the baseline methods. For all materials tested, RL^{2}C demonstrates rapid convergence, particularly during the training phase, where it stabilizes around 5 steps by the midpoint of the training episodes. In contrast, Random Search exhibits significant variability throughout both the training and testing phases, often requiring more than 15 steps to achieve optimization. Bayesian Optimization and PSO perform moderately, with PSO showing slightly more consistency than Bayesian Optimization, but both remain less

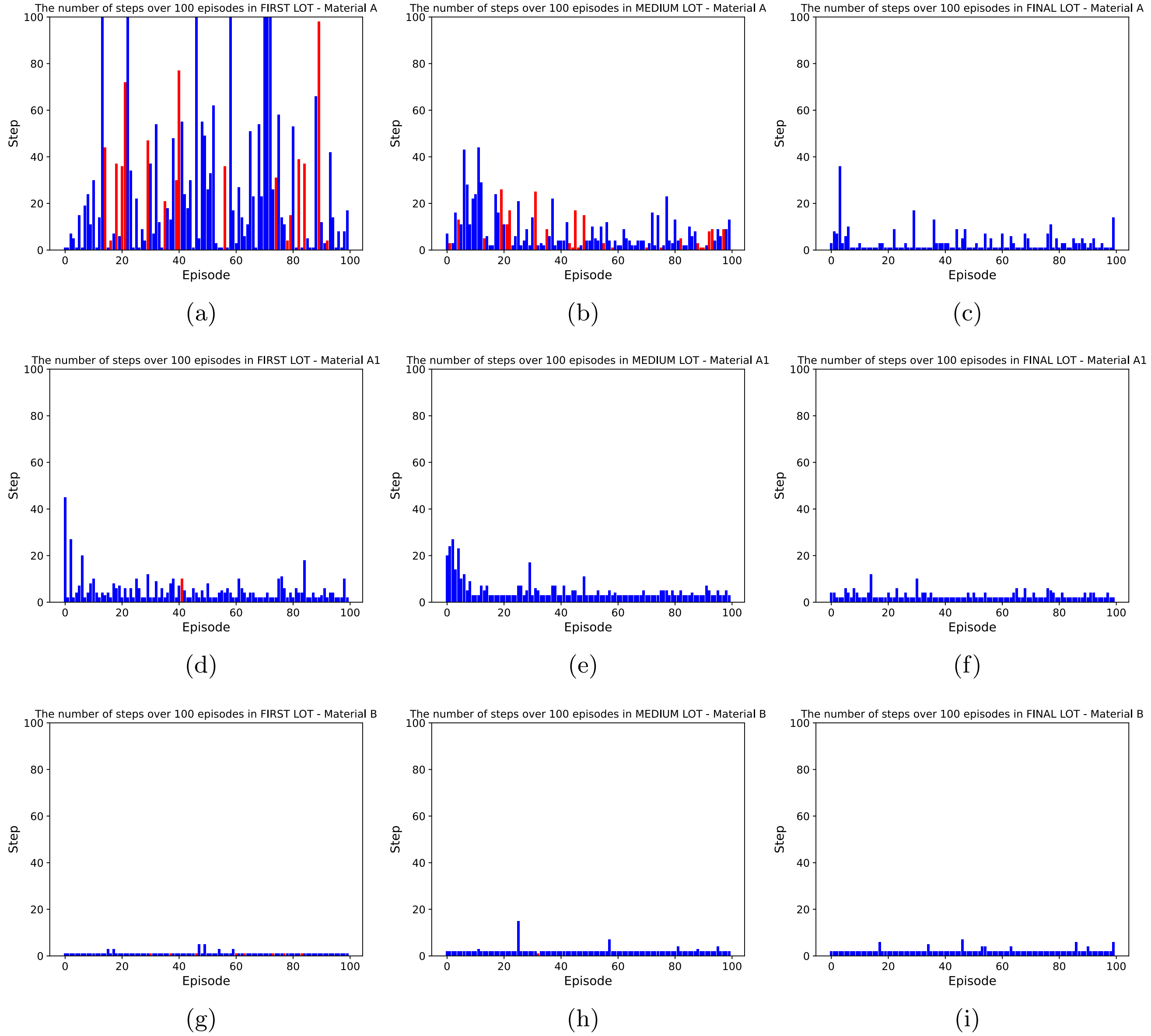
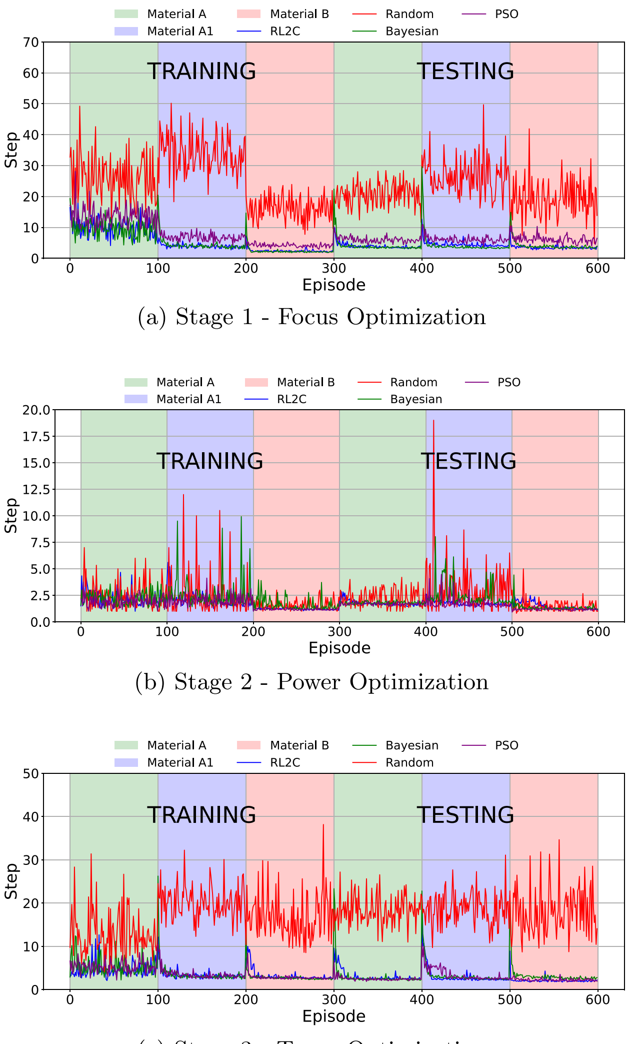


