



Optimization of laser annealing parameters based on bayesian reinforcement learning

Chung-Yuan Chang¹ · Yen-Wei Feng¹ · Tejender Singh Rawat¹ · Shih-Wei Chen² · Albert Shihchun Lin¹ 

Received: 3 August 2023 / Accepted: 28 February 2024
© The Author(s) 2024

Abstract

Developing new semiconductor processes consumes tremendous time and cost. Therefore, we applied Bayesian reinforcement learning (BRL) with the assistance of technology computer-aided design (TCAD). The fixed or variable prior BRL is tested where the TCAD prior is fixed or is changed by the experimental sampling and decays during the entire RL procedure. The sheet resistance (R_s) of the samples treated by laser annealing is the optimization target. In both cases, the experimentally sampled data points are added to the training dataset to enhance the RL agent. The model-based experimental agent and a model-free TCAD Q-Table are used in this study. The results of BRL proved that it can achieve lower R_s minimum values and variances at different hyperparameter settings. Besides, two action types, i.e., point to state and increment of levels, are proven to have similar results, which implies the method used in this study is insensitive to the different action types.

Keywords Laser annealing · Bayesian statistics · Reinforcement learning · Semiconductor

Introduction

Due to the scaling of silicon technology, the process tuning and optimization becomes critical in the advanced technology node. This work uses laser annealing as the case study for Bayesian reinforcement learning (BRL). As far as semiconductor annealing is concerned, traditional furnace annealing leads to a high thermal budget, while rapid thermal annealing (RTA) is generally full-wafer, non-localized. As a result, laser annealing is a promising alternative, and it possesses several advantages. First, the localized heating in laser annealing results in less dopant diffusion, which leads to a shaper junction profile and reduces the subthreshold leakage current (Gluschenkov & Jagannathan, 2018; Robinson, 1978; Takamura et al., 2002; Whelan et al., 2002). A low thermal budget is important in monolithic 3D integrated circuits in terms of reduced heat transfer to bottom layers. Controlling the shielding layer's refractive index can confine the

heated region and prevent metal interconnects or devices from being destroyed (Pey & Lee, 2018; Rajendran et al., 2007). Second, the solidification after laser annealing only takes several microseconds, and thus, the dopant diffusion is highly alleviated (Pey & Lee, 2018). This enables the doping concentration to exceed the solid solubility, improving the film's conductivity (Gluschenkov & Jagannathan, 2018; Zhang et al., 2006). Third, laser annealing can be conducted in the air because the impurities in the atmosphere hardly diffuse into the samples due to the fast solidification in laser annealing (White et al., 1979).

Parameter tuning is inevitable to optimize the laser annealing process or, in general, any semiconductor processes. In the new age of the industrial revolution, i.e., Industry 4.0, industries have developed strategic guidelines to facilitate process control and optimization by using emerging technologies such as big data analytics, cyber-physical systems (CPS), data science, cloud computing, and Internet of things (IoT) (Kotsiopoulos et al., 2021; Wang et al., 2018). The concept of intelligent manufacturing (IM) refers to the application of artificial intelligence (AI), IoT, and sensors in manufacturing to make wise decisions through real-time communication (Yao et al., 2017). It is also desired to adopt new methodologies for better-optimized design, manufacturing quality, and production in modern factories (Kusiak,

✉ Albert Shihchun Lin
httd5746@gmail.com

¹ Institute of Electronics Engineering, National Yang-Ming Chiao-Tung University, Hsinchu 30010, Taiwan

² Taiwan Semiconductor Research Institute, No. 26, Prosperity Road I, Hsinchu Science Park, Hsinchu City 300091, Taiwan

1990). To achieve the purpose of fully automated, optimized manufacturing, the keys are machine learning (ML) and deep learning (DL) algorithms that can help develop strategies to automatically analyze, diagnose, and predict patterns from high-dimensional data. Specific to the semiconductor industry, where the processes are complex, interwoven, and sensitive to process parameters, ML provides continuous quality improvement (Li & Huang, 2009; Monostori et al., 1998; Pham & Afify, 2005; Rawat et al., 2023). Supervised learning (SL) and unsupervised learning (USL) have been extensively used in manufacturing industries for process monitoring, control, optimization, fault detection, and prediction (Alpaydin, 2020; Çaydaş & Ekici, 2012; Gardner & Bicker, 2000; Khanzadeh et al., 2017; Li et al., 2019; Pham & Afify, 2005; Salahshoor et al., 2010; Susto et al., 2015). Nevertheless, SL/USL possesses some drawbacks, such as being more limited to the specific process of the entire manufacturing system (Doltsinis et al., 2012).

In contrast to SL/USL, reinforcement learning (RL) is another ML approach where no supervision is mandatory for training the model (Sutton & Barto, 2018). RL is a well-known approach that was initially used in control and prediction problems such as autonomous driving and go games. It has also been used in many optimization problems (Jacobs et al., 2021; Ruvolo et al., 2008; Wang et al., 2020). Specifically, in manufacturing industries, the production environment is often dynamic and non-deterministic, with unexpected scenarios or incidents (Monostori et al., 2004). In such a stochastic environment where the randomness in the dataset makes prediction difficult, an RL agent is more capable of learning the manufacturing process than SL/USL methods (Guevara et al., 2018; Günther et al., 2016; He et al., 2022; Hourfar et al., 2019; Kormushev et al., 2010; Silver et al., 2017). RL has been used in scheduling and dispatching in the semiconductor industry in literature. In particular, Washneck et al. used deep RL for production scheduling and realized optimization and decentralized self-learning (Stricker et al., 2018). Stricker et al. presented deep RL usability in semiconductor dispatching and demonstrated improved system performance (Stricker et al., 2018). Khader et al. use RL to tune the surface mount technology (SMT) in printed circuit boards (PCB) with experimental data (Khader & Yoon, 2021). Besides, some semiconductor process-control efforts have been using RL on a more theoretical side (Khakifirooz et al., 2021; Li et al., 2021; Pradeep & Noel, 2018). While RL in the semiconductor industry has been used extensively for production control (Altenmüller et al., 2020) or scheduling (Lee & Lee, 2022; Luo, 2020; Park et al., 2020; Shi et al., 2020; Waschneck et al., 2018), there have been fewer works using RL in intelligent semiconductor manufacturing, especially compared to the efforts in SL/USL. The advantage of using RL in the

semiconductor industry is that, in general, RL can learn complex and dynamic problems with reasonable generalizability and efficient data utilization (Silver et al., 2017; Wiering & Otterlo, 2012). In addition, RL has the advantage of splitting the primary task into several subtasks, resulting in a flexible, decentralized structure that facilitates computation parallelization (Chang et al., 2022; Wang & Usher, 2005). In our previous attempt, we used an RL agent to analyze its performance against human knowledge in the semiconductor fabrication field and show the importance of exploration (Chang et al., 2022; Rawat et al., 2022).

In addition to RL, Bayesian inference is investigated and applied to laser annealing problems in this work. Recently, Bayes' statistics and inference are regarded as highly promising in machine learning fields and applications, such as Bayes network (BN) and Bayesian neural network (BNN). BN is used to compute the conditional probability of each node and is usually applied in classifiers such as Naive Bayes and BN augmented Naïve Bayes (Muralidharan & Sugumaran, 2012). On the other hand, BNN can also be used in classification (Auld et al., 2007; Thiagarajan et al., 2022) and even regression problems (Chen et al., 2019). For example, BNN is applied in the field of semiconductor manufacturing by fitting the TCAD dataset and obtaining the weights and biases of each neuron as prior, thus achieving lower regression loss with a limited amount of true data (Chen et al., 2019). In addition to BN and BNN, Bayes' theorem is also applied to RL, leading to Bayesian reinforcement learning (BRL). Many methods have been applied to realize BRL, such as Myopic value of perfect information (Myopic-VPI), Q-value sampling, and randomized prior function (RPF) (Dearden et al., 1998; Ghavamzadeh et al., 2015; Hoel et al., 2020; Osband et al., 2018; Vlassis et al., 2012). BRL is widely used in various fields (Gronewold & Vallero, 2010; Hoel et al., 2020; Kong et al., 2020; Liu et al., 2021), such as autonomous driving, gaming, and energy management, and we have not found any prior arts used in semiconductor manufacturing. Advantages of BRL are that agent can better select the actions with the assistance from prior, which possesses the domain knowledge if informative prior is used. This reduces the required training time and data amount and can lead to a more robust model (Ghavamzadeh et al., 2015; Vlassis et al., 2012). The work is included in our student thesis (Chang, 2023).

