Journal of Manufacturing Processes 137 (2025) 376–396

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

journal homepage: www.elsevier.com/locate/manpro

Adaptive process parameters decision-making in robotic grinding based on meta-reinforcement learning

Jie Pan [a], Fan Chen [a,b,*], Dan Han [c], Shuai Ke [a], Zhiao Wei [a], Han Ding [a,b]

[a] State Key Laboratory of Intelligent Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Luo Yu Road No. 1037, Wuhan 430074, Hubei, China
[b] HUST-Wuxi Research Institute, Wuxi 214174, Jiangsu, China
[c] Jiangsu Jitri-Hust Intelligent Equipment Technology Co., Ltd, Wuxi 214174, Jiangsu, China

ARTICLE INFO

Keywords:
Adaptive grinding process parameters
Meta-reinforcement learning
Material removal accuracy
Robotic grinding
Intelligent manufacturing

ABSTRACT

In robotic grinding, the variability of workpiece characteristics, uneven machining allowances and nonlinear tool wear collectively pose challenges for consistent material removal. To address these dynamic grinding conditions and achieve high-accuracy material removal, this paper presents an adaptive decision-making model for grinding process parameters based on the meta-reinforcement learning. The proposed approach accurately adjusts grinding process parameters under a wide range of coating characteristics, multiple grinding tool types and progressive tool wear, with few-shot training samples. First, we develop an enhanced proximal policy optimization algorithm with better experience (PPO_{BE}) to optimize process parameters for a specific grinding task, improving material removal accuracy. Subsequently, building on the PPO_{BE} framework, we integrate model-agnostic meta-learning (MAML) to form MAML-PPO_{BE} algorithm, enabling fast adaptation across heterogeneous grinding tasks while preserving high accuracy. Comprehensive experiments on 16 distinct grinding tasks demonstrate a 51.4 %–68.9 % improvement in material removal deviation relative to the MAML, PPO_{BE}, SAC and FLC algorithms, respectively. This paper presents an adaptive parameters decision-making method with high accuracy in changing and complex grinding process.

1. Introduction

1.1. Research background

Intelligent manufacturing has emerged as the primary trend and cornerstone of development in the manufacturing industry [1,2]. Characterized by their operational flexibility, expansive workspace and versatile configurations [3,4], industrial robots are widely regarded as paradigms of intelligent manufacturing [5].

The current global landscape presents an urgent demand for the development of aerospace equipment [6]. Advancements in aerospace equipment emphasize lightweight design, stealth capabilities and high-speed flight performance [7,8]. The cabin, as a crucial load-bearing component in aerospace structures, plays a vital role in ensuring structural stability and safety [9]. With the trend toward longer ranges and higher speeds, the thermal protection coating on the exterior surface of the cabin is essential for reducing heat generated by air friction during flight and stabilizing the mechanical performance of the cabin [10,11].

Achieving uniform coating thickness on the surface of the cabin commonly relies on manual grinding. However, automation remains a challenge due to the following reasons: **1) Variable coating thickness:** The coating thickness varies across different areas of the cabin, necessitating accurate grinding to attain a uniform final coating thickness throughout the entire surface. The variation between locations must be controlled within a tolerance of 0.05 mm. Consequently, appropriate grinding process parameters must be continuously adjusted based on real-time measurement results and human expertise. **2) Diverse grinding requirements:** The coating grinding characteristics and grinding technology requirements differ among various cabins, requiring manual selection of suitable sandpapers and real-time adjustments to process parameters to achieve the desired material removal. **3) Dynamic tool wear:** As grinding tools wear dynamically during the process, adjustments to process parameters are necessary at different stages, even when coating characteristics, material removal

* Corresponding author at: State Key Laboratory of Intelligent Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Luo Yu Road No. 1037, Wuhan 430074, Hubei, China.
E-mail address: chenf04@163.com (F. Chen).

https://doi.org/10.1016/j.jmapro.2025.02.005
Received 7 August 2024; Received in revised form 13 January 2025; Accepted 1 February 2025
Available online 8 February 2025

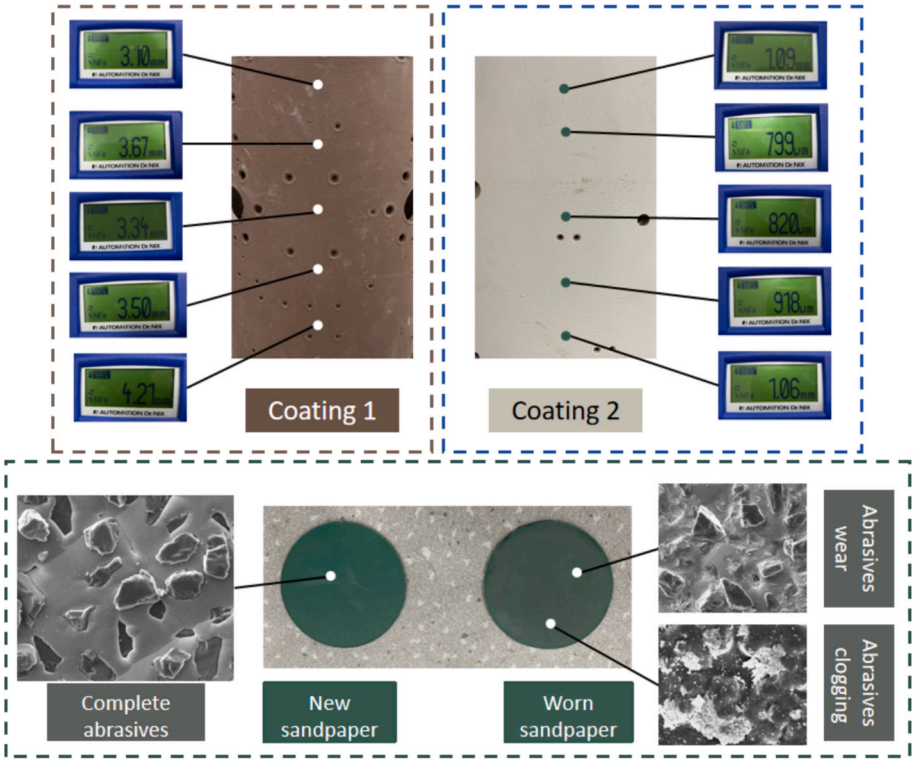

Fig. 1. Schematically illustrates the diverse coating, uneven thickness and dynamic tool wear encountered in robotic grinding.

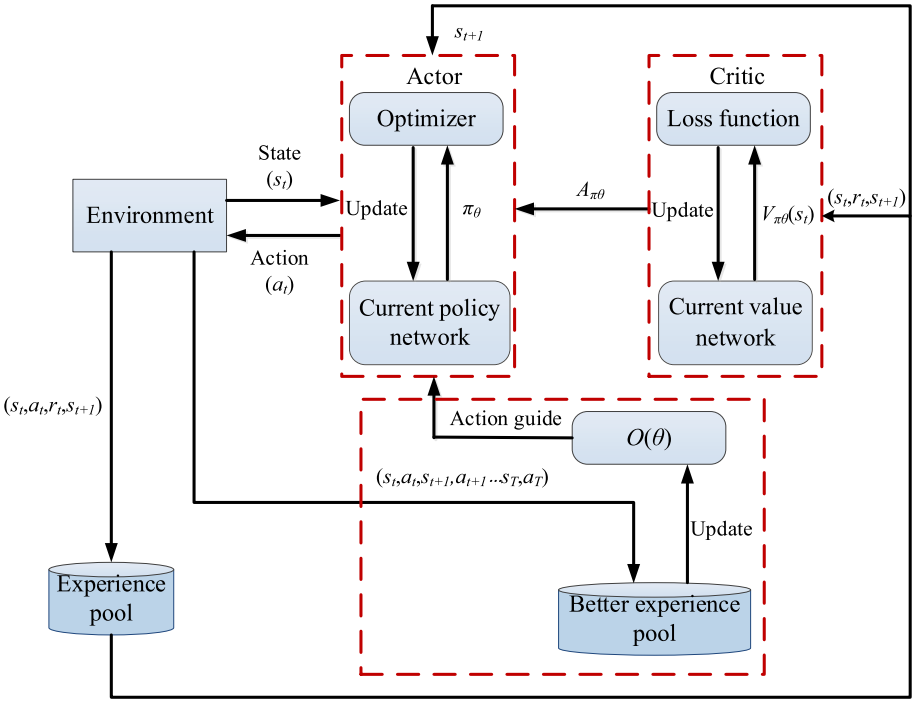

Fig. 2. Architecture of the PPO_{BE} algorithm.

targets and grinding tools are predefined [12,13] (Fig. 1).

1.2. Research status on decision-making of grinding process parameters

In robotic grinding, high variability in workpiece and tool conditions makes adaptive parameter control essential [14,15]. Researchers have addressed this by integrating machine learning algorithms with material removal models, which helps capture complex, nonlinear interactions among parameters. Hiroyuki et al. [16] utilized the random forest data mining method to construct a highly accurate decision support system for grinding wheel operations, continuously optimizing learning data to improve prediction accuracy. Zhang et al. [17] combined random forest and residual convolutional neural networks to predict material removal rates in chemical mechanical polishing. It improves accuracy by

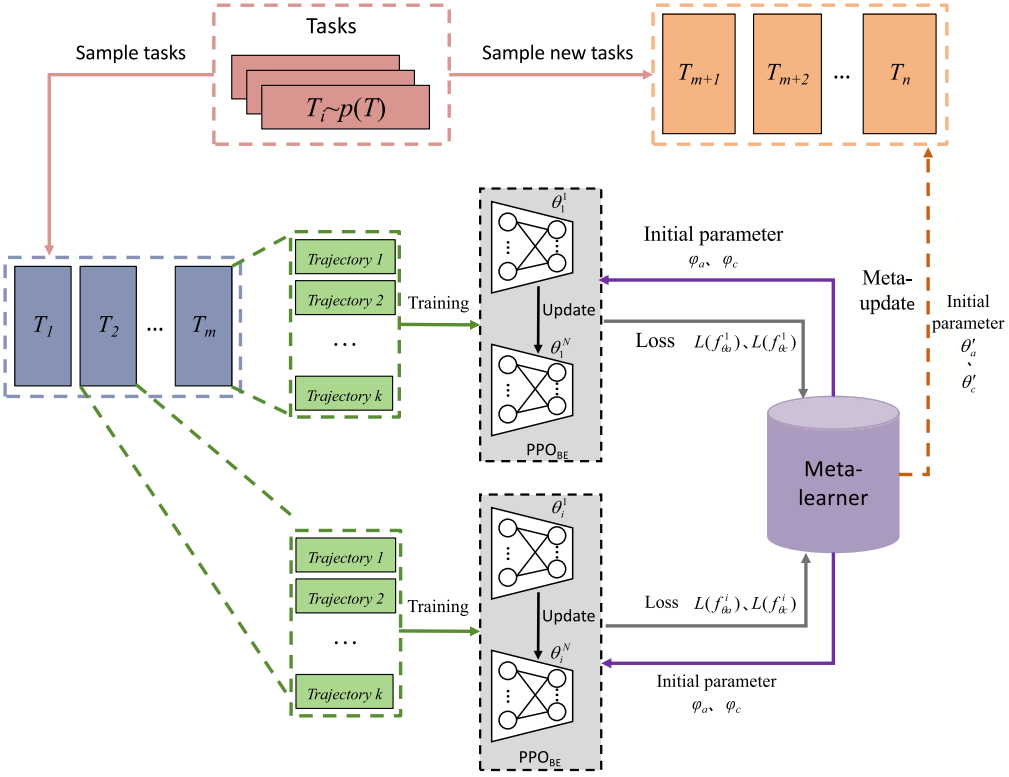

Fig. 3. Framework of MAML-PPO$_{BE}$ algorithm.

preventing gradient vanishing issues. Fu et al. [18] proposed a hybrid model combining physical modeling with LSTM and CNN to reduce grinding deviations by over 80 % post-compensation. Zhao et al. [19] developed an ensemble learning of support vector machine model for surface roughness prediction of LUVAG alumina ceramics. It achieves error reductions of more than 6 % compared to other individual prediction models. Gyeongho Kim et al. [20] introduced a data-driven system and meta-heuristic optimization algorithms to optimize grinding process parameters. Experiments with actual internal cylindrical grinding process data proved the proposed system's effectiveness during process parameter optimization.

However, traditional machine learning algorithms for parameters decision-making in grinding are typically constrained to specific process requirements. It relies on consistent characteristics between training and test samples for effective performance [21]. Models trained using reinforcement learning algorithms [22] or deep learning algorithms [23] struggle with generalization, as variations in training sample attributes often lead to substantial declines in accuracy [24]. This often necessitates retraining the model from scratch.

In contrast, human operators can leverage experience from similar tasks, applying learned knowledge without starting from scratch. Meta-learning operates on a similar principle, it enables agents to gain learning capabilities based on prior tasks, facilitating rapid adaptation to new challenges [25,26]. Reinforcement learning establishes a mapping from one data distribution X to another data distribution Y, whereas meta-learning establishes a mapping from a task set $F(x)$ to the optimal function $f(x)$ corresponding to each task. The knowledge gained from the past task set $F(x)$ helps to quickly establish a mapping for previously unseen tasks $f(x)$ [27,28]. Yu et al. [29] employed meta-learning algorithms to establish a model predicting vehicle driving speeds under varying visual road environments. Li et al. [30] used meta-learning algorithms for bearing fault recognition under changing working conditions and limited samples. Yu et al. [31] proposed a meta-learning algorithm framework to train a 3D facial recognition model. Meta-learning has been applied in various research fields due to its ability to rapidly learn new tasks from few sample data and excellent adaptive capabilities across different tasks. However, to better achieve few-shot learning, the shallow networks used in meta-learning methods lead to relatively poor feature extraction capabilities from samples.

1.3. Contributions

Current research predominantly centers on parameters optimization for a specific workpiece and grinding processes without accounting for the wear dynamics of grinding tools. Furthermore, rapid adaptation to complex and changing grinding conditions remains a gap. To address these challenges, we propose an adaptive decision-making model capable of few-shot parameters optimization with accurate material removal across various grinding tasks. Our main contributions include:

1) **Introduction of the PPO$_{BE}$ algorithm:** We propose the proximal policy optimization with better experience (PPO$_{BE}$) algorithm, This model achieves process parameter decision-making tailored to specific grinding tasks and enhances material removal accuracy. The PPO$_{BE}$ algorithm leverages better experience to train an optimizing function, which directs the agent in action selection. This enhances the efficient utilization of high-quality samples for targeted learning. Compared to the PPO (proximal policy optimization) algorithm, PPO$_{BE}$ demonstrates superior performance in terms of enhanced rewards, reduced loss, increased value and improved material removal accuracy.
2) **Development of the MAML-PPO$_{BE}$ adaptive decision-making model:** Building upon the PPO$_{BE}$ framework, we develop a model-agnostic meta-learning proximal policy optimization with better experience (MAML-PPO$_{BE}$) algorithm for adaptive grinding process parameters. The model facilitates adaptive parameter decision-making across varied coating characteristics, sandpaper grades and tool wear conditions, achieving high material removal accuracy with few-shot training samples. Compared to traditional reinforcement learning algorithms, the MAML-PPO$_{BE}$ algorithm enhances the

Table 1
MAML-PPO$_{BE}$ algorithm flow.