Fig. 7 Steps in first (**a**), medium (**b**), and final (**c**) lot in material A; Steps in first (**d**), medium (**e**), and final (**f**) lot in material A1; and Steps in first (**g**), medium (**h**), and final (**i**) lot in material B

efficient than RL^{2}C. A noticeable pattern across all methods is the sudden increase in the number of steps required when the agent transitions to a new material (e.g., from Material A to A1 or from A1 to B). This spike occurs at the start of each new material phase, indicating the challenge of adapting to previously unseen conditions. However, RL^{2}C quickly recovers from this situation, showcasing great adaptability compared to the baseline methods, which exhibit slower recovery and higher variability.

In Fig. 8b, the step efficiency for the power optimization stage shows a similar trend. RL^{2}C outperforms the baseline methods across all materials in both the training and testing phases. During training, RL^{2}C converges to the optimal state within approximately 3 to 5 steps on average for all three materials, indicating a high level of efficiency in finding the parameter space. Conversely, Random Search requires significantly more steps, fluctuating between 10 and 20 steps throughout the episodes. Bayesian Optimization and PSO demonstrate better performance compared to Random Search but still require more steps on average compared to RL^{2}C. The results for the taper optimization stage are presented in Fig. 8c. Similar to the previous stages, RL^{2}C consistently outperforms the baseline methods in terms of step efficiency. During training, RL^{2}C reaches the optimal state within 2 to 4 steps on average, with minimal variability across different episodes.



(a) Stage 1 - Focus Optimization

(b) Stage 2 - Power Optimization

(c) Stage 3 - Taper Optimization

Fig. 8 Step comparison of RL^{2}C and baseline methods across three optimization stages

Across all three optimization stages, RL^{2}C demonstrates a clear advantage in terms of step efficiency, particularly in adapting to new materials. The sudden increases in step count observed during material transitions are a critical challenge in real-world industrial processes. The ability of RL^{2}C to quickly adapt and stabilize after these transitions highlights its robustness and practical applicability in dynamic environments, where material properties may vary significantly.

Performance of RL^{2}C in step and time efficiency over baseline methods

Table 1 presents a performance comparison of the RL^{2}C algorithm with baseline methods (RL-Bayesian, RL-PSO, and Random Search) in terms of step and time efficiency across three stages of optimization (Stage 1, Stage 2, and Stage 3). Both training and testing phases are evaluated, with the most efficient performances for each metric and stage highlighted in bold to signify optimal results among all compared methods; these results underscore the significant advantages of RL^{2}C.