The optimization of the laser annealing process by itself is not well studied in the literature, and thus, it can be difficult to find many other papers on it (Alonso et al., 2022). Nevertheless, optimizing semiconductor processing in general is common in literature, but based on our literature review, very few papers have applied Bayesian RL to semiconductor processing (Li et al., 2023). We can only find Bayesian statistics such as Bayesian belief net, Bayesian neural network, hierarchical Bayesian model, Bayesian parameter estimation, etc., in the field of semiconductor intelligent manufacturing. The

novelty of this work lies in using TCAD prior and Bayesian formulation incorporated into RL, while Bayesian RL has been used in other fields instead of semiconductor processing.

To summarize the architecture of this paper, in “[Methodology](#)” section will describe the method, including the fabrication processes in laser annealing that is achieved in the clean room facilities. The detailed formulations constituting the entire BRL process and how the BRL framework and formulas are fitted to the laser annealing case are described. The definitions of the state, action, reward, and prior are clarified in “[Methodology](#)” section. In “[Results and discussion](#)” section provides experimental and numerical results of BRL. The RL, BRL with a fixed prior, and a BRL with a variable prior are compared. The variable prior refers to a prior that changes continuously during the RL process. The relative strengths and weaknesses of respective methods are stated, and the reason for improvement will be discussed. The contribution of this work is highlighted in the “[Results and discussion](#)” and “[Conclusion](#)” section. Essentially, the application of BRL in the field of semiconductor manufacturing is less studied in the literature, and by using the study case of laser annealing, the potential of using BRL in other semiconductor processes can be seen. More importantly, the TCAD informative prior provides valuable guidance in RL, which is well demonstrated in this work.

Methodology

Sample fabrication

First, 6-inch p-type silicon wafers of resistivity 1–10 Ω cm were cleaned using the standard (STD) clean process. In the STD clean process, wafers were cleaned using a wet bench before high-temperature deposition on wafers. First, the wafers were cleaned in a solution of $\text{NH}_4\text{OH}:\text{H}_2\text{O}_2:\text{H}_2\text{O}$ in the ratio of 1:4:20 at 75 °C for 10 min. The wafers were rinsed after the SC1 process. Then, the wafers are again cleaned using the SC2 process, which includes the solution of $\text{HCl}:\text{H}_2\text{O}_2:\text{H}_2\text{O}$ in the ratio of 1:1:6 at 75 °C for 10 min. Again, after rinsing, the wafers were cleaned with diluted hydrofluoric acid (DHF), which includes the $\text{HF}:\text{H}_2\text{O}$ in the ratio of 1:50 at room temperature for 1 min. After the DHF process, wafers were rinsed and dry spun.

After cleaning, the SVCS furnace system is used to deposit two films, 500 nm SiO_2 , and 100 nm polysilicon, to form silicon on insulator (SOI) structure to isolate the doped substrate because the target of this experiment is to get sheet resistance (R_s) after conducting laser annealing. If the substrate is not isolated, current may penetrate the substrate when measuring and make the experiment result wrong.

Table 1 Process parameters of laser annealing

	UV (Blue) HiPPO 355–5	Green HiPPO 532–11
Wavelength (nm)	355	532
Output power (W)	1–3 (50 kHz) 0.5–2 (100 kHz)	0.5–6
Repetition rate (kHz)	50 and 100	
Holder temperature (°C)	25 and 300	

The next step is implantation by Varian E500HP. In this experiment, two different ions, arsenic and phosphorus, four different ion energies, 10, 25, 40, and 55 keV, and four different doses, 5×10^{15} , 2×10^{15} , 8×10^{14} , and 5×10^{14} cm^{-2} , are used. After implantation, the wafer will be cut into 1.7×1.7 cm^2 to prevent a shift of correction factor of 4-point probe measuring. The following step is laser annealing. There are four different variables in this step: laser wavelength, laser repetition rate (Rep. rate), laser power (P), and processing temperature (T), as shown in Table 1. The laser annealing parameters selected in this work influence the annealing results. For example, the laser power is the main factor affecting the annealing procedure. Specifically, low laser power can lead to activation energy not being achieved due to heat loss, and overly high laser power leads to material damage and surface defects. The substrate temperature during laser annealing affects the lattice restoration phenomenon. Laser wavelengths affect photon absorption due to different photon energies, which in turn affect heating efficiency and profiles. Dopant types affect material chemistry and activation energy during annealing. Implant energy affects the dopant profiles in the polysilicon and the material damage. The implant dosage affects the sample doping concentration and the required annealing time and energy. The laser repetition rate effect is more ambiguous and affects the cooling and recrystallization during annealing.

NAPSON RT-80 measures every sample last to get sheet resistance. Some samples are selected to check film thickness by scanning electron microscope (SEM) with Hitachi SU-8010 and conducting secondary ion-mass spectrometry (SIMS) with CAMECA IMS 7F. Process steps are shown in Fig. 1a, b below.

TCAD simulation

In TCAD simulation, Synopsys Sentaurus 2016 is used (Sentaurus Process User Guide, 2016). From the manual of the laser used in this study, the pulse width of green and blue lasers is around 10–20 ns for both repetition rates (rep. rate), 50 and 100 kHz. The reason that TCAD cannot simulate it is

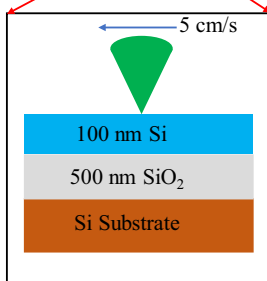
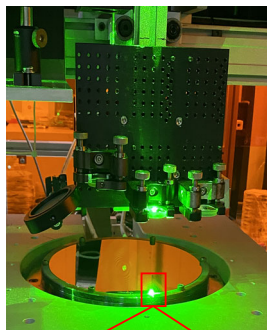
that the pulse width of the laser is too short. To solve this problem, $6t_s$, the time laser energy is mostly released in TCAD settings, is set to 1 ms and 2 ms for the rep. rate of 50 kHz and 100 kHz, respectively. The reason why $6t_s$ for 100 kHz rep. rate is twice larger than that for 50 kHz rep. rate is the total energy of two pulses at 100 kHz, the same as one pulse at 50 kHz. This means it takes twice the amount to release the same laser energy on wafers. To calculate fluence, the integral of Eq. (1) is set equal to the total energy per area of the laser used in this experiment, as shown in Eq. (1). In Eq. (1), the energy of the laser is assumed to distribute uniformly in the region of the laser spot to simplify the calculation

$$\int \frac{F}{\sqrt{2\pi}t_s} \exp\left[-\frac{(t-t_0)^2}{2t_s^2}\right] dt = \frac{P}{A} \times t_{real} \quad (1)$$

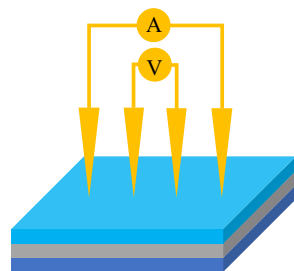
where F is the fluence, t_s is the full width at half maximum (FWHM) time interval divided by $2\sqrt{2\ln 2}$, t_0 is set to be $3t_s$, P is experimental laser power, A is laser area, and t_{real} is time illuminated by laser and defined in Eq. (2).

$$t_{real} = L/s \quad (2)$$

where L is the long axis of the laser spot, and s is the scanning rate, which is 5 cm/s.



(a)



Input parameters			Output
Ion	Rep. rate	Power	R_s
P	50	1.5	30
As	100	3	45
		...	
P	50	6	60

(b)

Fig. 1 Summary of the experimental setup for BRL/RL optimization. **a** Various steps are involved in device preparations, such as deposition of wet-oxide and polysilicon, ion-implantation and laser annealing, **b** device measurement using a 4 probe system for sheet resistance and the dataset, and **c** BRL/RL model implemented in this work

Regular reinforcement learning (regular RL)

In this study, Python 3.8.13 (Rossum & Drake, 2009), Tensorflow 2.7.0 (Abadi et al., 2016), Numpy 1.21.2 (Harris et al., 2020), Scipy 1.8.1 (Jones et al., 2001), scikit-learn 1.2.0 (Pedregosa et al., 2011), and Pandas 1.4.2 (McKinney, 2011) is used for constructing the models.