	MAML-PPO$_{BE}$ Algorithm Flow
1	**Hyper-parameters setting: Set** actor network learning rate η_a, critic network learning rate η_c, batch size n, meta-update batch size n', meta-objective function update step size v, discount factor γ, *clip* function limit parameter ε, GAE parameter λ, number of update steps for the actor and critic networks A_U, C_U, optimization factor β, number of trajectories in the better experience pool m;
2	**Parameters setting: Set** maximum number of actions per episode for the agent T, number of tasks sampled per training session T_A, number of episodes in the training set EN_{train}, number of episodes in the training set and testing set EN_{test};
3	**Initialization: Initialize** actor network parameters θ_a, critic network parameters θ_c, optimization network parameters ϕ, experience pool P_E, better experience pool P_{BE}, trajectory return O_n;
4	for ta=1 to TA do
5	Meta-learner provides the initial weights for the actor and critic neural network;
6	for i=1 to EN_{train} do
7	Obtain the initial state s_t;
8	for t=1 to T do
9	According to Eq.**Error! Reference source not found.**, obtain the current policy probability distribution;
10	Select action a_t and obtain reward r_t and the next state s_{t+1};
11	Calculate the cumulative return G=G+r_t;
12	Store the experience trajectories E=(s_t, a_t, r_t, s_{t+1}, v_t) in the experience pool P_E;
13	Store the better experience trajectories P=(s_t,a_t,r_t,s_{t+1},a_{t+1},r_{t+1}……s_T,a_T,r_T) in the experience pool P_{BE};
14	According to Eq.**Error! Reference source not found.**, for each sample, calculate the ratio $r(\phi, \beta)$ of the new policy to the old policy after action selection optimization;
15	Calculate the advantage function $\hat{A}_{\pi\theta}(s_t, a_t)$ based on the value function of the Critic network;
16	According to Eq. **Error! Reference source not found.**, update the actor network parameters θ_a by maximizing the PPO$_{BE}$ objective function;
17	According to Eq. **Error! Reference source not found.**, update the critic network parameters θ_c by minimizing the clipped loss function;
18	According to Eq. **Error! Reference source not found.**, update the optimization network parameters by minimizing the loss function ϕ;
19	Update the policy network parameters θ;
20	end while
21	end for
22	if $O_n \geq \overline{O}_m$ then Place the nth trajectories into the better experience pool P$_{BE}$ and recalculate $\overline{O}_m$;
23	if n trajectories are sampled then
24	Perform the meta-update;
25	end if
26	end for

Table 2
Parameter settings for PPO_{BE} algorithm.

Parameter	Value	Parameter	Value
Actor network update step size A_U	1×10^{-4}	optimization factor β	0.2
Critic network update step size C_U	1×10^{-4}	Value function standard deviation c_1	1
Actor network learning rate η_a	0.001	Policy π_θ entropy parameter c_2	0.01
Critic network learning rate η_c	0.001	Experience pool size P_E	5000
Discount factor γ	0.99	Better experience pool Size P_{BE}	1000
GAE parameter λ	0.95	Number of trajectories in better experience pool m	50
Clip function hyperparameter ε	0.2		

Table 3
Neural network parameter settings for PPO_{BE} algorithm.

Layer	Critic network		Actor network	
	Number of neurons	Activation function	Number of neurons	Activation function
Input layer	State space dimension	None	State space dimension	None
Hidden Layer 1	128	ReLU	128	ReLU
Hidden Layer 2	64	ReLU	64	ReLU
Output Layer	1	None	Action space dimension	Softmax function

Table 4
Training parameter ranges for PPO_{BE} algorithm.

Parameter	Range	Parameter	Range
Target coating thickness H	0.35 mm/0.8 mm/1.2 mm	Grinding contact force F	15N–65N
Actual coating thickness at current position H_a	0.6 mm–3 mm	Grinding rotational speed R_S	100 rpm–600 rpm
Grit size of sandpaper Gs	240–600	Robot movement speed M_S	2 mm/s–10 mm/s
Sandpaper usage time Tu	5 s–60s		

adaptive capability for process parameters, allowing for rapid adaptation to diverse grinding tasks. Furthermore, compared to conventional meta-learning algorithms, MAML-PPO_{BE} optimizes process parameters decision-making capabilities for specific grinding tasks, achieving higher material removal accuracy.

2. Method

2.1. Model of PPO_{BE} algorithm

Due to the fact that the coating characteristics, tool grinding capabilities and the range of grinding process parameters exhibit considerable variability. Effective interaction and coordination among various grinding process parameters are essential to achieve grinding objectives. However, value-based or model-based reinforcement learning algorithms typically require amounts of data samples and computational resources [32]. The PPO algorithm is a reinforcement learning algorithm based on policy gradient optimization and the actor-critic framework, applicable to continuous or discrete action spaces [33]. The PPO algorithm introduces a *Clip* function to restrict the extent of policy updates within a reasonable range. It helps the algorithm to balance between exploring new policies and exploiting existing knowledge. This enables the agent to achieve the maximum reward through interactions with tasks and the environment, thereby improving the algorithm's efficiency, performance and stability. The objective function $L^{clip}(\theta)$ is expressed as [34]:

$$L^{clip}(\theta) \approx \sum_{s_t,a_t} min\left[p_t(\theta)\widehat{A}_{\pi_\theta}(s_t,a_t)\,, clip(p_t(\theta)\,, 1-\varepsilon, 1+\varepsilon)\widehat{A}_{\pi_\theta}(s_t,a_t)\right] \quad (1)$$

In the formula, $p_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ is the importance sampling weight, representing the ratio of the new policy to the old policy. Where $\pi_\theta(a_t|s_t)$denotes the probability of taking action a_t in the given state s_t under the current policy π_θ, $\pi_{\theta old}(a_t|s_t)$ denotes the probability of taking action a_t in the given state s_t under the old policy $\pi_{\theta old}$. The larger $p_t(\theta)$ is, the greater the update magnitude of the new policy relative to the old policy. The $clip(p_t(\theta)\,, 1-\varepsilon, 1+\varepsilon)$ is clipping function used to control the policy update magnitude. It restricts the policy update magnitude $p_t(\theta)$ within the interval [1 − ε, 1 + ε], where ε is the hyperparameter. $\widehat{A}_{\pi\theta}(s_t,a_t) = Q_{\pi\theta}(s_t,a_t) - V_{\pi\theta}(s_t)$ is the estimate of the advantage function, representing the difference between the value of the current state-action pair $Q_{\pi_\theta}(s_t,a_t)$ and the average value $V_{\pi_\theta}(s_t)$ under state s_t. It is used to measure the relative advantage or disadvantage of the current state-action pair compared to the average level.

On this basis, the PPO algorithm also introduces a deviation term regarding the state value function estimate and an entropy regularization term to encourage exploration [35]. The optimized objective function $J^{clip}(\theta)$ is shown in Eq. (2):

$$J^{clip}(\theta) = \sum_{s_t,a_t} min\left[L^{clip}(\theta) - C_1\left(V_{\pi\theta}(s_t) - V_{target}\right)^2 + C_2 H\left(s_t\,, \pi_\theta\right)\right] \quad (2)$$

In the formula, C_1 and C_2 are two constant hyperparameters, $\left(V_{\pi\theta}(s_t) - V_{target}\right)^2$ is the deviation term of the state value function, and represents the entropy of the policy.

In the conventional PPO algorithm's learning and training process, the agent randomly selects actions for the next update, limiting targeted and efficient learning. To guide the agent toward learning from better experience trajectories, enhancing the utilization rate of superior samples and optimizing cumulative returns, we introduce the PPO_{BE} algorithm model. The PPO_{BE} algorithm defines an better experience pool P_{BE} to store trajectories with higher return values. It establishes an optimizing function $O(\phi)$. The optimizing function is trained using trajectories from the better experience pool P_{BE} and guides action selection. During training, trajectories are stored in P_{BE} in the form of (s_t,a_t,r_t,s_{t+1}, a_{t+1},r_{t+1}……s_T,a_T,r_T). As shown in Eq. (3), after each trajectory ends, it is determined whether the trajectory's return is greater than the average return of the trajectories in the P_{BE}. If it is greater, the trajectory is judged as an better trajectory and is placed into the P_{BE}, otherwise, it is discarded.

$$O_n = r_1 + r_2 + \ldots\ldots + r_T \geq \overline{O}_m \quad (3)$$

In the formula, $\overline{O}_m$ represents the average return of the m trajectories in the better experience pool P_{BE} and O_n represents the return of the n trajectory. When the return of the $m + 1$-th trajectory is greater than the average return of the previous m trajectories, the first trajectory is discarded. The $m + 1$-th trajectory is placed into the P_E. When calculating the average return of the trajectories, the cumulative total return becomes $O_2 + O_3 + \ldots\ldots + O_m + O_{m+1}$. Since the size of the better experience pool P_{BE} is fixed, by placing the new trajectory into the P_{BE}, the better trajectories within the pool are updated. This ensures that $O(\phi)$ is as close as possible to the best sample trajectories when guiding action selection.

The optimizing function $O(\phi)$ is trained using the trajectories in the better experience pool P_{BE}. It synchronously minimizes the loss function as the sample trajectories are updated. Its loss function $L_{O(\varphi)}$ is shown in Eq. (4):

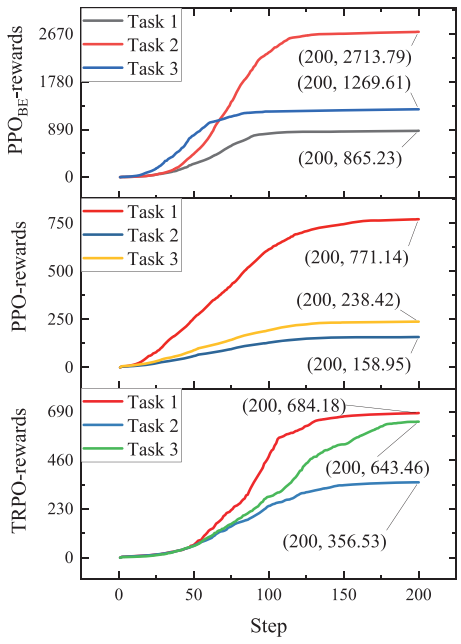

Fig. 4.a Reward training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm

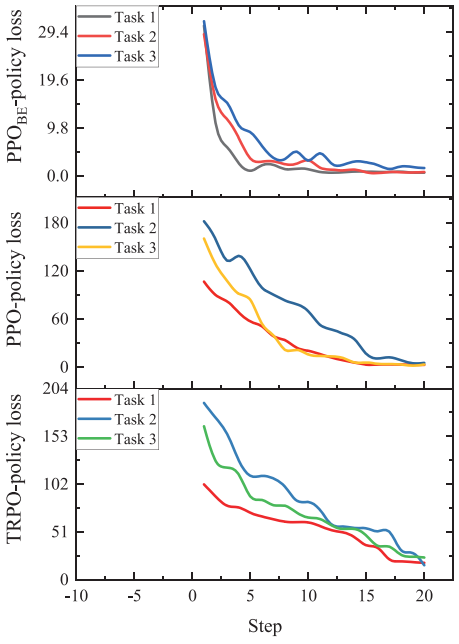

Fig. 4.b Policy loss training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm

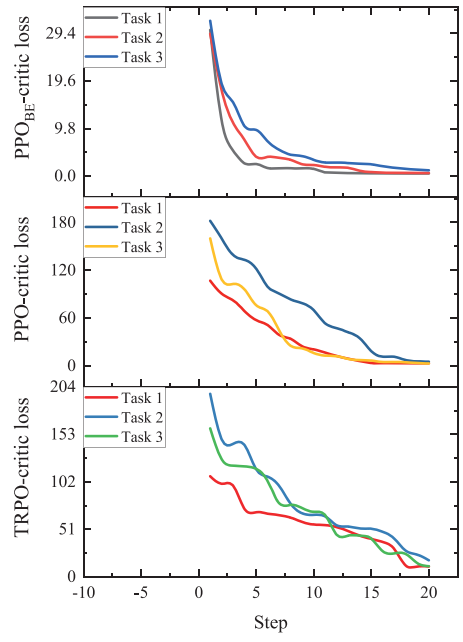

Fig. 4.c Critic loss training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm

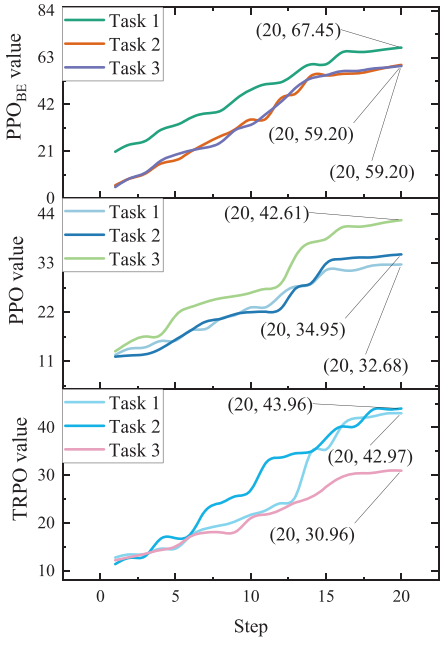

Fig. 4.d Value training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm

Fig. 4. Training and testing curves for PPO$_{BE}$, PPO and TRPO algorithms.
a Reward training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm.
b Policy loss training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm.
c Critic loss training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm.
d Value training curves for PPO$_{BE}$ algorithm, PPO algorithm and TRPO algorithm.
e Material removal deviations test curves for PPO$_{BE}$, PPO and TRPO algorithms.

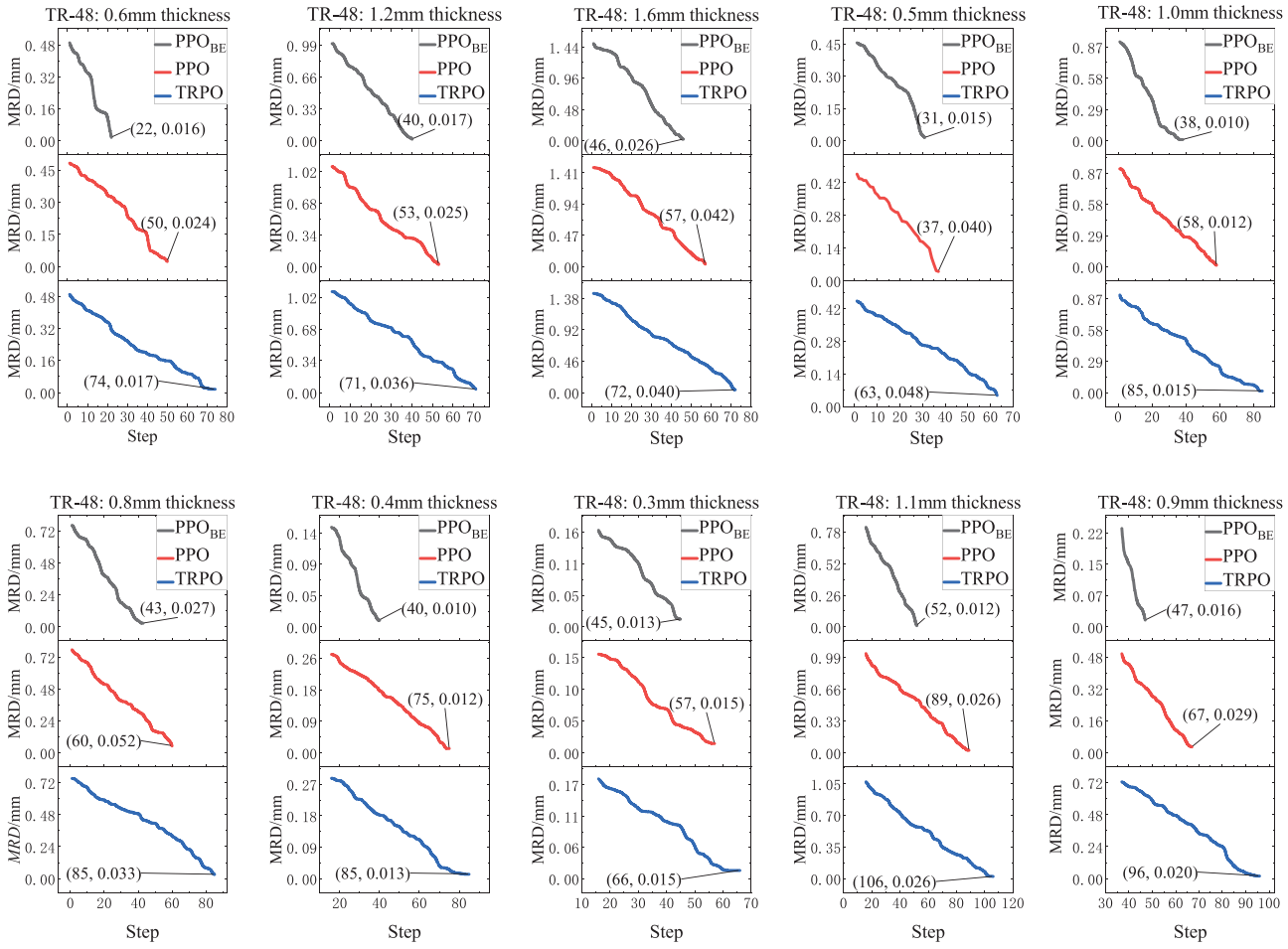

Fig. 4.e Material removal deviations test curves for PPO_{BE}, PPO and TRPO algorithms

Fig. 4. (*continued*).