For step efficiency, during the training phase, RL^{2}C consistently achieves the lowest average number of steps across all three stages. In Stage 1, RL^{2}C requires an average of 5.1 steps compared to 5.4 for RL-Bayesian and 25.3 for Random Search. Similarly, in Stage 2 and Stage 3, RL^{2}C maintains superior step efficiency with averages of 1.6 and 3.8 steps, respectively. The algorithm demonstrates strong performance in the testing phase as well, with lower average steps in all stages, further confirming its effectiveness in converging to optimal states with minimal exploration.

For time efficiency, RL^{2}C shows a significant improvement over the baseline methods in both training and testing phases. In the training phase, RL^{2}C completes Stage 1 with an average time of 0.03 s, which is significantly faster than RL-Bayesian (4.77 s) and RL-PSO (0.11 s). Similar trends are observed in Stages 2 and 3, where RL^{2}C's average times for both stages are 0.02 s. Notably, RL-Bayesian exhibits the highest time cost, particularly in Stage 2, where it requires an average of 11.23 s. In the testing phase, RL^{2}C maintains superior time efficiency, completing all stages in approximately 0.02 s on average, overcoming RL-Bayesian and RL-PSO.

The result demonstrates that RL^{2}C is not only capable of efficiently exploring and exploiting the parameter space but also exhibits superior generalization to new conditions, as evidenced by its consistent performance across both training and testing phases. Its ability to achieve optimal states with fewer steps and significantly lower computational time highlights its suitability for real-time optimization tasks, making it a robust solution for laser cutting parameter optimization. The consistent performance across all stages and phases further emphasizes its adaptability and scalability in diverse scenarios.

Conclusion

This study presents a comprehensive evaluation of the proposed RL^{2}C algorithm for optimizing laser cutting parameters across three vital stages: Focus Optimization, Power Optimization, and Taper Optimization. The results demonstrate that RL^{2}C consistently outperforms baseline methods, including Random Search, RL-based Bayesian Optimization, and RL-based PSO, in terms of step efficiency, time efficiency, and adaptability to dynamic environments. Across all tested materials (Material A, Material A1, and Material B), RL^{2}C exhibits faster convergence, maintaining low execution times and requiring fewer steps to reach the optimal state.

Table 1 Performance of RL^{2}C compared to baseline methods in step and time efficiency (in seconds)

Step efficiency	Stage 1			Stage 2			Stage 3		
Training (ε=1→0.1)	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
RL^{2}C	**5.1**	**17.2**	**1.8**	**1.6**	**5.6**	**1.0**	**3.8**	**12.7**	**2.3**
RL-Bayesian	5.4	18.0	1.9	2.2	9.9	1.0	3.8	26.2	2.1
RL-PSO	8.2	28.1	2.9	1.7	4.1	1.0	3.9	10.5	2.2
Random Search	25.3	50.2	7.8	7.2	12.0	1.0	16.6	38.2	4.1
Testing (ε=0.1)	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
RL^{2}C	**4.0**	**12.6**	**2.7**	**1.7**	**2.8**	**1.1**	**2.8**	**15.3**	**1.8**
RL-Bayesian	4.3	29.6	3.0	1.9	8.0	1.1	3.2	23.3	2.1
RL-PSO	6.1	19.9	3.2	1.5	4.5	1.0	4.4	10.7	1.8
Random Search	22.2	49.7	4.8	11.3	19.0	1.2	18.4	34.6	8.7

Time efficiency	Stage 1			Stage 2			Stage 3		
Training (ε=1→0.1)	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
RL^{2}C	**0.03**	**0.10**	**0.01**	**0.02**	**0.09**	**0.01**	**0.02**	**0.10**	**0.01**
RL-Bayesian	4.77	9.41	0.10	11.23	40.36	0.02	8.44	41.85	0.01
RL-PSO	0.11	0.20	0.05	0.09	0.81	0.01	0.10	0.24	0.01
Random Search	0.08	0.16	0.02	0.02	0.08	0.01	0.05	0.20	0.01
Testing (ε=0.1)	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min
RL^{2}C	**0.02**	**0.08**	**0.01**	**0.02**	**0.08**	**0.01**	**0.02**	**0.11**	**0.01**
RL-Bayesian	5.77	19.97	0.02	13.22	37.83	0.01	10.02	50.90	0.01
RL-PSO	0.11	0.29	0.06	0.13	1.41	0.01	0.11	1.09	0.01
Random Search	0.06	0.16	0.02	0.02	0.08	0.01	0.06	0.18	0.01

A key strength of RL^{2}C lies in its ability to handle transitions between materials efficiently. While baseline methods, particularly Bayesian Optimization, suffer from significant time and step overheads during these transitions, RL^{2}C adapts rapidly with minimal computational cost. This adaptability underscores its robustness and practicality for real-world applications, where material properties can vary significantly between production lots.