For data preprocessing, because some variations in laser power and temperature can cause an error in Python training, they are fixed to the ideal same values and transformed to discrete levels. For example, there are 4 different ion energies, 10, 25, 40, and 55 keV, and each corresponds to a level of 1, 2, 3, and 4, respectively. A deep Q network (DQN) is selected for the RL agent in this study. In every trial, it will compute the Q value of each action, as shown in Eq. (3) (Mnih et al., 2015), and select an action based on the epsilon-greedy algorithm expressed in Eq. (4).

$$Q(s, a) = \begin{cases} r_{real} + \gamma \max_{a' \in A} Q_{original}(s', a'), & \text{sampled points} \\ \text{Estimate from Q-Table neural network,} & \text{unsampled points} \end{cases} \quad (3)$$

$$\pi(s, a) = \begin{cases} \arg \max Q(s, a), & \text{with probability } 1 - \varepsilon \\ \text{random choice,} & \text{with probability } \varepsilon \end{cases} \quad (4)$$

where r_{real} is the reward received from the current state (s) to the next state (s'), $\gamma = 0.2$ is the discount factor, A is the action space, a stands for the current action, and a' stands for the next probable actions, and ε is the exploration rate defined in Eq. (5).

$$\varepsilon = 0.5 \times 0.9(\text{epoch} - 1) \quad (5)$$

For the state space, the state is defined as $s = [\text{Ion}, \text{Dose}, \text{Ion energy}, \text{Wavelength}, \text{rep. rate}, P, T]$, the experimental parameters used in laser annealing. According to this state and the action applied, the environment will return the reward. This study has two different action types for the action space. First, action (a) is defined as the direct transition to any state in the RL. This means the action space is the same as the state space. In addition, this also implies that regardless of the current state, the same action leads to the same next state, and the transition can be directed to any defined state in an RL problem. The other type of action is to raise, maintain, or drop the level of each parameter in a state, i.e., a $+1/0/-1$ action. In this way, the result of action depends not only on the action but also on the current state. As for the reward, each pair of a state and an action has its reward, and it is defined as $-R_s$, which is the negative value of sheet resistance at the next state. The agent will maximize the reward by choosing a lower sheet resistance value. In this study, the Q-Table neural network model has 2 or 5 hidden layers with 100 neurons in each hidden layer, and the batch

size is 2, 5, and 10 when updating the model based on Eq. (6) (Mnih et al., 2015). Training will be finished in 5 RL epochs with 10 timesteps in each epoch. The initial exploration rate is set to 0.5 in the first epoch. It will be 0.9 times lower than the previous epoch because the agent gradually receives more data from the environment. Therefore, the Q-Table training and selection should be based on the agent’s experience. The update of Q-Table is based on (Mnih et al., 2015)

$$L_t = E_{(s,a,r,s') \sim U(D)} \left[\left(r_{real} + \gamma \max_{a'} Q(s', a') - Q(s, a) \right)^2 \right] \tag{6}$$

where L_t is the loss function of the Q-Table neural network, (s,a,r,s') is the agent’s experience, and D is the restored dataset of experiences at each time step.

Bayesian reinforcement learning (BRL)

In BRL, the definitions of the state space, the action space, the reward, and the batch size are the same as those of the regular RL. One difference is that BRL will initialize two different Q-Table neural network models. Then, it selects one of them when determining an action to apply at every 5 RL timesteps, as shown in Eq. (7). Q values will be updated by the selected model, as shown in Eq. (8).

$$k = \text{Uniform}\{1, 2\}, \text{ every 5 timesteps} \tag{7}$$

$$Q_k(s, a) = \begin{cases} r_{real} + \gamma \max_{a' \in A} Q_{k,original}(s', a'), & \text{sampled points from } k=1 \text{ and } 2 \\ \text{Estimate from Q - Table neural networks,} & \text{unsampled points} \end{cases} \tag{8}$$

In addition, there is a prior from TCAD, which will also be incorporated to guide the agent to select better action, as shown in Eq. (9).

$$p(s, a) = r_{TCAD} \tag{9}$$

where r_{TCAD} is the reward value from TCAD Q-table.

Combining the prior in Eq. (9) and the experiential sampling, the agent selects an action based on Eq. (10) (Hoel et al., 2020)

$$\pi(s, a) = \begin{cases} \arg \max(Q_k(s, a) + 2p(s, a)), & \text{with probability } 1 - \varepsilon \\ \text{random choice,} & \text{with probability } \varepsilon \end{cases} \tag{10}$$

where ε is the exploration rate defined in Eq. (5).

There are two different types of priors, a fixed or variable prior. A fixed prior will keep its value until the end of BRL process. Nevertheless, the variable prior will keep changing its values during the BRL process. One update on the prior

is that we partially correct the TCAD prior by the experimentally sampled true values in our practice. Thus, at the end of every RL step, true data from the experiment will fix the TCAD values, and the data points to be modified in the TCAD Q-Table are selected based on their adjacency to the sampled data point in the input space. Since our input space in the optimization objective function is discrete and leveled, the difference of one level, when considering all seven semiconductor process parameter input variables, is regarded as being within the distance eligible for TCAD prior modification. Modification is expressed by Eq. (11). In addition to the TCAD prior modification by the true experimental values, the weight of prior will keep decreasing from timestep $i = 11$ to suppress the effect of prior inaccuracy, as shown in Eq. (13), and the action is selected based on Eq. (14)

$$P(s, a) = \begin{cases} r_{real}, & \text{sampled points from } k=1 \text{ and } 2 \\ r_{TCAD} \times \text{multiplicand}, & \text{around sampled points from } k=1 \text{ and } 2 \\ r_{TCAD}, & \text{others} \end{cases} \tag{11}$$

where multiplicand is defined in Eq. (12)

$$\text{multiplicand} = \text{abs}(\text{reward}_i / \text{TCAD}_i) \tag{12}$$

where TCAD_i and reward_i are -Rs value of experiment parameters selected at timestep i from TCAD and environment, respectively.

$$P'(s, a) = \begin{cases} P(s, a), & i \leq 10 \\ P(s, a) / (i - 10)^2, & i \geq 11 \end{cases} \tag{13}$$

$$\pi(s, a) = \begin{cases} \arg \max(Q_k(s, a) + 2P'(s, a)), & \text{with probability } 1 - \varepsilon \\ \text{random choice,} & \text{with probability } \varepsilon \end{cases} \tag{14}$$

where i is the timestep, ε is the exploration rate defined in Eq. (5), and Q_k is defined in Eq. (7). The comparison of algorithms between RL and BRL is shown in Fig. 2.

Results and discussion

Laser annealing analysis

Some samples with the same ion (Phosphorous), dose ($5 \times 10^{15} \text{ cm}^{-2}$), ion energy (40 keV), and wavelength (532 nm), but different power, rep. rate, and chuck temperature are shown in Table 2.