Table 5
Comparative performance metrics of PPO_{BE}, PPO and TRPO algorithms in training and testing results.

Performance metric	PPO_{BE} algorithm	PPO algorithm	TRPO algorithm
Average reward convergence steps	100	135	145
Maximum reward improvement (%)	PPO_{BE} 326 % more than PPO, 199 % more than TRPO		
Average material removal deviation convergence steps	41	61	81
Material removal deviation (mm)	0.017	0.028	0.027
Policy and critic network loss	Similar		
Policy network loss convergence speed (%)	PPO_{BE} 19 % faster than PPO, 28 % faster than TRPO		
Critic network loss convergence speed (%)	PPO_{BE} 60 % faster than PPO, 68 % faster than TRPO		
Value network loss convergence speed (%)	Similar		
Value improvement (%)	PPO_{BE} 69 % more than PPO, 58 % more than TRPO		

$$L_{O(\varphi)} = \frac{1}{N}\sum_{t=1}^{N}\left[(O(\varphi) - a_t)^2\right] \tag{4}$$

During the learning process, the agent is guided in action selection by the optimizing function $O(\phi)$. With a certain probability, it selects actions for the current state from the probability distribution of the old policy. After being guided by the function $O(\phi)$, the corresponding importance sampling weight transforms as shown in Eq. (5):

$$r(\varphi,\beta) = \frac{\pi_\theta(a_t|s_t) + \beta(O(\varphi)\text{-}\pi_\theta(a_t|s_t))}{\pi_{\theta old}(a_t|s_t)} \tag{5}$$

In the formula, β is a optimization factor constrained within (0,1). $\pi_\theta(a_t|s_t) + \beta(O(\phi)\text{-}\pi_\theta(a_t|s_t))$ represents the probability distribution of the current target policy. Thus, in the PPO_{BE} algorithm, the objective function used to update for the actor network is shown in Eq. (6)

$$J^{PPO_{BE}}(\theta) = \sum_{s_t,a_t} min\left(r(\varphi,\beta)*\widehat{A}_{\pi\theta}(s_t,a_t), C_t^{PPO_{BE}}(\varepsilon)\right) \tag{6}$$

Table 6
Parameter settings for MAML-PPO_{BE} algorithm.

Parameters	Ranges	Parameters	Ranges
Target coating thickness H	0.35 mm–2.2 mm	Coating material type	TR-52/TR-48/TR-42A/TR-42D
Current position actual coating thickness H_a	0.6 mm–4 mm	Grinding contact force F	15N–65N
Grit size of sandpaper Gs	180–800	Grinding rotational speed R_S	100 rpm–600 rpm
Sandpaper usage time Tu	3 s–90s	Robot movement speed M_S	2 mm/s–10 mm/s

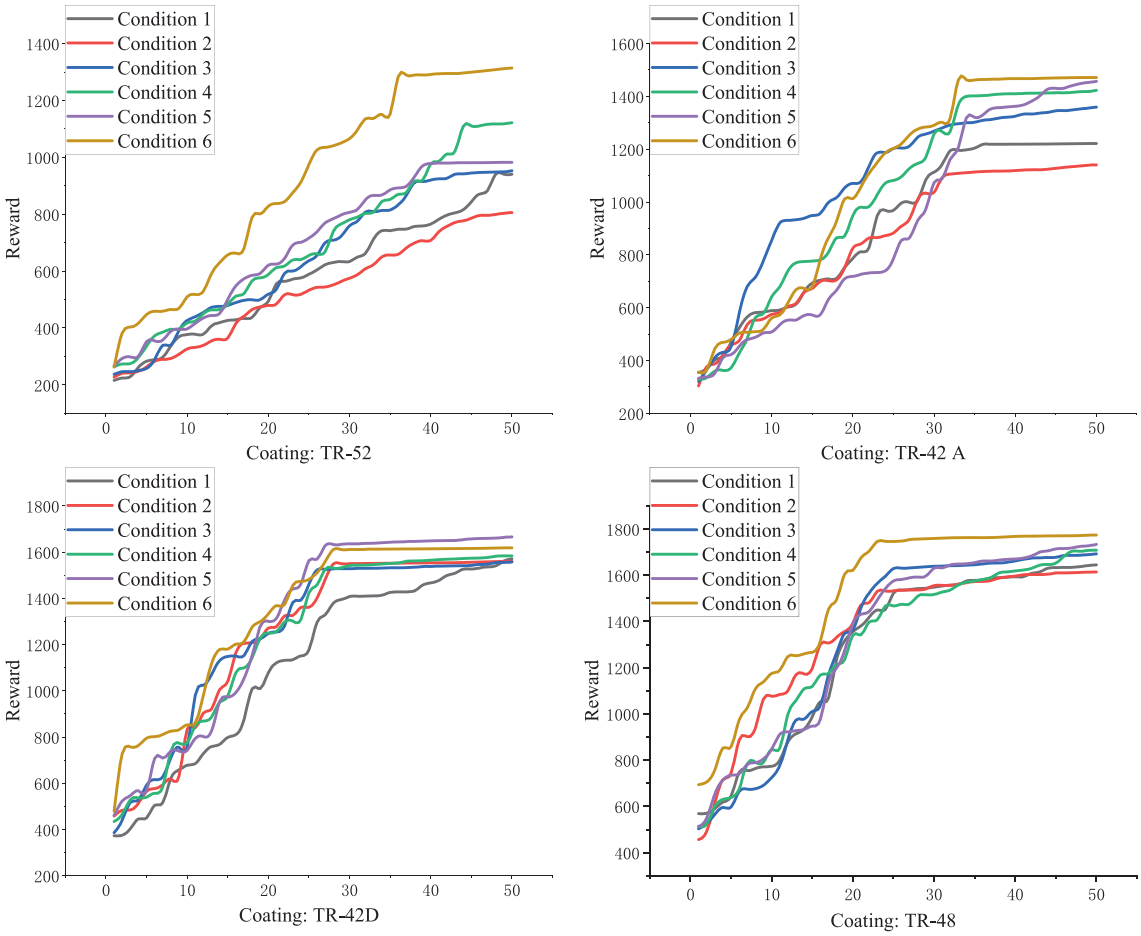

Fig. 5. Reward training curves for MAML-PPO$_{BE}$ algorithm across 24 diverse grinding tasks.

The PPO$_{BE}$ algorithm continues the idea from the PPO algorithm, which involves a clipping function in the objective function. The clipping function in the Eq. (6) is shown in Eq. (7). It is to constrain the importance sampling weights $r(\phi,\beta)$ within the range $[1-\varepsilon, 1+\varepsilon]$.

$$C_t^{PPO_{BE}}(\varepsilon) = \text{clip}(r(\varphi,\beta), 1-\varepsilon, 1+\varepsilon)^*\widehat{A}_{\pi\theta}(s_t,a_t) \tag{7}$$

In the PPO$_{BE}$ algorithm, during the update of the critic network, it employs the approach of minimizing the clipped loss function. The difference between the current state value function $V_{\pi_\theta}(s_t)$ and the previous time step's state value function $V_{\pi_\theta}(s_{t-1})$ is restricted to within $[-\varepsilon, \varepsilon]$. Thus, the constrained state value function $V_{\pi_\theta}^{clip}(s_t)$ is derived as shown in Eq. (8):

$$V_{\pi\theta}^{clip}(s_t) = V_{\pi\theta}(s_{t-1}) + clip(V_{\pi\theta}(s_t) - V_{\pi\theta}(s_{t-1}), -\varepsilon, \varepsilon) \tag{8}$$

Compare the deviation computed from the constrained state value function $V_{\pi_\theta}^{clip}(s_t)$ with the deviation computed from the current state value function $V_{\pi_\theta}(s_t)$ and select the minimum. It effectively reduces the loss of the critic network. This enhancement significantly improves the learning effectiveness of the PPO$_{BE}$ algorithm. The complete objective function of the PPO$_{BE}$ algorithm is shown in Eq. (9):

$$L^{PPO_{BE}}(\theta) = \sum_{(s_t,a_t)} min\left\{L^{clip}(\theta) - C_1 min\left[\left(V_{\pi_\theta}^{clip}(s_t) - V_{target}\right)^2, \left(V_{\pi_\theta}(s_t) - V_{target}\right)^2\right] + C_2 H(s_t,\pi_\theta)\right\} \tag{9}$$

The PPO$_{BE}$ algorithm extends the application of the actor-critic framework from the conventional PPO algorithm. It utilizes two different networks to separate the policy function from the state value function. During training, the critic network evaluates the current policy and provides state and action value functions, which the actor network uses to update the objective function. At each time step t, the actor network executes an action a_t based on the current policy and optimizing function, generating sample experience $P = (s_t, a_t)$, which is stored in the experience pool. After each trajectory ends, it is determined whether the trajectory qualifies to be placed into the better experience pool P_{BE}. Then, optimizing function $O(\theta)$ is synchronously updated using the better experience trajectories in P_{BE}. The architecture of the PPO$_{BE}$ algorithm model is shown in Fig. 2.

2.2. Model of MAML-PPO$_{BE}$ algorithm

Even though the PPO$_{BE}$ algorithm performs well on specific grinding task, it lacks generalization for unknown and random coating characteristics and dynamically wearing grinding tools. To address these limitations, we combine the MAML with PPO$_{BE}$ algorithm, introducing a

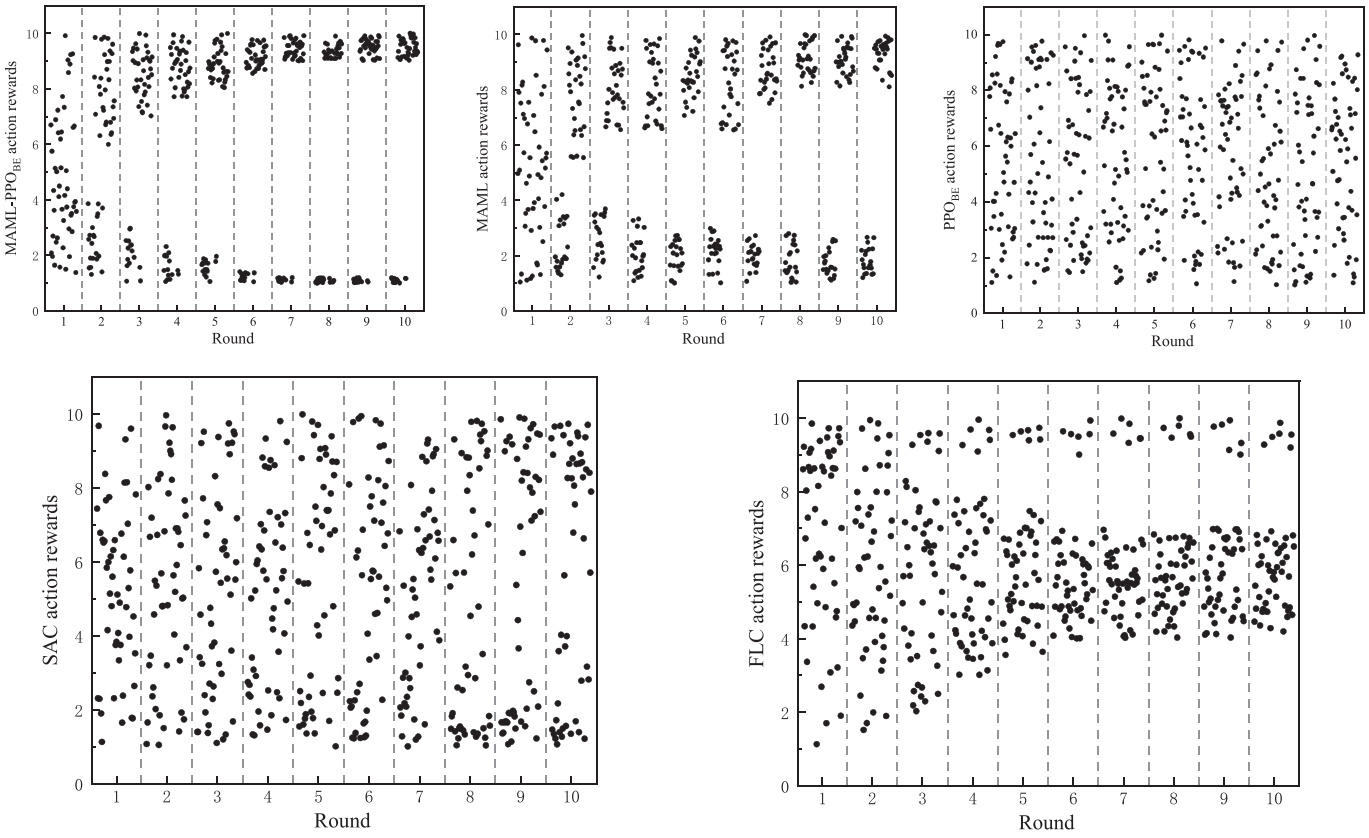

Fig. 6. Action rewards distribution for MAML-PPO$_{BE}$, MAML, PPO$_{BE}$, SAC and FLC algorithms.