In summary, RL^{2}C emerges as a highly efficient and scalable solution for laser cutting parameter optimization, demonstrating optimal performance in both static and dynamic environments. These findings pave the way for its deployment in industrial settings, where efficiency, consistency, and adaptability are important. Future work will explore extending RL^{2}C to additional manufacturing processes and integrating it with other advanced learning frameworks to further enhance its applicability and scalability.

Author contributions Quan Khanh Pham conducts the experiments, coding, and manuscript writing. Majid Kundroo assists in the coding and manuscript preparation. Geunwoo Ban, Seongho Bae, and Taehong Kim provide technical guidance, contribute to experimental design, revise the manuscript, and supervise the research.

Funding This work was supported by the Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2025-RS-2020-II201462).

Data availability The NPS dataset used in this study was acquired from the NPS, and they have not given their permission for researchers to share their data. Data requests can be made to the NPS via this email: nps@npstech.co.kr

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Behbahani, R., Yazdani Sarvestani, H., Fatehi, E., Kiyani, E., Ashrafi, B., Karttunen, M., & Rahmat, M. (2022). Machine learning-driven process of alumina ceramics laser machining. *Physica Scripta, 98*(1), Article 015834. https://doi.org/10.1088/1402-4896/aca3da

Chan, Y.-J., Yuan, T.-H., Sun, H.-C., & Lin, T.-C. (2016). Characterization and exposure assessment of odor emissions from laser cutting of plastics in the optical film industry. *Aerosol and Air Quality Research, 16*(9), 2216–2226. https://doi.org/10.4209/aaqr.2016.04.0136

Chang, C.-Y., Feng, Y.-W., Rawat, T. S., Chen, S.-W., & Lin, A. S. (2024). Optimization of laser annealing parameters based on Bayesian reinforcement learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-024-02363-w

Chen, J., Wang, X., Yue, C., & Zuo, T. (2005). The influence of laser head movement on 3D cutting. *Lasers in Material Processing and Manufacturing II, 5629*, 232–236. https://doi.org/10.1117/12.568956

Dixin, G., Jimin, C., & Yuhong, C. (2006). Laser cutting parameters optimization based on artificial neural network. In *The 2006 IEEE international joint conference on neural network proceedings* (pp. 1106–1111). https://doi.org/10.1109/IJCNN.2006.246813

Genna, S., Menna, E., Rubino, G., & Tagliaferri, V. (2020). Experimental investigation of industrial laser cutting: The effect of the material selection and the process parameters on the kerf quality. *Applied Sciences, 10*(14), 4956. https://doi.org/10.3390/app10144956

Ghany, K. A., & Newishy, M. (2005). Cutting of 1.2mm thick austenitic stainless steel sheet using pulsed and CW Nd:YAG laser. *Journal of Materials Processing Technology, 168*(3), 438–447. https://doi.org/10.1016/j.jmatprotec.2005.02.251

Kundroo, M., & Kim, T. (2023). Efficient federated learning with adaptive client-side hyper-parameter optimization. In *2023 IEEE 43rd international conference on distributed computing systems (ICDCS)* (pp. 973–974). https://doi.org/10.1109/ICDCS57875.2023.00103

Kuprikov, E., Kokhanovskiy, A., Serebrennikov, K., & Turitsyn, S. (2022). Deep reinforcement learning for self-tuning laser source of dissipative solitons. *Scientific Reports, 12*(1), 7185. https://doi.org/10.1038/s41598-022-11274-w

Li, H., Xu, Z., Pi, J., & Zhou, F. (2020). Precision cutting of the molds of an optical functional texture film with a triangular pyramid texture. *Micromachines, 11*(3), 248. https://doi.org/10.3390/mi11030248

Liu, J., Ye, J., Silva Izquierdo, D., Vinel, A., Shamsaei, N., & Shao, S. (2023). A review of machine learning techniques for process and performance optimization in laser beam powder bed fusion additive manufacturing. *Journal of Intelligent Manufacturing, 34*(8), 3249–3275. https://doi.org/10.1007/s10845-022-02012-0