In Table 2, it is easy to observe that sheet resistance increases with lower power because the depth of the melted zone decreases, and lots of defects remain in the silicon. The phenomenon can be solved with a high holder temperature to compensate for the lack of laser power. From the top two

Fig. 2 BRL/RL model implemented in this work

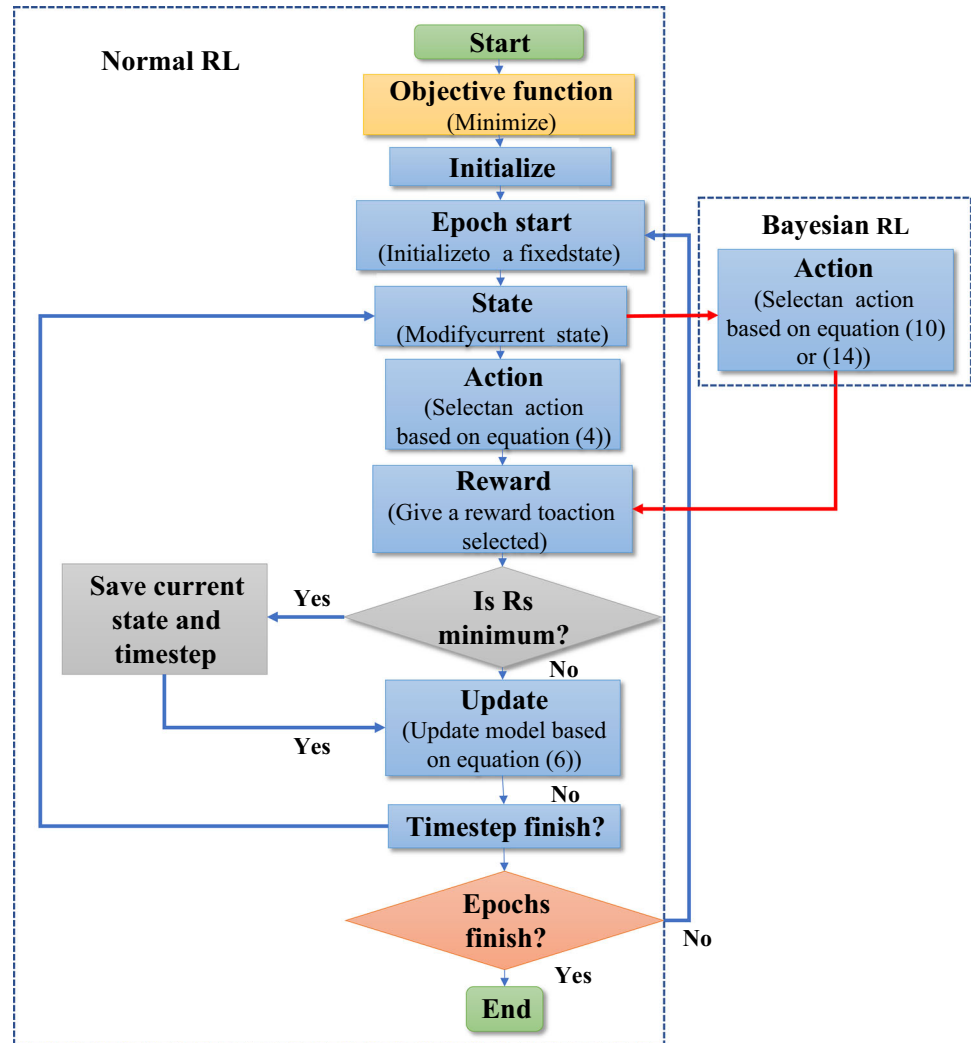


Table 2 Result of laser annealing

Power (W)	Rep. Rate (kHz)	Holder temperature (°C)	Sheet resistance (Ω/sq)
2.996	50	25	118.9
2.996	50	302	76.3
2.995	100	25	417.87
2.999	100	299	242.37
6.001	50	25	50.58
5.999	50	304	47.54
5.997	100	25	100.68
5.996	100	299	63.93

columns of Table 2, the difference between these results is about 40 Ω/sq with the same power and rep. rate, but there is almost no variation in a sample with 6 W and 50 kHz. The reason is that laser power is high enough to make silicon

recrystallize well without holder temperature. Another point to pay attention to is that the holder temperature cannot be too high. A higher holder temperature causes the dopant to diffuse more easily in silicon, which loses the advantage of laser annealing to confine the dopant in the region illuminated by the laser.

The cross-sectional SEM images of the fabricated samples are shown in Fig. 3, where the thicknesses of deposited layers are labeled. SEM images confirm the deposited thicknesses of 100 nm polysilicon and 500 nm wet-oxide over the p-type silicon wafer. Figure 3a, b have arsenic as a dopant with a dose of $5 \times 10^{15} \text{ cm}^{-2}$ and energy of 25 keV and 55 keV, respectively. Figure 3c, d have phosphorous as a dopant with a dose of $5 \times 10^{15} \text{ cm}^{-2}$ and energy of 25 keV and 55 keV, respectively.

Figure 4 shows the images SIMS. SIMS is a useful tool for analyzing the surface composition and studying the depth profiling of the dopants in the multilayer sample. SIMS provides a detailed analysis of the composition formed on the

Fig. 3 Cross-sectional SEM images of the samples. Arsenic as the dopant with dose $5 \times 10^{15} \text{ cm}^{-2}$ and energy of **a** 25 keV and **b** 55 keV, respectively, and phosphorous as the dopant with dose $5 \times 10^{15} \text{ cm}^{-2}$ and energy of **c** 25 keV and **d** 55 keV, respectively

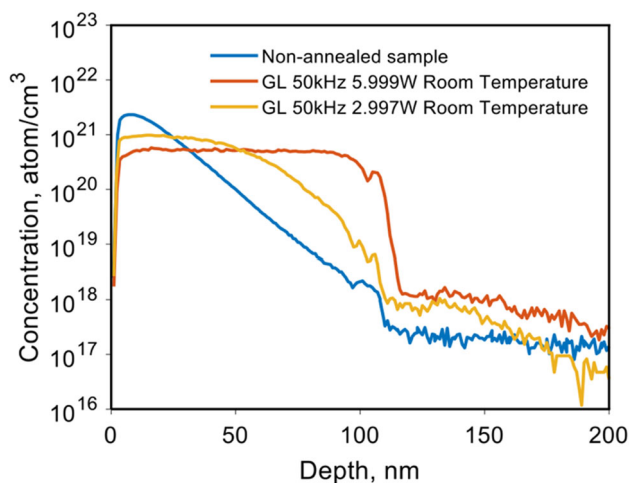
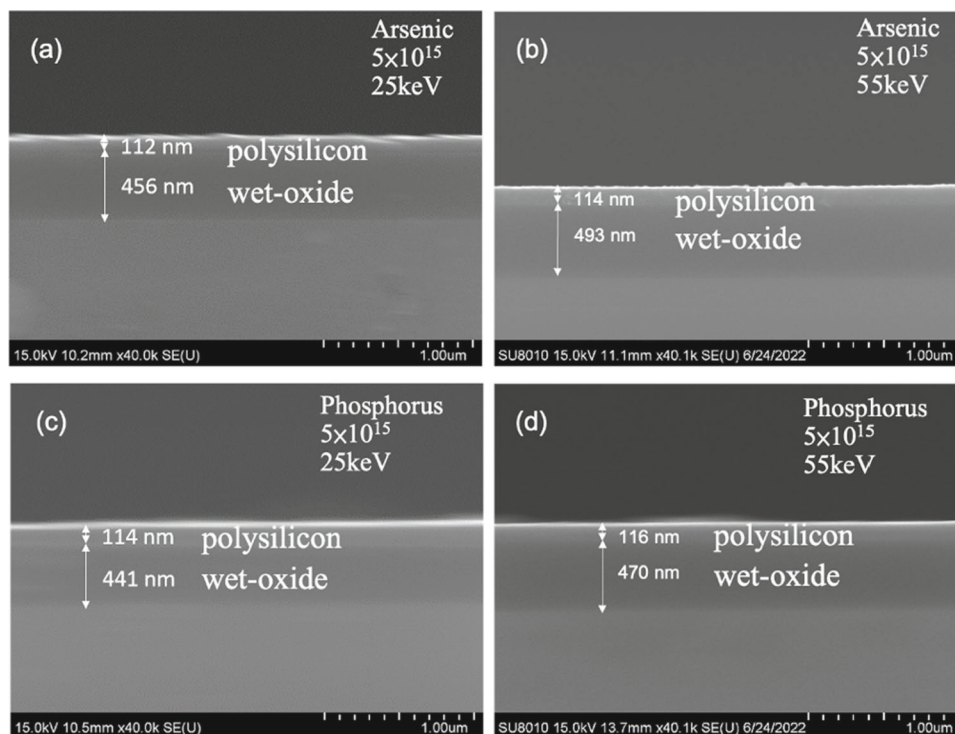


Fig. 4 SIMS profile confirms the presence of arsenic dopant in the samples. The dopant profiles for the sample annealed in a green light laser at a 50kHz repetition rate and power of 5.999 and 2.997 W

sample. For depth profiling, SIMS is very efficient in determining even the very low concentration of sub-ppm or ppb of dopants. In Fig. 4, the SIMS profile confirms the presence of As dopant and its diffusion along the depth of the annealed and non-annealed samples. Samples in Fig. 4 were annealed at room temperature using a green light laser with a repetition rate of 50kHz and two different powers of 5.999 and 2.997 W, respectively. SIMS profile also confirms the diffusion of As dopant with increased laser power used in the annealing.