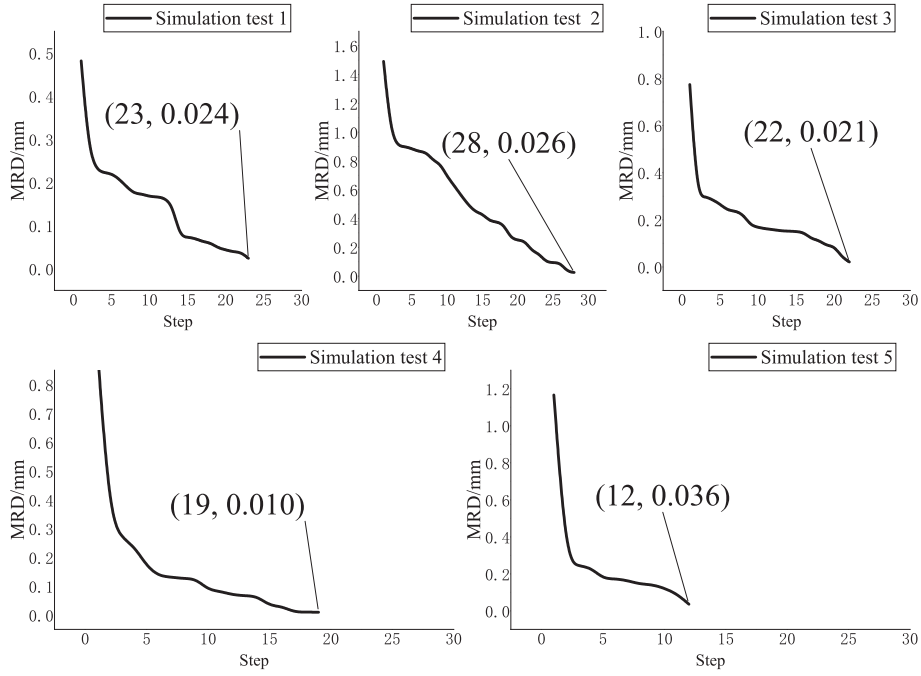

Fig. 7. Simulation test results under 5 different simulation grinding tasks.

MAML-PPO$_{BE}$ algorithm. The MAML algorithm is one of the premier meta-learning methods in the field of few-shot learning. It focuses on learning an optimal model initialization parameter φ, that allows the model to achieve optimal performance on new tasks after one or more gradient updates. The MAML algorithm framework comprises two components: the base learning layer and the meta-learning layer. The base learning layer trains on specific tasks to adapt to their characteristics, while the meta-learning layer synthesizes the training results of multiple tasks, describing the common parameters of the tasks and guiding the base learner [36,37].

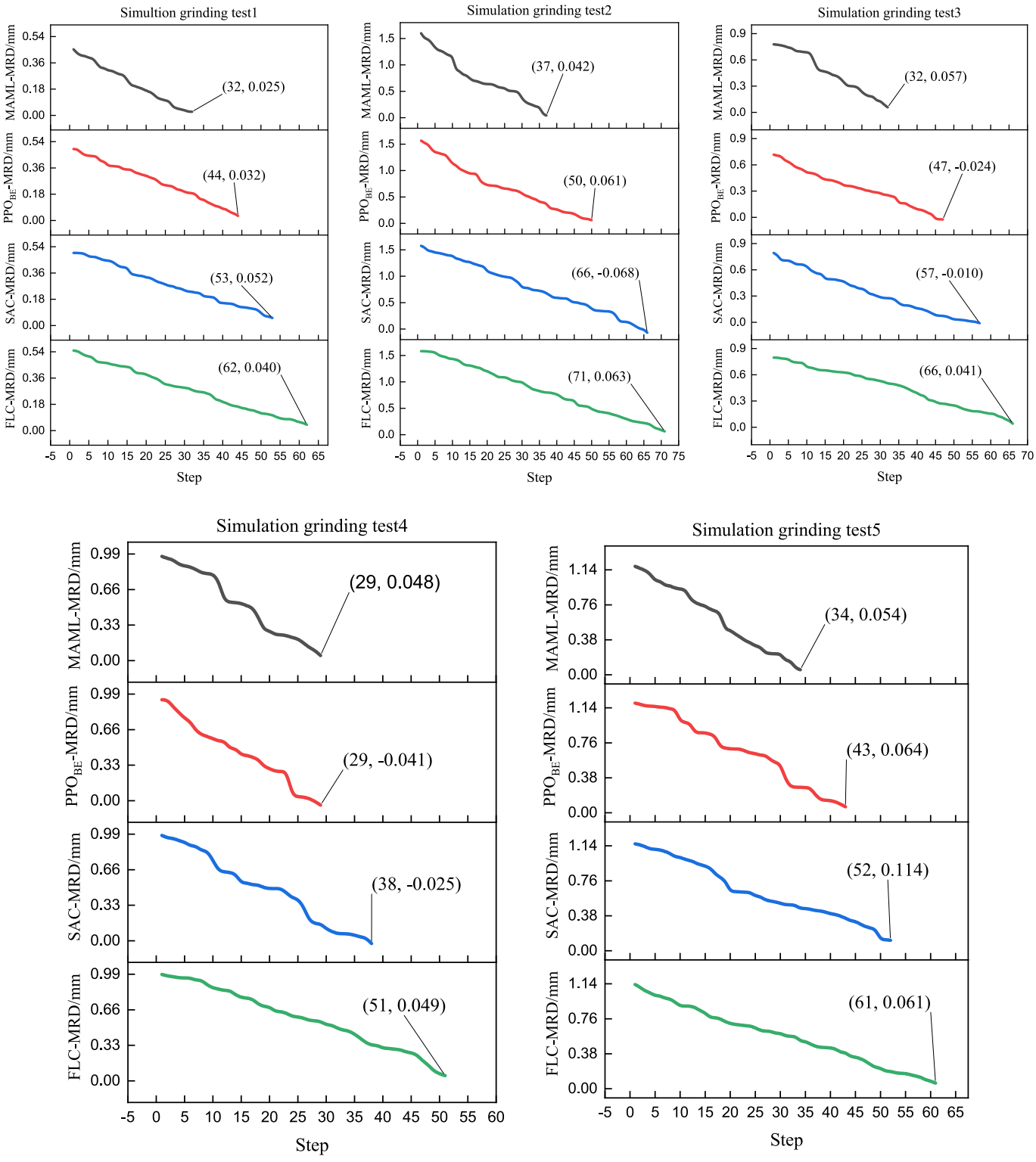

Fig. 8. Material removal deviations simulation results based on the MAML, PPO$_{BE}$, SAC and FLC algorithm under 5 different simulation grinding tests.

Consider a base learner model represented by a parameterized function f_θ, with training parameters θ. Task T_i is randomly sampled from the task distribution $p(T)$. To adapt to task T_i, k data points are sampled from it to train the model. Suitable parameters θ are sought through one or more gradient descent steps to minimize the loss function $L_{Ti}(f_\theta)$. Performing one iteration on task T_i yields:

$$\theta_i^1 = \phi - \alpha[\nabla_\theta L_{Ti}(f_\theta)]_{\theta=\phi} \tag{10}$$

Let φ be the initial parameter values output by the meta-learner to the base learner. Let θ_i^1 be the parameters obtained after one iteration of updating the initial parameters φ for task T_i. α is a hyperparameter representing the meta-update step size. $[\nabla_\theta L_{Ti}(f\theta)]_{\theta=\phi}$ corresponds to the gradient value of the loss function T_i at the initial parameter φ. Similarly, after the N-th iteration:

$$\theta_i^N = \theta_i^{N-1} - \alpha[\nabla_\theta L_{Ti}(f_\theta)]_{\theta=\theta_i^{N-1}} \tag{11}$$

Table 7
Comparative analysis of MAML-PPO$_{BE}$, MAML, PPO$_{BE}$, SAC and FLC algorithms across key performance indicators.

Algorithm	Algorithm category	Reward sensitivity	Average material removal deviations (mm)	Average convergence steps
MAML-PPO$_{BE}$	Meta-reinforcement learning	High	0.0234	21
MAML	Meta-learning	Moderate	0.0452	33
PPO$_{BE}$	Reinforcement learning	Low	0.0444	43
SAC	Deep reinforcement learning	Low	0.0538	53
FLC	Traditional fuzzy control systems	Low	0.0508	62

Table 8
Experimental parameter settings.

Coating material	Target coating thickness H (mm)	Actual coating thickness H_a (mm)	Sandpaper grade	Moving speed of robot (mm/s)
TR-52	1.2/1.0	1.77–3.16	320	3 mm/s
TR-42A	0.5/0.3	0.77–1.31	800	8 mm/s
TR-42D	1.1/0.9	1.59–2.16	600	10 mm/s
TR-48	1.5/1.2	2.12–3.37	240	5 mm/s

In the formula, $[\nabla_\theta L_{Ti}(f_\theta)]_{\theta=\theta_i^{N-1}}$ represents the gradient value of the loss function after $N-1$ iterations. These gradient values are fed back to the meta-learner, which uses them to update the initial parameter φ in the memory module. The objective function of the meta-learner is as follows:

$$\min_{\phi} \sum_{T_i \sim p(T)} L_{T_i}\left(f_{\theta_i^N}\right) \tag{12}$$

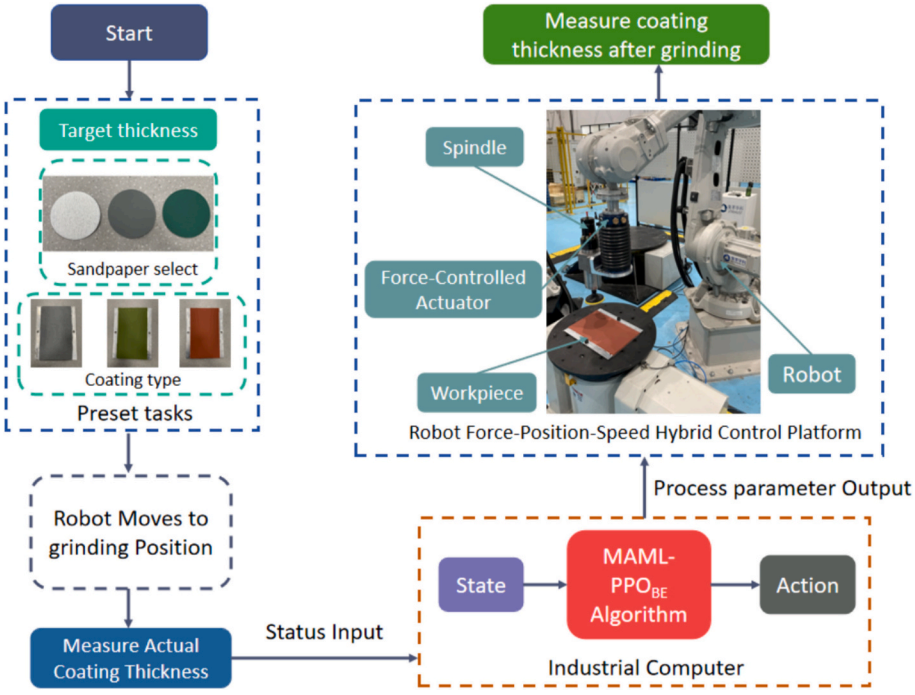

Fig. 9. Principle of the experimental platform for robotic coating grinding process parameter adaptive decision-making.

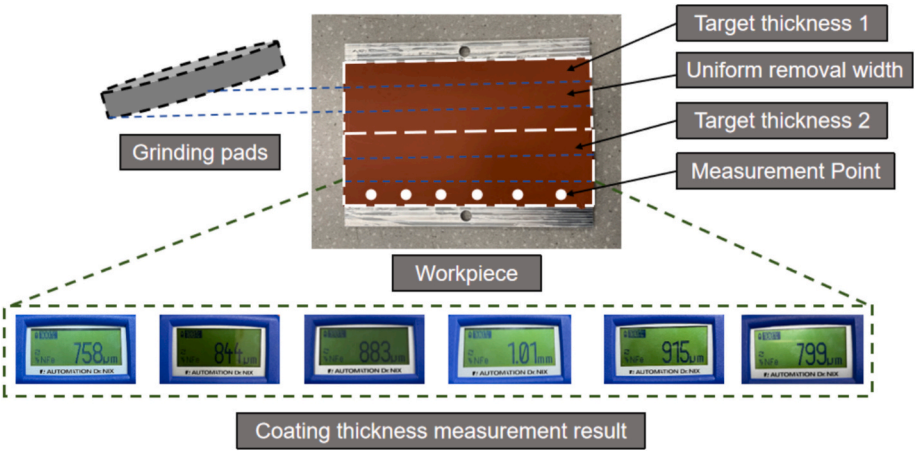

Fig. 10. Measurement of coating thickness before grinding.

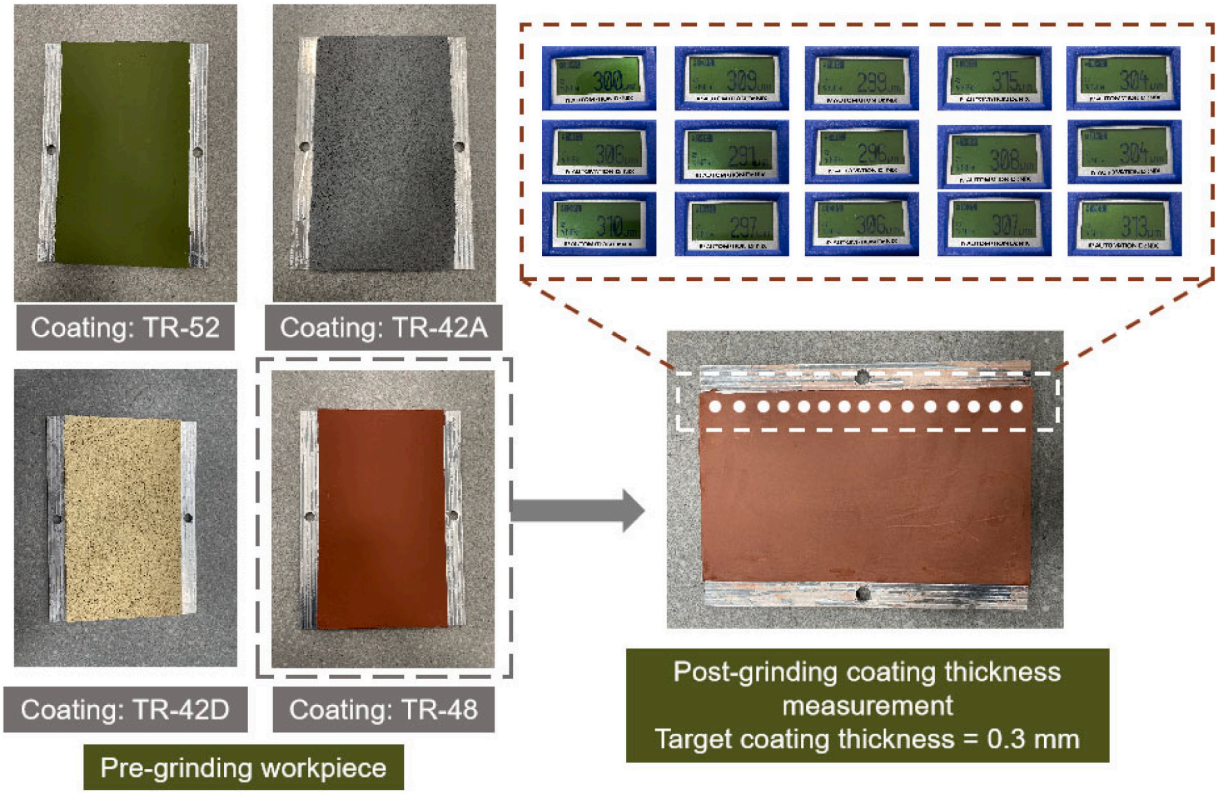

Fig. 11. Measurement of coating thickness after grinding.