Mamond, A. W., Kundroo, M., Yoo, S.-E., Kim, S., & Kim, T. (2025). FLDQN: Cooperative multi-agent federated reinforcement learning for solving travel time minimization problems in dynamic environments using sumo simulation. *Sensors, 25*(3), 911. https://doi.org/10.3390/s25030911

Marimuthu, S., Eghlio, R., Pinkerton, A., & Li, L. (2013). Coupled computational fluid dynamic and finite element multiphase modeling of laser weld bead geometry formation and joint strengths. *Journal of Manufacturing Science and Engineering, 135*, Article 011004. https://doi.org/10.1115/1.4023240

Mi, J., Zhang, Y., Li, H., Shen, S., Yang, Y., Song, C., Zhou, X., Duan, Y., Lu, J., & Mai, H. (2023). In-situ monitoring laser based directed energy deposition process with deep convolutional neural network. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01820-0

Park, H. S., Nguyen, D. S., Le-Hong, T., & Van Tran, X. (2022). Machine learning-based optimization of process parameters in selective laser melting for biomedical applications. *Journal of Intelligent Manufacturing, 33*(6), 1843–1858. https://doi.org/10.1007/s10845-021-01773-4

Peirovi, S., Pourasghar, M., Nejad, A. F., & Hassan, M. A. (2017). A study on the different finite element approaches for laser cutting of aluminum alloy sheet. *The International Journal of Advanced Manufacturing Technology, 93*, 1399–1413. https://doi.org/10.1007/s00170-017-0599-0

Pramanik, D., Roy, N., Kuar, A., Sarkar, S., & Mitra, S. (2022). Experimental investigation of sawing approach of low power fiber laser cutting of titanium alloy using particle swarm optimization technique. *Optics & Laser Technology, 147*, Article 107613. https://doi.org/10.1016/j.optlastec.2021.107613

Ren, X., Fan, J., Pan, R., & Sun, K. (2023). Modeling and process parameter optimization of laser cutting based on artificial neural network and intelligent optimization algorithm. *The International Journal of Advanced Manufacturing Technology, 127*(3), 1177–1188. https://doi.org/10.1007/s00170-023-11543-6

Singh, Y., Singh, J., Sharma, S., Sharma, A., & Singh Chohan, J. (2022). Process parameter optimization in laser cutting of coir fiber reinforced epoxy composite–A review. *Materials Today: Proceedings, 48*, 1021–1027. https://doi.org/10.1016/j.matpr.2021.06.344

Tercan, H., Khawli, T. A., Eppelt, U., Büscher, C., Meisen, T., & Jeschke, S. (2017). Improving the laser cutting process design by machine learning techniques. *Production Engineering, 11*(2), 195–203. https://doi.org/10.1007/s11740-017-0718-7

Tsai, M.-J., Li, C.-H., & Chen, C.-C. (2008). Optimal laser-cutting parameters for QFN packages by utilizing artificial neural networks and genetic algorithm. *Journal of Materials Processing Technology, 208*(1–3), 270–283. https://doi.org/10.1016/j.jmatprotec.2007.12.138

Ürgün, S., Yigit, H., Fidan, S., & Sinmazçelik, T. (2024). Optimization of laser cutting parameters for PMMA using metaheuristic algorithms. *Arabian Journal for Science and Engineering, 49*(9), 12333–12355. https://doi.org/10.1007/s13369-023-08627-6

Wahab, H., Jain, V., Tyrrell, A. S., Seas, M. A., Kotthoff, L., & Johnson, P. A. (2020). Machine-learning-assisted fabrication: Bayesian optimization of laser-induced graphene patterning using in-situ raman analysis. *Carbon, 167*, 609–619. https://doi.org/10.1016/j.carbon.2020.05.087

Wang, H., Li, B., Zhang, S., & Xuan, F. (2025). Traditional machine learning and deep learning for predicting melt-pool cross-sectional morphology of laser powder bed fusion additive manufacturing with thermographic monitoring. *Journal of Intelligent Manufacturing, 36*(3), 2079–2104. https://doi.org/10.1007/s10845-024-02356-9

Xu, H., Jun, H., & Wu, W. (2014). Optimization of 3D laser cutting head orientation based on the minimum energy consumption. *The International Journal of Advanced Manufacturing Technology, 74*, 1283–1291. https://doi.org/10.1007/s00170-014-6080-4

Zhang, W., Geng, H., Li, C., Gen, M., Zhang, G., & Deng, M. (2023). Q-learning-based multi-objective particle swarm optimization with local search within factories for energy-efficient distributed flow-shop scheduling problem. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-023-02227-9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.