RL and BRL analysis

From Table 3 and Table 4, the average minimum R_s of RL is over 50 using regular RL without a prior. Besides, the variance of the result is also significant, i.e., 20.73 in Table 3 and 44.25 in Table 4, and this is a serious issue since this indicates the result strongly depends on hyperparameter selection and can arrive at much degraded results with improper hyperparameters. Due to a lack of prior knowledge of each action, regular RL can only retrieve information by experimentally sampling the environment, which leads to selected states whose rewards are very small, especially when the experimental sampling is insufficient. To circumvent the problem of regular RL without a prior, BRL with a fixed prior is investigated. From the results of the BRL with a fixed prior in Tables 3 and Table 4, the average minimum based on two different action types are 47.29 and 44.32 Ω/sq , with the average count before the minimum of only 11.50 and 10.50 counts. The average is over different settings of hyperparameters. The reduced count before the minimum is an important benefit in semiconductor process development, where the experimental cost is very high. Additionally, from Tables 3 and Table 4, it can be observed that the variance of the minimum R_s values from different hyperparameters is reduced in reference to regular RL. The variance of BRL with a fixed prior in this case is 2.29 in Table 3 and 0 in Table 4, in reference to 20.73 in Table 3 and 44.25 in Table 4 in the case of regular RL w/o a prior. Therefore, the burden and risk of hyperparameter selection are alleviated with the assistance

Table 3 Results in the first 50 timesteps for various hyperparameters and algorithms

	Model		Prior		Result				Average		
	Network	Batch size	Fixed	Variable	Minimum R_s (Ω)	Min. R_s variance (Ω^2)	Count before minimum	Total experiment count	Minimum R_s (Ω)	Count before minimum	Total experiment count
RL	2	2	0	0	57.97	20.73	14	21	53.70	15.17	28.67
	2	5	0	0	47.54		8	36			
	2	10	0	0	59.71		15	25			
	5	2	0	0	54.25		13	34			
	5	5	0	0	51.67		26	34			
	5	10	0	0	51.07		15	22			
BRL w/fixed prior	2	2	1	0	48.4	2.29	1	23	47.29	11.50	23.33
	2	5	1	0	44.32		23	23			
	2	10	1	0	47.54		17	24			
	5	2	1	0	48.4		5	21			
	5	5	1	0	47.54		8	26			
	5	10	1	0	47.54		15	23			
BRL w/variable prior	2	2	0	1	45.19	0	9	39	45.19	14.00	36.50
	2	5	0	1	45.19		9	39			
	2	10	0	1	45.19		9	39			
	5	2	0	1	45.19		11	28			
	5	5	0	1	45.19		35	35			
	5	10	0	1	45.19		11	39			

Bold value indicates the best performance metrics
Action can point to any state directly

of a fixed prior. From the advantages mentioned above, a fixed-prior BRL proves that it can guide the agent well to the RL state parameters, i.e., the semiconductor process input parameters, that possess a higher potential to attain lower sheet resistance. Subsequently, the more effective locating of the RL state reduces the optimization path and leads to lower R_s .

It is observed that the fixed-prior BRL achieves better results than the regular RL does, but there is still a drawback. From Figs. 5b, e, and 6b, e, it is clearly shown that whatever the hyperparameters are set, it is easy to be stuck at the same state because it is possible that the value of prior is higher than the predicted value from experimental Q-Table model. This makes the agent choose an action based on the prior instead of the experimental Q-Table model, except at the timestep that exploration is activated. Intuitively, when more experimental steps are conducted, the experimental Q-Table should be more and more reliable. In this scenario, it can be desired to let the effect of the prior decay even if the prior is informative. As a result, the prior decay here is established based on the RL timestep, where the effect of the prior is gradually diminished. In addition, the information

stored in the prior will also be corrected according to the true experimental data to improve the TCAD prior if the state is sampled.

From Tables 3 and Table 4, The minimum R_s values of BRL with a variable prior for two different action types are 45.19 Ω /sq and 44.32 Ω /sq, respectively, and the counts before the minimum are 14.00 and 4.50, respectively. Compared to fixed-prior BRL, the variable-prior method achieves either smaller R_s at the expense of a higher step count before the minimum or a reduced step count before the minimum with the same resultant R_s . This reflects the improvement from using variable priors to take into account the gradually improved experimentally sampled Q-Table. Besides, another important advantage is that its variance is smaller than the other two methods across different hyperparameters, where 0 is achieved in reference to 20.73 in Table 3 and 44.25 in Table 4 in RL w/o a prior and 2.29 in Table 3 and 0 in Table 4 in RL with a fixed prior. In conclusion, using the BRL with a variable prior can achieve either lower R_s at the expense of a higher step count or similar R_s with a reduced step count. Besides, lower variance in the optimal R_s value is observed in the BRL with a variable prior. One

Table 4 Results in the first 50 timesteps for various hyperparameters and algorithms. Action is +1/0/-1 applied to the input parameters in the state

	Model		Prior		Result				Average		
	Network	Batch Size	Fixed	Variable	Minimum R_s (Ω)	Min. R_s variance (Ω^2)	Count before minimum	Total experiment count	Minimum R_s (Ω)	Count before minimum	Total experiment count
RL	2	2	0	0	47.54	44.25	26	37	51.84	18.17	42.83
	2	5	0	0	50.27		9	46			
	2	10	0	0	50.27		9	45			
	5	2	0	0	65.14		3	45			
	5	5	0	0	47.54		26	37			
	5	10	0	0	50.27		36	47			
BRL w/ fixed prior	2	2	1	0	44.32	0	15	27	44.32	10.50	28.00
	2	5	1	0	44.32		15	29			
	2	10	1	0	44.32		24	31			
	5	2	1	0	44.32		3	25			
	5	5	1	0	44.32		3	27			
	5	10	1	0	44.32		3	29			
BRL w/ variable prior	2	2	0	1	44.32	0	6	45	44.32	4.50	40.00
	2	5	0	1	44.32		6	45			
	2	10	0	1	44.32		6	44			
	5	2	0	1	44.32		3	38			
	5	5	0	1	44.32		3	33			
	5	10	0	1	44.32		3	35			

Bold value indicates the best performance metrics

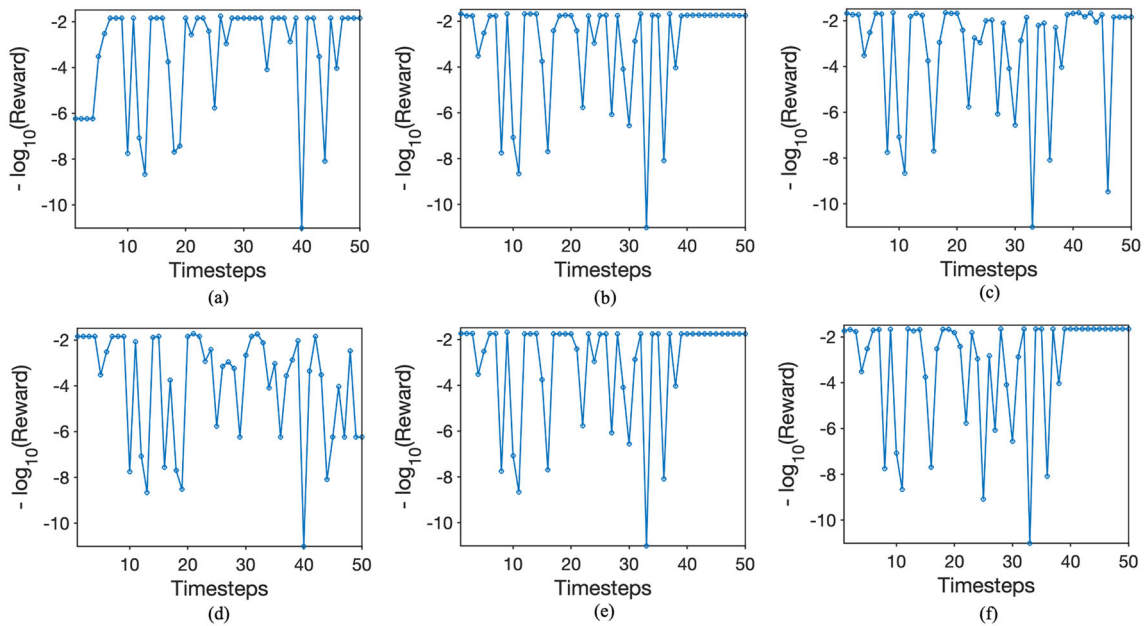


Fig. 5 Path of RL with network = 2 and batch size = 2 **a** RL w/o prior **b** BRL w/ fixed prior **c** BRL w/ variable prior, and network = 5 and batch size = 2 **d** w/o prior **e** w/ fixed prior **f** w/ variable prior whose action can point to any state directly