By optimizing the initial parameter values φ of the meta-learner's objective function, the sum of losses over all tasks in the task distribution $p(T)$ is minimized. Based on the initial parameter values φ, through continuous gradient updates, the optimal parameter set $\theta^{'} = \{\theta_1^{'}, \theta_2^{'}, \ldots\ldots \theta_i^{'}\}$ for the sampled i tasks can be obtained. At the same time, the gradients of the loss function $[\nabla_\theta L_{Ti}(f\theta)]_{\theta=\theta'}$ corresponding to these optimal parameters can be obtained.

Using these loss function gradients to update the initial parameter values φ. The updated initial parameter values ϕ' allow the training of a new batch of tasks T_n to start from a better position, without requiring a large number of gradient steps. The expression is as follows:

$$\phi' \leftarrow \phi - \delta \sum_{T_i \sim p(T)} \nabla_\phi \left[L_{T_i}(f\theta) \right]_{\theta=\theta_i^{'}} \tag{13}$$

In the formula, δ is a hyperparameter representing the step size for optimizing the meta-objective function in the meta-learner. $\nabla_\phi \left[L_{T_i}(f\theta) \right]_{\theta=\theta_i^{'}}$ represents the gradient of the loss function with respect to the initial parameter φ, when the optimal parameters $\theta_i^{'}$ for task T_i are obtained through iterative training.

In this paper, an adaptive decision-making model for grinding process parameters is designed by combining the MAML algorithm with the PPO_{BE} algorithm. In this integrated approach, the PPO_{BE} algorithm serves as the base learner in the meta-reinforcement learning framework, while the MAML algorithm functions as the meta-learner. The PPO_{BE} algorithm searches for the optimal policy through the parameters θ of the policy π_θ, while the MAML algorithm finds the optimal parameters θ' that can adaptive across tasks. The PPO_{BE} algorithm includes both actor and critic networks, each with its own loss function. Therefore, the loss function of the MAML-PPO_{BE} algorithm can be defined as the networks $f_{\theta a}$ and $f_{\theta c}$. They respectively take the loss parameters θ_a and θ_c from the PPO_{BE} algorithm as parameters, along with a task distribution $p(T)$. By randomly initializing θ_a and θ_c, we obtain the initial parameters φ_a and φ_c.

A batch of tasks T_i is sampled from $p(T)$ and $T_i \sim p(T)$. Then k trajectories are sampled from each task to construct training and testing sets: $D_i^{train} D_i^{test} \sim T_i$. Through the PPO_{BE} algorithm, optimal parameters $\theta_a^{'}$ and $\theta_c^{'}$ for the actor and critic networks are found to minimize the losses $L\left(f_{\theta a}^{i}\right)$ and $L\left(f_{\theta c}^{i}\right)$ on the training set D_i^{train}. Before sampling the next batch of tasks, a meta-update is performed. It computes the loss gradients relative to the optimal parameters $\theta_a^{'}$ and $\theta_c^{'}$ to minimize the loss on the testing set D_i^{test}, thereby updating the randomly initialized parameters of two networks. The framework of the MAML-PPO_{BE} algorithm is shown in Fig. 3, and the algorithm flow is detailed in Table 1.

3. Simulation and analysis

3.1. Simulation and analysis of PPO_{BE} algorithm

The core of reinforcement learning algorithms is the Markov decision process (MDP), which mainly comprises five elements: (S, A, R, P, γ) [38]. This paper firstly design an MDP for the decision-making model of grinding process parameters based on the PPO_{BE} algorithm.

1) State space design

The design of the state space needs to consider the characteristics of the grinding process and requirements. It should clearly define the target coating thickness H and the actual coating thickness H_a at the current position. It also need to consider the material removal rate M_a of the sandpapers at the current moment. Therefore, the state space S is (H, H_a, M_a).

The physical quantities in the state space have different scales and units, it is necessary to normalize the state space to achieve dimensionless state values on the same scale. Thus, the normalized state space S_0 is $(\frac{H}{H_0}, \frac{H_a}{H_{a0}}, \frac{M_a}{M_0})$, where H_0 represents the mean target coating thickness, H_a represents the mean actual coating thickness and M_0 represents the

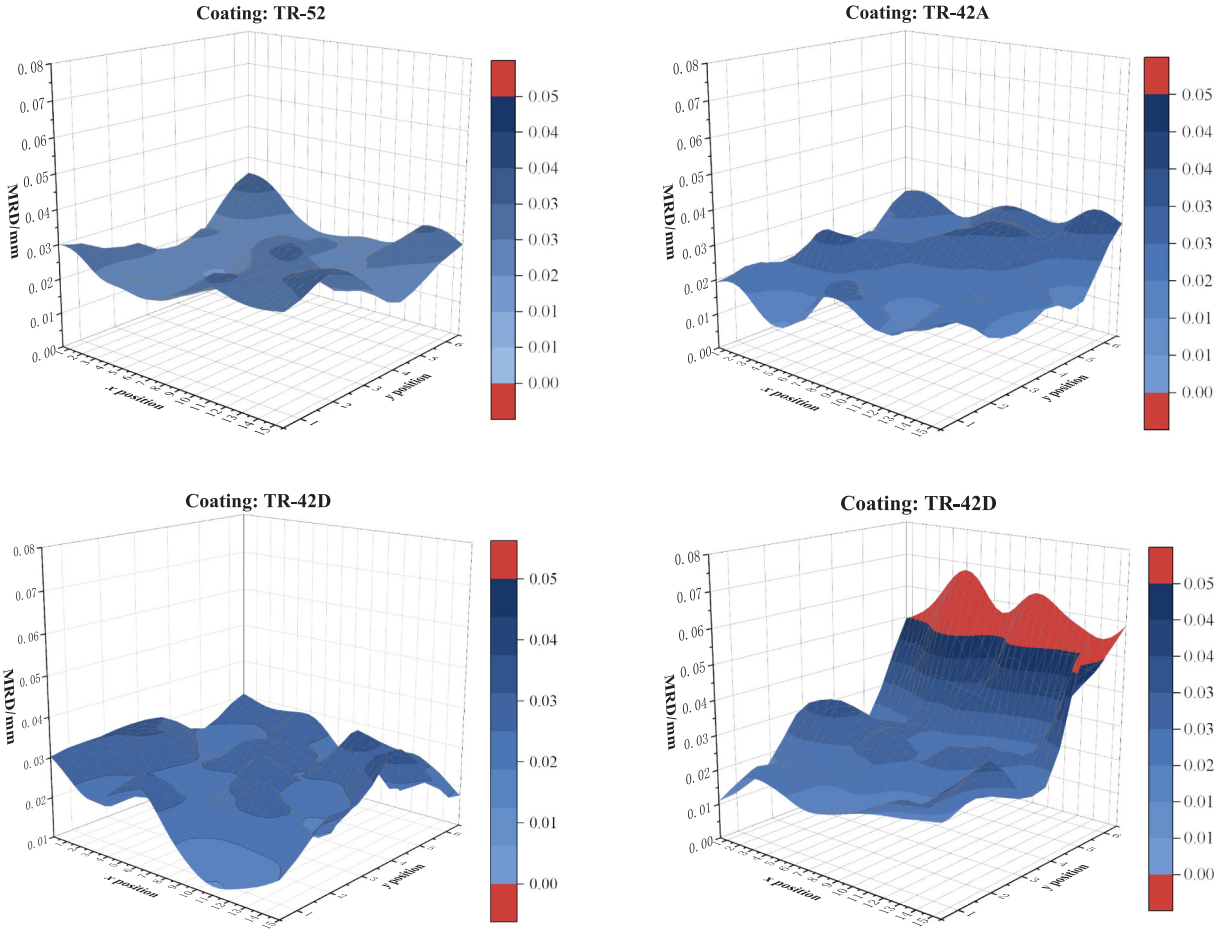

Fig. 12. Distribution of material removal deviations after grinding with process parameters calculated based on the MAML-PPO$_{BE}$ algorithm.

initial material removal rate of the sandpapers.

2) Action space design

The adaptive decision-making model for grinding process parameters achieves the target coating thickness H by adjusting the grinding process parameters in real-time. The main process parameters that affect the material removal rate are the grinding contact force F, the grinding rotational speed R_s and the movement speed of robot M_s. Therefore, the action space A is (F, R_s, M_s). Similarly, the action space is normalized to obtain a normalized action space A_0 ($\frac{F}{F_0}$, $\frac{R_s}{R_{s0}}$, $\frac{M_s}{M_{s0}}$), where F_0, R_{s0} and M_{s0} represent the mean values of the grinding contact force, grinding rotational speed and robot movement speed, respectively.

3) Design of the reward function

The reward function evaluates the effect of an agent's action in each state, guiding the adjustment of the agent's behavior policy [39]. The primary objective of the adaptive decision-making for grinding process parameters is to achieve accurate material removal without over-grinding. Accordingly, a reward function incorporating both rewards and penalties is designed. When the agent's actions bring the current coating thickness closer to the target thickness, a positive reward is given. Conversely, if the current coating thickness falls below the target coating thickness, a penalty is imposed.

$$R = \begin{cases} e^{(H_a - H)^{-1}}, & H_a > H \\ +\infty, & H_a = H \\ -e^{(H - H_a)}, & H_a < H \end{cases} \tag{14}$$

The reward function, defined in Eq. (14), provides feedback based on the relationship between the actual coating thickness H_a and the target coating thickness H.

When $H_a > H$: If the actual thickness is greater than the target, the function gives a positive reward $e^{(H_a - H)^{-1}}$. The inverse in the exponent exhibits a heightened response as H_a approaches H, effectively magnifying subtle variations and enhancing the function's sensitivity to minor deviations. This increased sensitivity is critical for promoting accurate adjustments toward the target thickness, thereby ensuring that the reward structure fosters an accurate and stable optimization process.

When $H_a = H$: If the actual thickness matches the target exactly, an infinite reward $+\infty$is assigned. This value serves as an ideal point, strongly incentivizing the agent to achieve the exact target thickness.

When $H_a < H$: This negative reward is designed to deter actions that could lead to over-grinding. If the actual thickness H_a marginally less than the target thickness H, the penalty $e^{(H_a - H)^{-1}}$ becomes significantly pronounced. It effectively prompts an immediate cessation of the iteration process. This mechanism guarantees that even in cases of minor overshooting during the grinding process, material removal is kept to a minimum, thereby preserving accuracy and mitigating excessive material loss.

To verify the effectiveness of the PPO$_{BE}$ algorithm, the PPO and TRPO (Trust Region Policy Optimization) algorithms were used as benchmark algorithms. Both PPO and TRPO algorithms are well-regarded for their ability to enhance training stability and efficiency in reinforcement learning applications. The PPO algorithm is selected as a benchmark algorithm for the PPO$_{BE}$ algorithm due to its simplicity and reliable performance, which allows for stable policy updates while ensuring computational efficiency in robotic grinding situations. TRPO

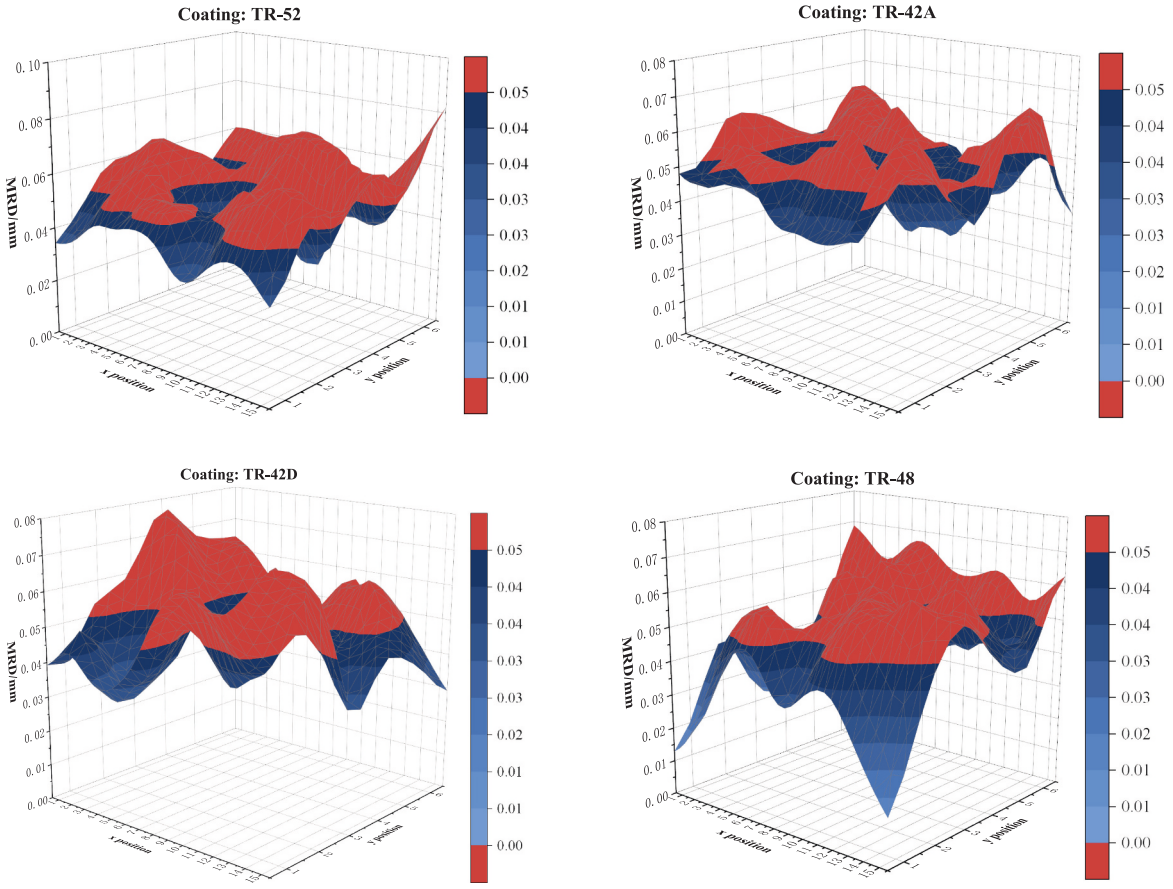

Fig. 13. Distribution of material removal deviations after grinding with process parameters calculated based on the conventional MAML algorithm.

algorithm, on the other hand, provides an additional perspective on stability by employing trust region constraints, making it particularly effective for maintaining robustness in complex environments. By comparing PPO_{BE} algorithm with these established algorithms, we aim to provide a comprehensive evaluation of its strengths in balancing stability and efficiency, crucial for complex and high-accuracy applications such as robotic grinding.

Comparative simulations were conducted to evaluate the performance of the PPO_{BE} algorithm. During the algorithm training process, the policy is optimized across multiple episodes by maximizing a clipped surrogate objective function. Specifically, PPO_{BE} algorithm builds upon the traditional PPO algorithm by incorporating better experience to guide action selection, enhancing the optimization of policy decisions for a specific type of grinding tasks. In the simulation analysis, the network structure and parameters used for different grinding tasks were kept consistent. The specific algorithm parameters are shown in Table 2 and the neural network parameters are detailed in Table 3.

Choose TR-48 thermal protection coating and set three target coating thicknesses: 0.35 mm, 0.8 mm and 1.2 mm. Measure the actual coating thickness on the workpiece surface. Conduct grinding experiments with different combinations of process parameters (grit size of sandpaper *Gs*, sandpaper usage time *Tu*, grinding rotational speed R_s, grinding contact force *F* and robot movement speed M_s) to achieve the target coating thickness and collect data. Use these data as training samples to train the model. For each characteristic coating thickness, collect 200 training samples and set aside 10 test samples. The range of training parameters is shown in Table 4.