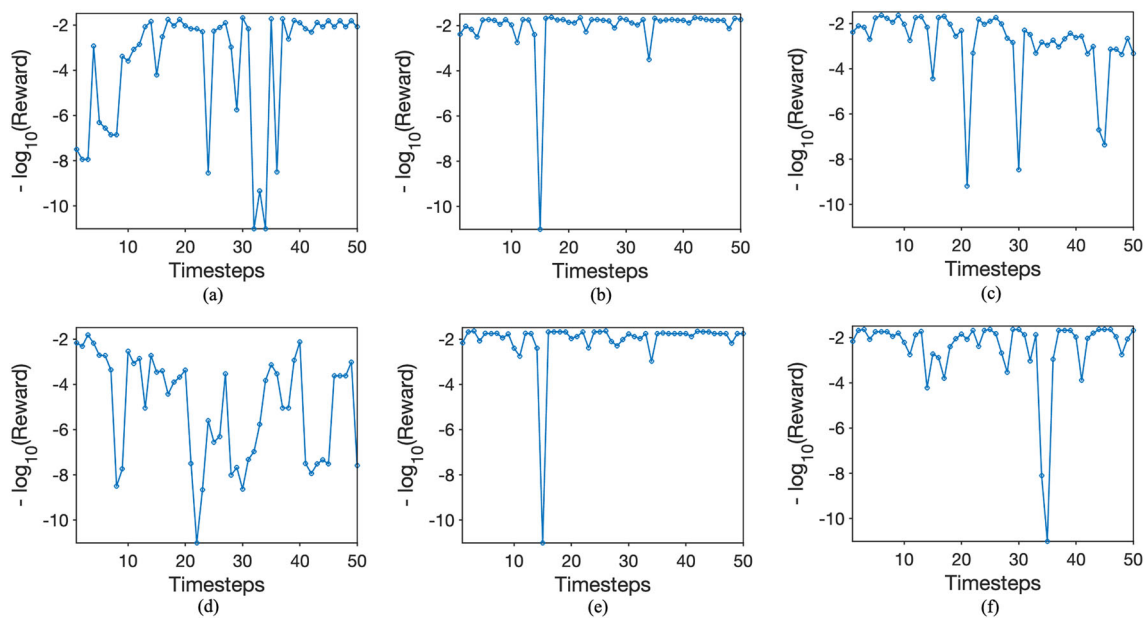


Fig. 6 Path of RL with network = 2 and batch size = 2 **a** RL w/o prior **b** BRL w/ fixed prior **c** BRL w/ variable prior, and network = 5 and batch size = 2 **d** w/o prior **e** w/ fixed prior **f** w/ variable prior whose action is + 1/0/− 1 of level to each input parameter in the RL/BRL state

Table 5 The results of conventional optimization methods

Method	Minimum R_S	Count before minimum	Total experiment count
Taguchi array level 2 (Leung & Wang, 2001)	190.94	–	8
Taguchi array level 4 (Leung & Wang, 2001)	55.27	–	64
Differential evolution (Jones et al., 2001)	55.98	31	34
Basin hopping (Jones et al., 2001)	57.98	35	46

potential risk of using a variable-prior BRL is that the agent can start to sample the areas with degraded sheet resistance when the prior almost disappears, as shown in Fig. 6c. As a result, balancing the decay of the prior and the guidance capability of the informative prior on the agent in the BRL is essential. While we use a decay rate specified in Eq. (13) in this work, meta-learning can also be used to control the decay rate adaptively to improve this aspect further. Table 5 further shows the results of some conventional optimization

methods applied to the same laser annealing parameter tuning. The Taguchi method achieves either a higher R_S or a larger total experiment count compared to the BRL method in Tables 3 and Table 4. Specifically, 2-level discretization gives $R_S = 190.94$, which is too high for practical use. 4-level discretization gives $R_S = 55.27$ using 64 total experiment counts, though the R_S value is higher than the values in BRL using 50 total experimental counts. Scipy differential evolution gives a reasonably low $R_S = 55.98$, but the count before the minimum is large. On the other hand, Scipy basin hopping gives $R_S = 57.98$, which is still higher than the R_S achieved by BRL and has a large count before the minimum.

Discussions

Comparison of two different actions

The results of RL are shown in Tables 3 and Table 4. It is clearly shown that the case of action pointing to any state directly is worse than the case of + 1/0/− 1 actions. The reason is that there are 1024 different actions, and the agent does not know their true rewards before sampling and can only learn them one by one. In addition, once an agent finds an action with a higher reward, it is likely to use the same action in the next trial and get the same reward. This can mislead the agent by making it think this is the best action and thus trapped. Although the case of + 1/0/− 1 actions also selects an action with a large reward from previous trials, the next state and reward are dependent on the current state. This means it can get a better or worse reward when choosing the

same action. As a result, this prevents the agent from being trapped in the same state and facilitates learning each action's meaning. The disadvantage of this method is that it samples more data points.

Bayesian formulation

With normal Bayes' theorem, which is a multiplication of likelihood and prior, the agent is excessively influenced by the likelihood that is non-informative in the beginning and wastes lots of trials to modify those points. The advantage of the addition of prior and likelihood is that it can adjust prior decay. This benefits the agent to follow the instructions of prior appropriately to sample the space with lower sheet resistance to increase sampling efficiency. Compared to VPI, which embeds prior information in the likelihood model, the methods applied in this study can easily change the decay of prior because our method does not have to fit the likelihood model with prior embedded inside at each step. Besides, VPI is computationally expensive due to the required VPI integration at each potential action. Therefore, in our case, at least 128 different integrations at each step are needed, which will take a lot of time to finish one trial.

Fixed and variable prior

In prior work (Hoel et al., 2020), 10 different non-informative prior functions are used in each member of the ensemble RL model, which can prevent agents from being trapped in local minima with priors' uncertainty. In fact, there is no need to modify the decay of the prior because they are non-informative and are trained and varied together with the likelihood in most cases. In semiconductor manufacturing, informative prior methods are always preferred over non-informative prior methods due to the extremely high cost in fabrication. In this study, there may be wrong data due to the inaccuracy of TCAD simulation, which leads the agent to be stuck or easily waste trails. Due to the above reasons, balancing the decay of prior and information stored in prior is the most important issue. In our algorithm, the prior is only modified according to collected data but not diminished at the first epoch to specify where underestimation or overestimation is. This makes the agent modify its searching direction and avoid wasting trails. The advantage of initiating prior decay from the second epoch is that the agent will not be trapped in future trials and can incorporate more environmental knowledge and sample more efficiently. In addition, it can also prevent wrong data from being accumulated, according to Eq. (12).

Comparison with other machine learning methods

Compared to supervised learning (SL) and unsupervised learning (USL), RL trains the agent based on the data collected by the interactions with the environment instead of a fixed dataset. In SL and USL, the prediction accuracy is highly dependent on whether the data collection properly spans the entire sample space. Suppose the SL is going to be used to optimize the process condition. In that case, the data collection should be extended over all possible choices in input process parameters, leading to inefficient optimization. Besides, active learning (AL) is another method to sample other data from unlabeled pools like RL to search the environment. The major difference between AL and RL is that RL can predict future rewards based on the gamma factor, which means RL can make a decision based on the current and future rewards while AL cannot. In addition, RL can be more flexible by adjusting its reward function according to specific tasks instead of labeling data directly. In this regard, RL can select a better path and be a more potent way to optimize parameters.