Fig. 4 illustrates the evolution of reward training curves, loss training curves, value training curves and material removal deviation testing curves for PPO_{BE} algorithm, PPO algorithm and TRPO algorithm under 3 target coating thickness requirements. The term “step” on the x-axis signifies a complete state-action-reward sequence in the training process of the PPO_{BE} and comparison algorithms. From Fig. 4, it is evident that the PPO_{BE} algorithm achieves higher rewards and values, faster loss convergence rates and smaller material removal deviations compared to the PPO and TRPO algorithms. From Fig. 4.a, it can be observed that the PPO_{BE} algorithm reaches reward convergence in an average of 100 steps, while the PPO and TRPO algorithms require averages of 135 steps and 145 steps, respectively. Additionally, compared to the PPO and TRPO algorithms, the PPO_{BE} algorithm shows average maximum reward improvements of approximately 326 % and 199 % across different target coating thickness requirements.

According to the reward formula (14), higher rewards correspond to reduced deviations in material removal. On testing samples, the coating thickness difference is set to be less than 0, meaning that if over-grinding occurs, iteration is immediately stopped. We demonstrate the trained reinforcement learning model’s application in selecting optimal grinding parameters to control coating thickness. Using inputs of current and target coating thickness, as well as the coating characteristic, the model selects actions, such as grinding parameters, which is based on learned policies. Thickness changes are simulated using a state transition function, derived from a combination of physical modeling and training data, to predict the impact of each grinding action. Deviations from the target thickness are calculated to guide subsequent actions.

Fig. 4.e shows that the PPO_{BE} algorithm achieves convergence of material removal deviation (MRD) in approximately 41 steps, with an average deviation between target and actual coating thicknesses as low as 0.017 mm. In comparison, the PPO and TRPO algorithms require approximately 61 steps and 81 steps, resulting in post-training material removal deviations of 0.028 mm and 0.027 mm, respectively.

Fig. 4.b, .c and .d indicates that while the PPO_{BE} algorithm achieves minimal policy and critic network losses similar to those of the PPO and

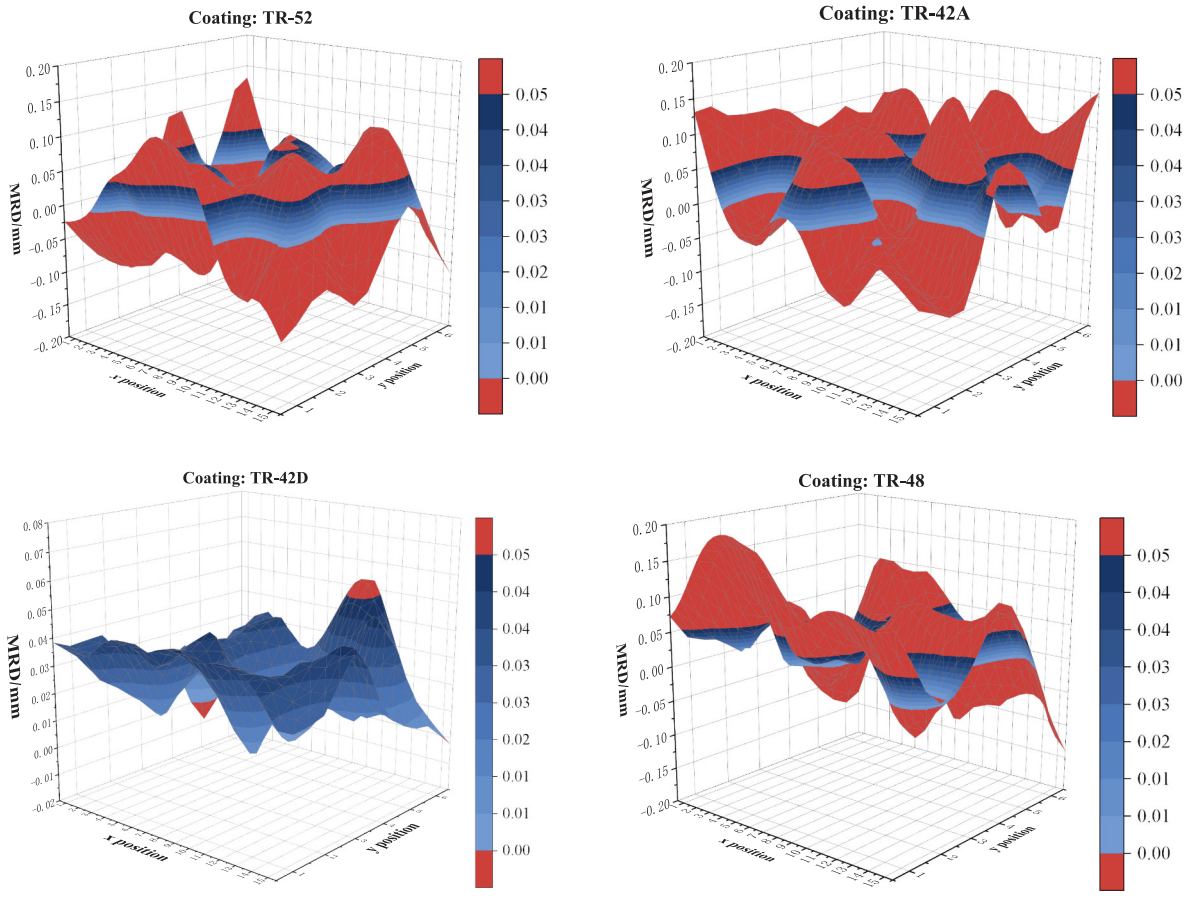

Fig. 14. Distribution of material removal deviations after grinding with process parameter calculated based on the PPO_{BE} algorithm.

TRPO algorithms. It exhibits 19 % and 28 % faster convergence in policy network and 60 % and 68 % faster convergence in critic network, respectively. The PPO_{BE} algorithm demonstrates comparable convergence rates to PPO and TRPO algorithms in value network however, it achieves an average maximum value improvement of 69 % and 58 % respectively. This indicates that the PPO_{BE} algorithm offers significant advantages in action evaluation, selection and execution compared to the PPO and TRPO algorithms. This further demonstrates that the approach guided by its own better experiences can direct the agent to effectively utilize optimal strategies during training, leading to excellent training results. To provide a more comprehensive and structured comparison of algorithmic performance, we have summarized the key performance metrics of the PPO_{BE}, PPO and TRPO algorithms in terms of both training and testing results into Table 5.

3.2. Simulation and analysis of MAML-PPO_{BE} algorithm

The adaptive decision-making model for grinding process parameters, based on the MAML-PPO_{BE} algorithm, is trained within the same MDP framework as the PPO_{BE} algorithm. This MAML-PPO_{BE} model is designed to address tasks with various types of coatings, requirements and sandpapers. During the training process, the agent interacts with the environment to gather information about the state space and corresponding reward values. The policy is then continuously adjusted based on the obtained state information and rewards. A notable difference in the MAML-PPO_{BE} algorithm is the use of meta-reinforcement learning. Unlike traditional reinforcement learning, which focuses on a single task, the MAML-PPO_{BE} algorithm trains the model across a diverse set of tasks. This approach enables the model to quickly adapt to new tasks by leveraging knowledge gained from previous ones. Meta-learning updates allow the model to learn a set of initial parameters that can be fine-tuned efficiently, accelerating the training process and improving the model's ability to generalize across different grinding tasks. The training parameters for the model are set as shown in Table 6.

Differentiating by the type of coating materials, the entire training data set is divided into 4 categories of tasks. Each category of coating is assigned 2 target coating thicknesses and 3 different grinding capabilities sandpapers are selected, resulting in a total of 24 tasks. The PPO_{BE} algorithm is used to find the optimal parameters to minimize the loss on the training set. Before proceeding to the next batch task, a meta-update is performed. This involves minimizing the loss on the test set by calculating the gradient of the loss with respect to the optimal parameters, thereby updating the randomly initialized parameters.

The main objective of designing the adaptive process parameters decision-making model for multi-grinding tasks is to enhance the model's adaptive capability with a small amount of training data, while ensuring the accuracy of material removal. Each task collects 50 data samples for gradient updates (fewer than the 200 samples required by the PPO_{BE} algorithm mentioned above), with 5 types of grinding tasks set for testing purposes. In this process, the focus is on optimizing parameters across multiple training sets of tasks, with the goal of generalizing learning to new tasks. Unlike traditional reinforcement learning, which iterates within a single task to refine policy decisions based on step-by-step interactions with the environment. Meta-reinforcement learning evaluates broader optimization results across various tasks. As a result, the learning process is organized into rounds, where each round encompasses several episodes. It aims at improving the algorithm's overall training performance across multiple tasks and its ability to quickly adapt to new tasks.

The trend in Fig. 5 illustrates the MAML-PPO_{BE} model's efficient reward convergence and adaptive action evaluation, demonstrating its ability to quickly optimize parameters across varying tasks with minimal

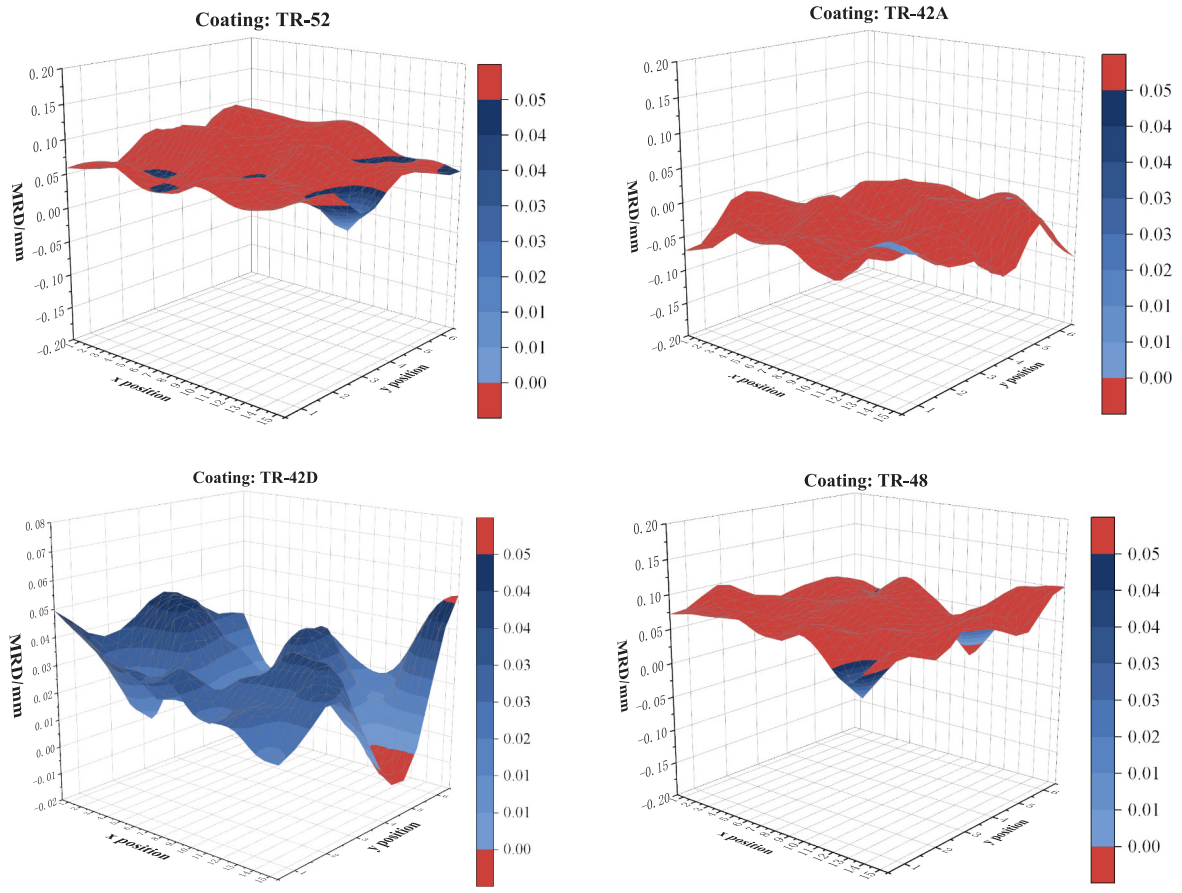

Fig. 15. Distribution of material removal deviations after grinding with process parameters calculated based on the conventional SAC algorithm.

sample data. This reflects the model's flexibility in enhancing material removal accuracy and process parameters optimization. In Fig. 5, the observed trend underscores the MAML-PPO$_{BE}$ model's robustness and generalization ability. Despite limited training samples (50 per task) across 24 diverse grinding tasks, the model achieves rapid reward convergence. It averaged within 32 training rounds across varying coating materials and sandpaper capacities. Notably, the difference in average maximum reward between the MAML-PPO$_{BE}$ and the PPO$_{BE}$ model mentioned above, which utilizes significantly larger sample data on specific tasks, is a mere 7.3 %. As task diversity increases, the model performs meta-updates that optimize initial parameters, enabling a more advantageous starting point for subsequent reward learning. By the fifth batch of tasks, convergence accelerates further, achieving stabilization within only 26 training rounds. This trend signifies the model's advanced adaptive capacity in parameter initialization, facilitating efficient reward optimization across heterogeneous task sets.

We rigorously evaluate the MAML-PPO$_{BE}$ algorithm to validate its theoretical advancements and practical effectiveness in optimizing action selection. For a robust comparison, we select MAML, PPO$_{BE}$, Soft Actor-Critic (SAC) and Fuzzy Logic Controllers (FLC) as benchmark algorithms, each representing core approaches in meta-learning, reinforcement learning, deep reinforcement learning and traditional fuzzy control, respectively.

The MAML algorithm is included due to its strong capability in meta-learning, particularly for rapid adaptation and generalization in few-shot training samples. This enables us to evaluate the adaptability of the MAML-PPO$_{BE}$ algorithm across various unseen tasks. PPO$_{BE}$ algorithm serves as a baseline to assess the MAML-PPO$_{BE}$ algorithm's ability to achieve efficient and accurate action selection in specific tasks. It provides a clear basis for performance comparison. SAC algorithm is known for its balanced approach to exploration and exploitation through soft-policy optimization, which helps to evaluate the robustness of the MAML-PPO$_{BE}$ algorithm in complex and high-dimensional action spaces. Additionally, the FLC, a well-established and simple rule-based method for handling uncertainty and non-linearity, is chosen as a comparative algorithm. It emphasizes the advantages of MAML-PPO$_{BE}$ algorithm in dynamic conditions.

Each "black point" in Fig. 6 represents the reward value corresponding to actions evaluated by different learning algorithms. The term "round" on the x-axis in Fig. 6 refers to a structured phase in the meta-reinforcement learning training process. As shown in Fig. 6, the MAML-PPO$_{BE}$ algorithm demonstrates marked advantages in action evaluation and refinement. During the early stages of algorithm iteration, the reward values display a wide distribution within the range [0, 10]. As the algorithm iteratively refines, actions that significantly improve material removal accuracy receive higher rewards, while less effective actions are assigned lower rewards. This reflects the algorithm's capacity for distinguishing action efficacy, which is instrumental in accurate decision-making under complex conditions.

In comparison, the conventional MAML algorithm, though equipped with an action evaluation network, displays reduced sensitivity in reward differentiation. It yielded more homogenized reward distributions across actions. This less differentiated reward assignment impedes both learning stability and convergence efficiency.