Conclusion

In this study, we investigate the Bayesian approach in RL to accelerate the optimization of semiconductor process parameter tuning. The RL without TCAD prior knowledge shows that it requires more trials to achieve lower R_s values and has a large variance with different settings of hyperparameters. In contrast, BRL with a fixed prior can guide the agent to the search space where the reward is larger. This significantly prevents the agent from wasting trials. The fixed TCAD prior effectively guides the learning process in the laser annealing problem. On the other hand, the variable TCAD prior, where the TCAD prior effect is gradually decayed and corrected to reflect the increased experimental data collection, is improved so that the agent can achieve lower R_s at the expense of more step count or a similar R_s with a reduced step count. Specifically, using the action with direct state transition, the achieved R_s and the step count before minimum R_s for RL, BRL with a fixed prior, and BRL with a variable prior are 53.70 Ω , 15.17, 47.29 Ω , 11.50, 45.19 Ω , and 14.00, respectively. On the other hand, using the +1-0/-1 action, the achieved R_s and the step count before minimum R_s for RL, BRL with a fixed prior, and BRL with a variable prior are 51.84 Ω , 18.17, 44.32 Ω , 10.50, 44.32 Ω , 4.50, respectively. Based on this study, it is seen that BRL with informative prior can be an essential infrastructure for future intelligent semiconductor manufacturing.

Author contributions Chung-Yuan Chang conducts the experiments, coding, and manuscript writing. Yen-Wei Feng and Tejender Singh

Rawat assist in the coding and manuscript preparation. Shih-Wei Chen and Albert Shihchun Lin supervise the research.

Funding Open Access funding enabled and organized by National Yang Ming Chiao Tung University. Ministry of Science and Technology (MOST), Taiwan (MOST 111-2221-E-A49-187).

Data availability The data and code is available at <https://github.com/albertlin11/BRL>.

Declarations

Conflict of interest The authors declare no competing financial interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., & Devin, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Preprint at <http://arxiv.org/abs/1603.04467>. <https://doi.org/10.48550/arXiv.1603.04467>
- Alonso, A. A., Alba, P. A., Rahier, E., Kerdilès, S., Gauthier, N., Bernier, N., & Claverie, A. (2022). Optimization of solid-phase epitaxial regrowth performed by UV nanosecond laser annealing. *MRS Advances*. <https://doi.org/10.1557/s43580-022-00443-8>
- Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press. https://doi.org/10.1007/978-1-62703-748-8_7
- Altenmüller, T., Stüker, T., Waschneck, B., Kuhnle, A., & Lanza, G. (2020). Reinforcement learning for an intelligent and autonomous production control of complex job-shops under time constraints. *Production Engineering*. <https://doi.org/10.1007/s11740-020-00967-8>
- Auld, T., Moore, A. W., & Gull, S. F. (2007). Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, 18, 223–239.
- Çaydaş, U., & Ekici, S. (2012). Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-010-0415-2>
- Chang, C.-Y. (2023). Optimization of laser annealing processing parameters based on Bayesian reinforcement learning. *Institute of Electronics, National Yang Ming Chiao Tung university*. <https://etd.lib.nctu.edu.tw/cgi-bin/gs32/tugsweb.cgi/ccd=1dnAiF/record?r1=8&h1=1>
- Chang, C.-Y., Hsu, C.C., Rawat, T., Chen, S.-W., & Lin, A. (2022). Human machine competition in intelligent laser manufacturing in semiconductor processes. In *SPIE Optical Engineering + Applications SPIE*. <https://doi.org/10.1117/12.2632661>
- Chen, C. H., Parashar, P., Akbar, C., Fu, S. M., Syu, M. Y., & Lin, A. (2019). Physics-prior Bayesian neural networks in semiconductor processing. *IEEE Access*, 7, 130168–130179.
- Dearden, R., Friedman, N., & Russell, S. (1998). Bayesian Q-learning. In *Innovative applications of artificial intelligence conferences (IAAI)* (pp. 761–768).
- Doltsinis, S., Ferreira, P., & Lohse, N. (2012). Reinforcement learning for production ramp-up: A Q-batch learning approach. In *2012 11th international conference on machine learning and applications* (pp. 610–615).
- Gardner, R., & Bicker, J. (2000). Using machine learning to solve tough manufacturing problems. *International Journal of Industrial Engineering: Theory Applications and Practice*, 7, 359–364.
- Ghavamzadeh, M., Mannor, S., Pineau, J., & Tamar, A. (2015). Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*. <https://doi.org/10.1561/22000000049>
- Gluschenkov, O., & Jagannathan, H. (2018). Laser annealing in CMOS manufacturing. *ECS Transactions*. <https://doi.org/10.1149/08506.0011ecst>
- Gronewold, A., & Vallero, D. (2010). Applications of Bayes' theorem for predicting environmental damage. <https://doi.org/10.1036/1097-8542.YB100249>
- Guevara, J. L., Patel, R. G., & Trivedi, J. J. (2018). Optimization of steam injection for heavy oil reservoirs using reinforcement learning. *SPE International Heavy Oil Conference and Exhibition*. <https://doi.org/10.2118/193769-MS>
- Günther, J., Pilarski, P. M., Helfrich, G., Shen, H., & Diepold, K. (2016). Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning. *Mechatronics*. <https://doi.org/10.1016/j.mechatronics.2015.09.004>
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*. <https://doi.org/10.1038/s41586-020-2649-2>
- He, J., Tang, M., Hu, C., Tanaka, S., Wang, K., Wen, X.-H., & Nasir, Y. (2022). Deep reinforcement learning for generalizable field development optimization. *SPE Journal*. <https://doi.org/10.2118/203951-PA>
- Hoel, C.J., Wolff, K., & Laine, L. (2020). Tactical decision-making in autonomous driving by reinforcement learning with uncertainty estimation. In *2020 IEEE intelligent vehicles symposium (IV)* (pp. 1563–1569).
- Hourfar, F., Bidgoly, H. J., Moshiri, B., Salahshoor, K., & Elkamel, A. (2019). A reinforcement learning approach for waterflooding optimization in petroleum reservoirs. *Engineering Applications of Artificial Intelligence*. <https://doi.org/10.1016/j.engappai.2018.09.019>
- Jacobs, T., Alesiani, F., & Ermis, G. (2021). Reinforcement learning for route optimization with robustness guarantees. In *International joint conferences on artificial intelligence organization (IJCAI)* (pp. 2592–2598).
- Jones, E., Oliphant, T., & Peterson, P. (2001). SciPy: Open source scientific tools for Python.
- Khader, N., & Yoon, S. W. (2021). Adaptive optimal control of stencil printing process using reinforcement learning. *Robotics and Computer-Integrated Manufacturing*, 71, 102132.
- Khakifirooz, M., Fathi, M., & Chien, C. F. (2021). Partially observable Markov decision process for monitoring multilayer wafer fabrication. *IEEE Transactions on Automation Science and Engineering*, 18, 1742–1753.
- Khazadeh, M., Rao, P., Jafari-Marandi, R., Smith, B. K., Tschopp, M. A., & Bian, L. (2017). Quantifying geometric accuracy with unsupervised machine learning: Using self-organizing map on fused