Both PPO$_{BE}$ and SAC algorithms exhibit less effective reward differentiation. Although the PPO$_{BE}$ algorithm represents an improvement over the conventional PPO algorithm, it still requires multiple steps of training to progressively filter out high-quality experiences, thereby enabling effective differentiation of reward values across actions. This reliance on large amounts of training samples limits PPO$_{BE}$ algorithm's adaptability and convergence rate in reward value within data-constrained scenarios. The SAC algorithm, through its stochastic

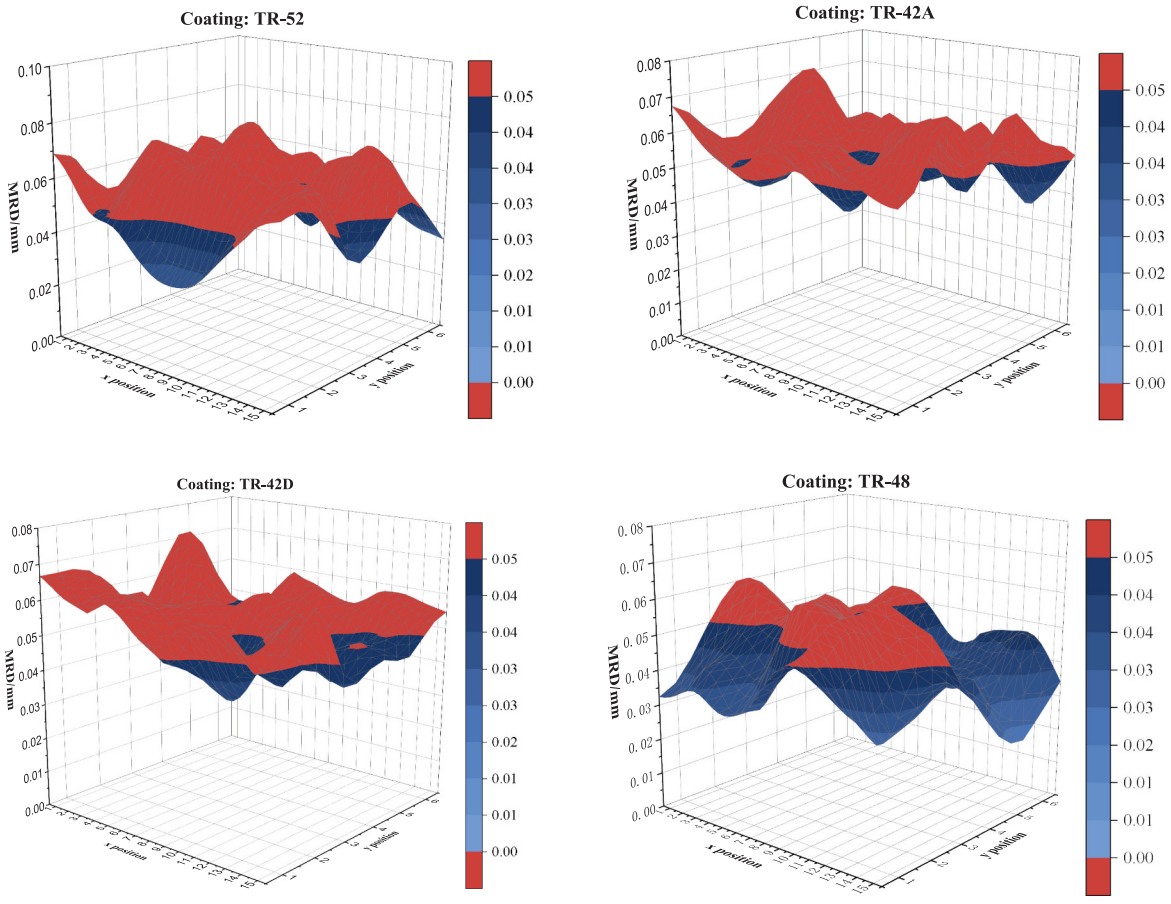

Fig. 16. Distribution of material removal deviations after grinding with process parameters calculated based on the FLC algorithm.

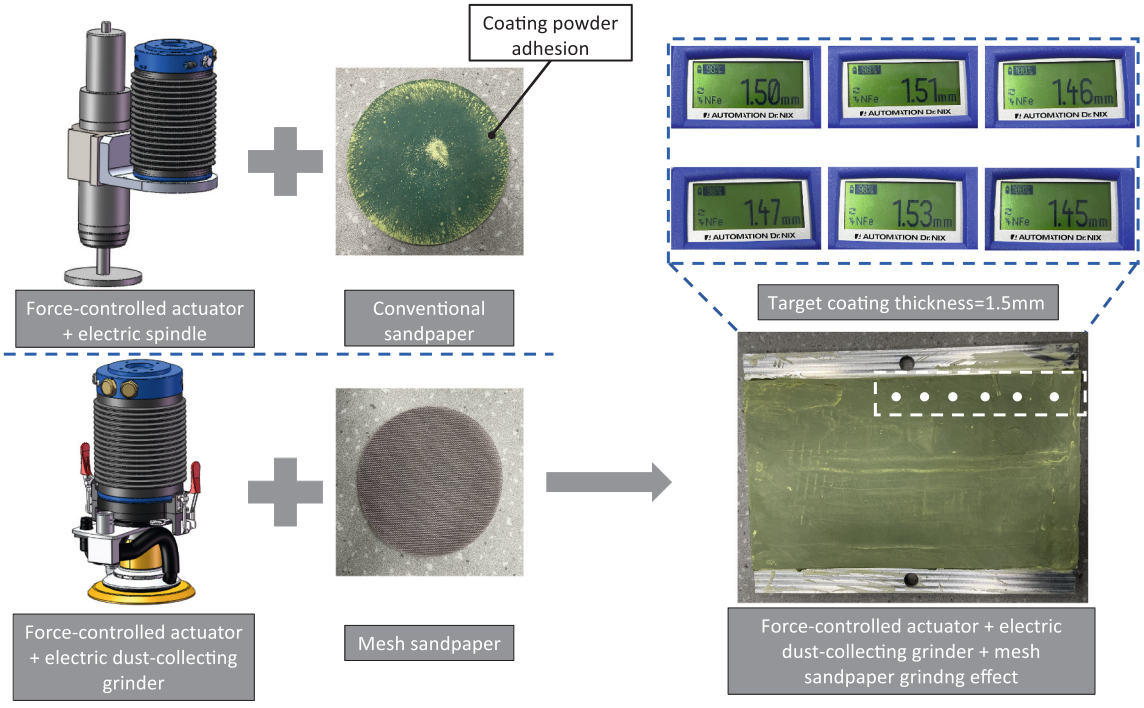

Fig. 17. Experiment on coating grinding using force-controlled actuators, electric vacuum grinders and mesh sandpaper combinations.

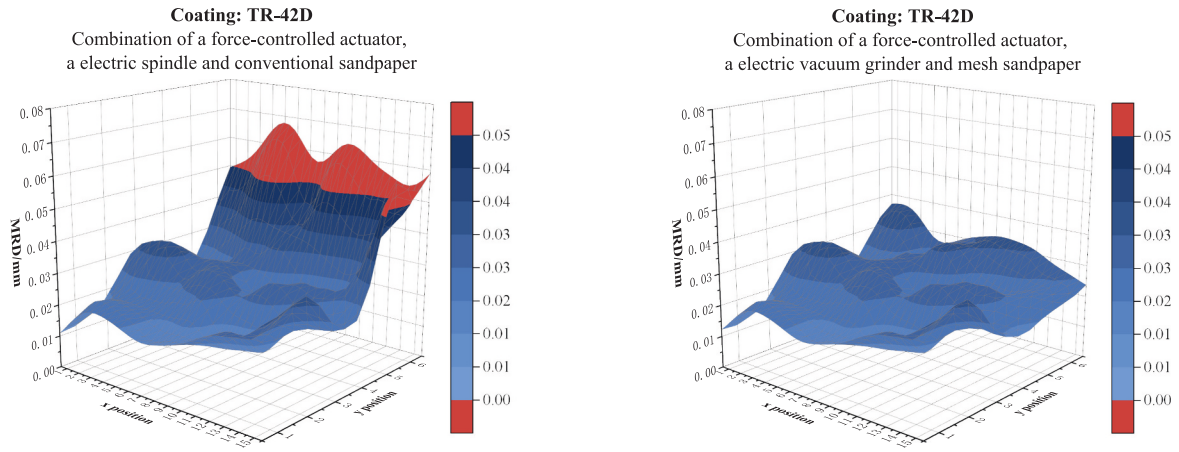

Fig. 18. Comparison of material removal accuracy between conventional sandpaper loaded by electric grinder and mesh sandpaper loaded by electric vacuum grinders.

policy, achieves a balance between exploration and exploitation. It allows for a smooth adjustment of action reward values, granting the SAC algorithm a relative advantage over traditional reinforcement learning in optimizing action selection under complex, high-dimensional conditions. As training progresses, the SAC algorithm can distinguish certain task-specific details to a very limited extent. However, it remains insufficiently sensitive to variations across tasks.

By contrast, the FLC algorithm's action rewards distribution remains relatively uniform due to its static, rule-based framework. Some data points received high reward values, this could be due to certain specific situations where the FLC rules happen to align well with the task requirements. Lacking an adaptive mechanism to differentiate action efficacy, FLC neither excessively amplifies rewards for high-precision outcomes nor over-penalizes suboptimal decisions. Consequently, its reward assignment shows limited responsiveness to dynamic variations such as tool wear, leading to reduced effectiveness in refining material removal strategies under complex, time-varying conditions.

Fig. 7 highlights the MAML-PPO$_{BE}$ model's enhanced ability for rapid adaptation. The model adjusts its policy based on the current task, even without full feedback or iterative updates during testing. By taking the current and target thicknesses as inputs, it quickly selects the most suitable polishing parameters and predicts thickness changes. In contrast to traditional reinforcement learning, the meta-reinforcement learning model can modify its policy with minimal feedback in just a few rounds, enabling efficient adaptation to new tasks or environments.

As shown in Fig. 7, five grinding tasks are designated as the test set, with an average of 22 steps required to achieve convergence in material removal deviation, resulting in an average deviation of 0.0276 mm between actual and target coating thicknesses.

Fig. 8 illustrates the adaptive decision-making models for grinding task process parameters established respectively based on MAML, PPO$_{BE}$ SAC and FLC algorithms, these models were tested under 5 distinct simulation grinding tasks. From Fig. 8, it can be observed that although the conventional MAML algorithm performs adequately with shot-few training samples. However, compared to the MAML algorithm, the MAML-PPO$_{BE}$ algorithm demonstrates significantly stronger feature extraction capabilities, resulting in a 48 % reduction in material removal deviations and a 36 % improvement in the convergence rate.

It is evident that PPO$_{BE}$ demonstrates excellent decision-making performance when encountering test tasks closely aligned with the training tasks (Simulation grinding test1). However, when the test tasks differ substantially the training tasks, the adaptive decision-making model based on the PPO$_{BE}$ algorithm shows multiple occurrences of over-grinding and high material removal deviations. Compared to the PPO$_{BE}$ algorithm, the MAML-PPO$_{BE}$ algorithm reduces material removal deviations by 47 % and a 51 % improvement in the convergence rate.

Over-grinding was observed in testing scenarios where the tasks differed significantly from those encountered during training. Despite the reward function being structured to terminate iterations when the coating thickness deviations become negative, the PPO$_{BE}$ model failed to consistently prevent such occurrences. This discrepancy arises from the inherent limitations of PPO$_{BE}$ algorithm in generalization across tasks with substantial variations. Its training heavily relies on patterns observed in the training dataset. Consequently, facing with testing scenarios significantly different from the training data, PPO$_{BE}$ struggles to extrapolate accurately due to its limited capacity to adapt to unseen environments. This results in suboptimal policy updates and the inability to prevent over-grinding effectively.

From Fig. 8, it is evident that the SAC algorithm encounters difficulties in decision-making efficiency for robotic grinding tasks. Its reliance on a stochastic policy to balance exploration and exploitation, which results in a notably slower convergence rate. It requires an average of 32 additional iterations. Furthermore, the SAC algorithm's decision accuracy is tightly coupled with large training samples. Due to its inclination toward exploration, it initially sacrifices stability in action selection, leading to high variability. Under limited training samples, the MAML-PPO$_{BE}$ algorithm exhibits superior adaptability and generalization compared to the SAC algorithm. It experiences a 56.5 % decrease in material removal deviation and frequent occurrences of over-grinding. These enhancements render the MAML-PPO$_{BE}$ algorithm highly suitable for real-time adaptability and accurate parameter control.

Furthermore, we evaluated the FLC algorithm under the same five grinding tasks to investigate its capacity for adaptive material removal control. Compared to FLC algorithm across these tasks, MAML-PPO$_{BE}$ demonstrated a 54 % average increase in material removal deviation. Notably, the material removal deviation of FLC algorithm grew proportionally in high-removal grinding tasks, such as grinding task 2 and 5. This phenomenon is primarily driven by progressive tool wear, which the static membership functions and rule-based framework of the FLC algorithm cannot compensate for effectively. Additionally, FLC required approximately 41 more steps to converge than MAML-PPO$_{BE}$, making it the slowest converging algorithm among all comparison methods. This is due to its control adjustments depend on fixed rules rather than iterative learning. These results highlight FLC's limitations in addressing tool wear and adapting to dynamic, high-accuracy grinding tasks.

Overall, this demonstrates that the MAML-PPO$_{BE}$-based adaptive decision-making model harmonizes the decision-making strengths of reinforcement learning with the adaptability of meta-learning. To enhance clarity and facilitate intuitive understanding, Table 7 provides a concise yet detailed comparison of key performance indicators for MAML-PPO$_{BE}$, MAML, PPO$_{BE}$, SAC and FLC algorithms.

4. Experiment

4.1. Experimental plan

To validate the practical effectiveness of the adaptive decision-making model for grinding process parameters based on the MAML-PPO_{BE} algorithm, an experimental platform for robotic coating grinding was established. This platform consists of an robot, a force control actuator, an electric spindle, a grinding workbench and an industrial computer. The coated test workpiece is fixed on the grinding workbench, the robot performs different grinding trajectories and movement speeds M_S. The force control actuator outputs various grinding contact forces F and the electric spindle controls different grinding rotation speeds R_S. The target coating thickness is set for each experimental workpiece, the actual coating thickness in different areas of the workpiece is measured and the sandpaper specifications are selected. These information serves as the input to the state space of the adaptive decision-making model. The MAML-PPO_{BE} algorithm runs on the industrial computer, the grinding contact force F, spindle rotation speeds R_S and robot movement speeds M_S are calculated and output by the adaptive decision-making model to the respective actuators. Ultimately, this realizes the adaptive decision-making of coating grinding process parameters on the workpiece surface. The principle of the experimental platform for robotic coating grinding process parameter adaptive decision-making is shown in Fig. 9.

4.2. Experimental results and discussion

As depicted in Fig. 10, the coating on each workpiece surface is divided into two regions, with each region assigned a target coating thickness and a selected sandpaper. The grinding sandpaper maintains a constant angle of 34 degrees relative to the surface of the workpiece. The uniform removal width is measured. Each target coating thickness area is divided into multiple equal material removal zones. Within each uniform removal area, measurement points are taken at intervals of 20 mm to determine the actual coating thickness at that position. The data, provided in Table 8, serves as the input to the grinding process parameters adaptive decision-making model based on the MAML-PPO_{BE}, MAML, PPO_{BE}, SAC and FLC algorithm.