- filament fabrication additive manufacturing parts. *Journal of Manufacturing Science and Engineering*. <https://doi.org/10.1115/1.4038598>
- Kong, H., Yan, J., Wang, H., & Fan, L. (2020). Energy management strategy for electric vehicles based on deep Q-learning using Bayesian optimization. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-019-04556-4>
- Kormushev, P., Calinon, S., & Caldwell, D.G. (2010). Robot motor skill coordination with EM-based reinforcement learning. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3232–3237).
- Kotsiopoulos, T., Sarigiannidis, P., Ioannidis, D., & Tzovaras, D. (2021). Machine learning and deep learning in smart manufacturing: The smart grid paradigm. *Computer Science Review*. <https://doi.org/10.1016/j.cosrev.2020.100341>
- Kusiak, A. (1990). Intelligent manufacturing. In *System*. Prentice-Hall.
- Lee, Y. H., & Lee, S. (2022). Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. *Expert Systems with Applications*. <https://doi.org/10.1016/j.asoc.2012.03.021>
- Leung, Y.-W., & Wang, Y. (2001). An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, 5, 41–53.
- Li, T.-S., Huang, C.-L. (2009). Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing. *Expert Systems with Applications*. 374–385. <https://doi.org/10.1016/j.eswa.2007.09.023>
- Li, Y., Du, J., & Jiang, W. (2021). Reinforcement learning for process control with application in semiconductor manufacturing. Preprint at <http://arxiv.org/abs/2110.11572>. <https://doi.org/10.1080/24725854.2023.2219290>
- Li, Y., Du, J., Jiang, W. (2023). MFRL-BI: Design of a model-free reinforcement learning process control scheme by using Bayesian inference. <http://arxiv.org/abs/2309.09205>. <https://doi.org/10.48550/arXiv.2309.09205>
- Li, Z., Zhang, Z., Shi, J., & Wu, D. (2019). Prediction of surface roughness in extrusion-based additive manufacturing with machine learning. *Robotics and Computer-Integrated Manufacturing*. <https://doi.org/10.1016/j.rcim.2019.01.004>
- Liu, J., Wang, X., Shen, S., Yue, G., Yu, S., & Li, M. (2021). A Bayesian Q-learning game for dependable task offloading against DDoS attacks in sensor edge cloud. *IEEE Internet of Things Journal*, 8, 7546–7561.
- Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing*. <https://doi.org/10.1016/j.asoc.2020.106208>
- McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14, 1–9.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Belle-mare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., & Ostrovski, G. (2015). Human-level control through deep reinforcement learning. *Nature*. <https://doi.org/10.1038/nature14236>
- Monostori, L., Csáji, B. C., & Kádár, B. (2004). Adaptation and learning in distributed production control. *CIRP Annals*. [https://doi.org/10.1016/S0007-8506\(07\)60714-8](https://doi.org/10.1016/S0007-8506(07)60714-8)
- Monostori, L., Hornyák, J., Egresits, C., & Viharos, Z. J. (1998). *Soft computing and hybrid AI approaches to intelligent manufacturing*. Springer. https://doi.org/10.1007/3-540-64574-8_463
- Muralidharan, V., & Sugumaran, V. (2012). A comparative study of Naïve Bayes classifier and Bayes net classifier for fault diagnosis of monoblock centrifugal pump using wavelet analysis. *Applied Soft Computing*. <https://doi.org/10.1016/j.asoc.2012.03.021>
- Osband, I., Aslanides, J., Cassirer, A. (2018). Randomized prior functions for deep reinforcement learning. *Advances in Neural Information Processing Systems*. <https://doi.org/10.48550/arXiv.1806.03335>
- Park, I. B., Huh, J., Kim, J., & Park, J. (2020). A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. *IEEE Transactions on Automation Science and Engineering*, 17, 1420–1431.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., & Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pey, K. L., & Lee, P. S. (2018). Chapter 12—Pulsed laser annealing technology for nano-scale fabrication of silicon-based devices in semiconductors. In *Advances in laser materials processing* (2nd Ed., pp. 299–337). Woodhead Publishing. <https://doi.org/10.1016/B978-0-08-101252-9.00012-1>
- Pham, D. T., & Afify, A. A. (2005). Machine-learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. <https://doi.org/10.1243/095440505X32274>
- Pradeep, D. J., & Noel, M. M. (2018). A finite horizon Markov decision process based reinforcement learning control of a rapid thermal processing system. *Journal of Process Control*. <https://doi.org/10.1016/j.jprocont.2018.06.002>
- Rajendran, B., Shenoy, R. S., Witte, D. J., Chokshi, N. S., DeLeon, R. L., Tompa, G. S., & Pease, R. F. W. (2007). Low thermal budget processing for sequential 3-D IC fabrication. *IEEE Transactions on Electron Devices*. <https://doi.org/10.1007/s10845-024-02323-4>
- Rawat, T., Chung, C.-Y., Chen, S.-W., & Lin, A. (2022). Reinforcement learning based intelligent semiconductor manufacturing applied to laser annealing. *engrxiv*.
- Rawat, T. S., Chang, C. Y., Feng, Y.-W., Chen, S., Shen, C.-H., Shieh, J.-M., & Lin, A. S. (2023). Meta-learned and TCAD-assisted sampling in semiconductor laser annealing. *ACS Omega*. <https://doi.org/10.1021/acsomega.2c06000>
- Robinson, A. L. (1978). Laser annealing: Processing semiconductors without a furnace. *Science*, 201, 333–335.
- Ruvolo, P., Fasel, I., & Movellan, J. (2008). Optimization on a budget: A reinforcement learning approach. *Advances in Neural Information Processing Systems*.
- Salahshoor, K., Kordestani, M., & Khoshro, M. S. (2010). Fault detection and diagnosis of an industrial steam turbine using fusion of SVM (support vector machine) and ANFIS (adaptive neuro-fuzzy inference system) classifiers. *Energy*. <https://doi.org/10.1016/j.asoc.2012.03.021>
- Sentaurus process user guide. (2016) Synopsys, Inc.
- Shi, D., Fan, W., Xiao, Y., Lin, T., & Xing, C. (2020). Intelligent scheduling of discrete automated production line via deep reinforcement learning. *International Journal of Production Research*. <https://doi.org/10.1080/00207543.2020.1717008>
- Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., & Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*. <https://doi.org/10.1038/nature24270>
- Stricker, N., Kuhnle, A., Sturm, R., & Friess, S. (2018). Reinforcement learning for adaptive order dispatching in the semiconductor industry. *CIRP Annals*. <https://doi.org/10.1016/j.cirp.2018.04.041>
- Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2015). Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11, 812–820.
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. *MIT Press*. <https://doi.org/10.1017/S0263574799271172>
- Takamura, Y., Jain, S., Griffin, P., & Plummer, J. (2002). Thermal stability of dopants in laser annealed silicon. *Journal of Applied Physics*. <https://doi.org/10.1063/1.1481975>

- Thiagarajan, P., Khairmar, P., & Ghosh, S. (2022). Explanation and use of uncertainty quantified by Bayesian neural network classifiers for breast histopathology images. *IEEE Transactions on Medical Imaging*, 41, 815–825.
- Van Rossum, G., & Drake, F. L. (2009). Introduction to python 3: Python documentation manual part 1. *CreateSpace*. <https://doi.org/10.1145/3450613.3456825>
- Vlassis, N., Ghavamzadeh, M., Mannor, S., & Poupart, P. (2012). Bayesian reinforcement learning. *Reinforcement Learning: State-of-the-Art*. https://doi.org/10.1007/978-3-642-27645-3_11
- Wang, H., Shen, H., Liu, Q., Zheng, K., & Xu, J. (2020). A reinforcement learning based system for minimizing cloud storage service cost. In *Proceedings of the 49th international conference on parallel processing. association for computing machinery* (p. 30). <https://doi.org/10.1145/3404397.3404466>
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*. <https://doi.org/10.1016/j.jmsy.2018.01.003>
- Wang, Y.-C., & Usher, J. M. (2005). Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*. <https://doi.org/10.1016/j.engappai.2004.08.018>
- Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., & Kyek, A. (2018). Deep reinforcement learning for semiconductor production scheduling. In *2018 29th annual SEMI advanced semiconductor manufacturing conference (ASMC)* (pp. 301–306).
- Whelan, S., Privitera, V., Italia, M., Mannino, G., Bongiorno, C., Spinella, C., Fortunato, G., Mariucci, L., Stanizzi, M., & Mitiga, A. (2002). Redistribution and electrical activation of ultralow energy implanted boron in silicon following laser annealing. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures Processing, Measurement, and Phenomena*, 20, 644–649.
- White, C. W., Narayan, J., & Young, R. T. (1979). Laser annealing of ion-implanted semiconductors. *Science*. <https://doi.org/10.1126/science.204.4392.461>
- Wiering, M. A., & Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, Learning, and Optimization*. https://doi.org/10.1007/978-3-642-27645-3_1
- Yao, X., Zhou, J., Zhang, J., & Boër, C. R. (2017). From intelligent manufacturing to smart manufacturing for industry 4.0 driven by next generation artificial intelligence and further on. In *2017 5th international conference on enterprise systems (ES)* (pp. 311–318).
- Zhang, Q., Huang, J., Wu, N., Chen, G., Hong, M., Bera, L. K., & Zhu, C. (2006). Drive-current enhancement in Ge n-channel MOSFET using laser annealing for source/drain activation. *IEEE Electron Device Letters*, 27, 728–730.

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