Based on the adaptive decision-making model trained with above 50 samples, optimal grinding process parameter combinations for different position coatings on the experimental workpieces were calculated. Robotic grinding experiments were conducted on 4 different characteristic coatings. As illustrated in Fig. 11, the residual coating thickness at the corresponding positions after grinding was measured and compared with the target coating thickness H to analyze the material removal accuracy.

Data was collected for each type of coating in 90 groups, resulting in a total of 360 sets of different coatings' residual thickness data after grinding. A material removal deviations distribution plot was fitted based on this data, as illustrated in Figs. 12 to 15. The blue areas represent deviations within [0, 0.05 mm], while the red areas represents deviations outside [0, 0.05 mm]. As illustrated in Fig. 12, using process parameters determined by the MAML-PPO_{BE} algorithm, 357 instances of material removal deviations were within 0.05 mm, with an average material removal deviation of 0.025 mm. Compared to the deviations results based on the conventional MAML algorithm shown in Fig. 13, the PPO_{BE} algorithm shown in Fig. 14, the conventional SAC algorithm shown in Fig. 15 and the FLC algorithm shown in Fig. 16, the material removal deviations were reduced by 51.4 %, 68.9 %, 57.2 % and 55.2 %, respectively.

The MAML algorithm, while offering better adaptability across different grinding tasks with fewer training samples, still struggles to achieve high material removal accuracy in specific tasks. This is primarily due to its lack of reinforcement learning capabilities, which limits its effectiveness in handling complex and high-dimensional problems.

The PPO_{BE} algorithm, due to its inability to adapt across different grinding tasks, results in poor accuracy when networks and parameters learned from old grinding training samples used on new grinding test samples. It often causes over-grinding, where the residual coating thickness is less than the target coating thickness. It could only achieve relatively good material removal accuracy on coatings similar to the training samples (Coating: TR-42D).

Moreover, while the SAC algorithm is recognized for its strong reinforcement learning capabilities, it has notable limitations in its application to new grinding tasks. Specifically, it requires extensive optimization and training to perform effectively in the unfamiliar environments. This dependency on prior experience and the need for adaptation hinder its performance when applied to tasks that deviate from previously encountered scenarios. Even when faced with similar or closely related coatings, the material removal accuracy still exhibits significant fluctuations. This is primarily due to the SAC algorithm's reliance on a stochastic policy to balance exploration and exploitation, which results in slower convergence and suboptimal performance when limited training samples are available.

Finally, the FLC algorithm, leveraging expert knowledge and fuzzy inference, adjusts parameters in response to coating variations. Its material removal accuracy falls short of the MAML-PPO_{BE} algorithm, this limitation stems from the inherent constraints of rule-based membership functions in capturing highly complex and multidimensional system behaviors. Notably, the FLC algorithm often exhibits a tendency toward positive deviation, resulting in final coating thicknesses exceeding the target value. This phenomenon arises from the inflexibility of FLC's rule-based membership functions, which struggle to quickly adapt to dynamic conditions such as tool wear and variations in coating characteristics. As the grinding tool undergoes progressive degradation, the static fuzzy rules fail to adjust in a timely manner, leading to persistent deviations in the final coating thickness.

Further analysis of the residual coating thickness deviation distribution in Fig. 17, it reveals three instances of deviations (0.078 mm, 0.076 mm and 0.088 mm), where the residual coating thickness exceeds the target thickness. This type of coating adheres easily to the sandpaper surface. Additionally, the thickness of the coatings in these two areas has a significant allowance. During grinding, the robot's movement speed is relatively slow, resulting in longer grinding time. Consequently, more coating material adheres to the sandpaper surface.

As illustrated in Fig. 18, conventional sandpaper was replaced with mesh sandpaper of the same grit and abrasive. The electric spindle was also substituted with an electric grinder equipped with a dust extraction function. A robotic grinding process experiment was then conducted again on the same material coating and the residual coating height at the same positions was measured. The experimental results demonstrated that the deviation between the residual coating height and the target was less than 0.05 mm. The robotic multi-coating grinding experiments validated that the grinding process parameter adaptive decision-making model based on the MAML-PPO_{BE} algorithm, with only few-shot training samples, the model can quickly adapt to different types of grinding tasks while achieving high material removal accuracy.

5. Conclusion

This paper presents an adaptive decision-making model or grinding process parameters based on the MAML-PPO_{BE} algorithm. This model is capable of adaptively adjusting the process parameters with minimal training samples across different, coatings, grinding tool types and tool wear conditions. The main conclusions of this paper are as follows:

1) **Introduction of the PPO_{BE} Algorithm:** We proposed the PPO_{BE} algorithm, which achieves process parameter decision-making for specific grinding tasks, thereby enhancing material removal accuracy. PPO_{BE} utilizes a self-guided network trained on better experience trajectories to optimize the agent's action selection. This

approach allows for more effective learning from exemplary samples. When trained and tested on specific grinding tasks, the PPO_{BE} algorithm exhibits superior performance in terms of rewards, loss, value and material removal accuracy, compared to conventional reinforcement learning algorithms such as PPO and TRPO.

2) **Development of the MAML-PPO_{BE} adaptive decision-making model:** We proposed an adaptive decision-making model for grinding process parameters based on the MAML-PPO_{BE} algorithm. This model not only retains the material removal accuracy of the PPO_{BE} algorithm but also enhances the adaptability of the process parameters. Experimental results for robotic coating grinding indicate that, the MAML-PPO_{BE} algorithm achieves a material removal standard deviation of 0.025 mm in 16 different tasks. Compared to the process parameters derived from the MAML, PPO_{BE}, SAC and FLC algorithms, the material removal deviation is reduced by 51.4 %, 68.9 %, 57.2 % and 55.2 %, respectively.

The adaptive decision-making model proposed in this paper helps to reduce the dependence on human expertise in grinding tasks, thereby advancing the development of intelligent manufacturing. In future work, we will further explore adaptive planning of grinding trajectories and integrating it with adaptive process parameters decision-making to enable fully adaptive grinding operations.

CRediT authorship contribution statement

Jie Pan: Writing – original draft, Software, Methodology, Conceptualization. **Fan Chen:** Writing – review & editing, Supervision, Resources, Funding acquisition. **Dan Han:** Software, Investigation, Formal analysis, Data curation. **Shuai Ke:** Software, Formal analysis, Data curation. **Zhiao Wei:** Formal analysis, Data curation. **Han Ding:** Supervision, Resources.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT 3.5 in order to enhance sentence accuracy. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 52188102 and 52090054).

References

[1] Lv LS, Deng ZH, Liu T, et al. Intelligent technology in grinding process driven by data: a review. J Manuf Process 2020:581039–51. https://doi.org/10.1016/j.jmapro.2020.09.018.
[2] Li XW, Huang ZX, Ning WH. Intelligent manufacturing quality prediction model and evaluation system based on big data machine learning. Comput Electr Eng 2023;111(PA). https://doi.org/10.1016/j.compeleceng.2023.108904.
[3] Xu FJ, Xu YL, Zhang HJ, et al. Application of sensing technology in intelligent robotic arc welding: A review. J Manuf Process 2022:79854–80. https://doi.org/10.1016/j.jmapro.2022.05.029.
[4] Marc-Andrè B, Akhloufi Moulay A. Reinforcement learning for swarm robotics: an overview of applications, algorithms and simulators. Cogn Robot 2023:3226–56. https://doi.org/10.1016/j.cogr.2023.07.004.
[5] Li T, Yan YH, Yu CS, et al. A comprehensive review of robot intelligent grasping based on tactile perception. Robot Comput-Integr Manuf 2024;90:102792. https://doi.org/10.1016/j.rcim.2024.102792.
[6] Gregor K, Donka N. Production technology research – building blocks for competitiveness and solution for future challenges in aerospace component manufacturing. Procedia CIRP 2021:10162–8. https://doi.org/10.1016/J.PROCIR.2020.09.189.
[7] Soni R, Verma R, Garg KR, et al. Progress in aerospace materials and ablation resistant coatings: A focused review. Opt Laser Technol 2024;177:111160. https://doi.org/10.1016/J.OPTLASTEC.2024.111160.
[8] Yao JS. Research on key Technologies of Multi-Missile Weapon-Target Allocation. National University of Defense Technology 2020. https://doi.org/10.027052/d.cnki.gzjgu.2020.000415.
[9] Segade RM, Hernández S, Díaz J. Multi-level and multi-objective structural optimization for hypersonic vehicle design. Aerosp Sci Technol 2024:152109346. https://doi.org/10.1016/j.ast.2024.109346.
[10] Hu D, Fu QG, Dong ZJ, et al. Design of ablation resistant Zr-ta-O-C composite coating for service above 2400 °C[J]. Corros Sci 2022;180:110221. https://doi.org/10.1016/j.corsci.2022.110221.
[11] Wang ZY, Huang YF, Zhou J, et al. Effect of Fe content on the Tribological properties of Ni60 coatings applied by pulsed magnetic field assisted supersonic plasma spraying[J]. Materials & Design 2022. https://doi.org/10.1016/j.matdes.2022.111127.
[12] Zhuang KJ, Wu ZZ, Wan LY, et al. Investigation of different abrasive jet machining methods applied to milling tool coatings for post-treatment. Surf Coat Technol 2024;491(15):131156. https://doi.org/10.1016/j.surfcoat.2024.131156.
[13] Zhang HY, Li L, Zhao JB, et al. Theoretical investigation and implementation of nonlinear material removal depth strategy for robot automatic grinding aviation blade. J Manuf Process 2022:74441–55. https://doi.org/10.1016/J.JMAPRO.2021.12.028.
[14] Zhuang KJ, Zhu K, Wei XY, et al. A dual-stage wear rate model based on wear mechanisms analysis during cutting Inconel 718 with TiAlN coated tools. J Manuf Process 2024;126:24–34. https://doi.org/10.1016/j.jmapro.2024.07.089.
[15] Wei CX, He CL, Chen G, et al. Material removal mechanism and corresponding models in the grinding process: a critical review. J Manuf Process 2023:103354–92. https://doi.org/10.1016/J.JMAPRO.2023.08.045.
[16] Hiroyuki K, Takao M, Yuta W, et al. Construction of grinding wheel decision support system using random forests for difficult-to-cut material. Precision Eng 2023:84162–76. https://doi.org/10.1016/J.PRECISIONENG.2023.08.004.
[17] Zhang JS, Jiang YC, Luo H, et al. Prediction of material removal rate in chemical mechanical grinding via residual convolutional neural network. Control Eng Pract 2021:107104673. https://doi.org/10.1016/j.conengprac.2020.104673.
[18] Fu SL, Wang LP, Wang D, et al. Accurate prediction and compensation of machining error for large components with time-varying characteristics combining physical model and double deep neural networks. J Manuf Process 2023:99527–47. https://doi.org/10.1016/J.JMAPRO.2023.05.067.
[19] Zhao ML, Xue BX, Li BH, et al. Ensemble learning with support vector machines algorithm for surface roughness prediction in longitudinal vibratory ultrasound-assisted grinding. Precision Eng 2024:88382–400. https://doi.org/10.1016/j.precisioneng.2024.02.018.
[20] Kim G, Park S, Choi GJ, et al. Developing a data-driven system for grinding process parameter optimization using machine learning and metaheuristic algorithms. CIRP J Manuf Sci Technol 2024:5120–35. https://doi.org/10.1016/j.cirpj.2024.04.001.
[21] Paturi RMU, Cheruku S. Application and performance of machine learning techniques in manufacturing sector from the past two decades: a review. Mater Today Proc 2020. https://doi.org/10.1016/j.matpr.2020.07.209 [prepublish].
[22] Íñigo E, Antonio S, Dimitrios C, et al. A review on reinforcement learning for contact-rich robotic manipulation tasks. Rob Comput-Integr Manuf 2023;81. https://doi.org/10.1016/J.RCIM.2022.102517.
[23] Saleh EM, Wazery MY, Ali AA. A systematic literature review of deep learning-based text summarization: techniques, input representation, training strategies, mechanisms, datasets, evaluation, and challenges. Expert Syst Appl 2024;252(PA):124153. https://doi.org/10.1016/J.ESWA.2024.124153.
[24] Kamal K, Ram SC, Kumar MS. Application of machine learning techniques in environmentally benign surface grinding of Inconel 625. Tribol Int 2023;188. https://doi.org/10.1016/J.TRIBOINT.2023.108812.
[25] Verbeke P, Verguts T. Reinforcement learning and meta-decision-making. Curr Opin Behav Sci 2024;57:101374. https://doi.org/10.1016/J.COBEHA.2024.101374.
[26] Oluwaseyi O, Homayoun N. Reinforcement and deep reinforcement learning-based solutions for machine maintenance planning, scheduling policies, and optimization. J Manuf Syst 2023:70244–63. https://doi.org/10.1016/J.JMSY.2023.07.014.
[27] Tang Q, Liang J, Zhu GQ. A comparative review on multi-modal sensors fusion based on deep learning. Signal Process 2023;213. https://doi.org/10.1016/J.SIGPRO.2023.109165.
[28] Zhang XQ, Dang JW, Wang YP, et al. Meta-learning framework with updating information flow for enhancing inductive prediction. Knowl-Based Syst 2024;294:111720. https://doi.org/10.1016/j.knosys.2024.111720.
[29] Yu B, Feng XY, Kong Y, et al. Using meta-learning to establish a highly transferable driving speed prediction model from the visual road environment. Eng Appl Artif Intel 2024:130107727. https://doi.org/10.1016/j.engappai.2023.107727.
[30] Li XZ, Su H, Xiang L, et al. Transformer-based meta learning method for bearing fault identification under multiple small sample conditions. Mech Syst Signal Process 2024;208. https://doi.org/10.1016/j.ymssp.2023.110967.
[31] Yu CC, Zhang ZH, Li HB, et al. Meta-learning-based adversarial training for deep 3D face recognition on point clouds. Pattern Recognit 2023;134. https://doi.org/10.1016/J.PATCOG.2022.109065.

[32] Kumar AS, Gopinatha P, Sohom C. Reinforcement learning algorithms: A brief survey. Expert Syst Appl 2023;231. https://doi.org/10.1016/j.eswa.2023.120495.
[33] Zhao BX, Dong HB, Wang YJ, et al. PPO-TA: adaptive task allocation via proximal policy optimization for spatio-temporal crowdsourcing. Knowl-Based Syst 2023; 264. https://doi.org/10.1016/j.knosys.2023.110330.
[34] Mayer S, Classen T, Endisch C. Modular production control using deep reinforcement learning: proximal policy optimization. J. Intell. Manuf. 2021;32(8): 1–17. https://doi.org/10.1007/s10845-021-01778-z.
[35] Boudlal A, Khafaji A, Elabbadi J. Entropy adjustment by interpolation for exploration in Proximal Policy Optimization (PPO). Eng Appl Artif Intel 2024;133 (PE):108401. https://doi.org/10.1016/J.ENGAPPAI.2024.108401.
[36] Cheng JD, Cheng MH, Liu Y, et al. Knowledge transfer for adaptive maintenance policy optimization in engineering fleets based on meta-reinforcement learning. Reliab Eng Syst Saf 2024;247:110127. https://doi.org/10.1016/j.ress.2024.110127.
[37] Pang SC, Zhang L, Yuan YD, et al. Adaptive-MAML: few-shot metal surface defects diagnosis based on model-agnostic meta-learning. Measurement 2023:223. https://doi.org/10.1016/j.measurement.2023.113612.
[38] Cheng Y. On the theory of single-task and multitask reward-free reinforcement learning under low-rank MDPs. University of Science and Technology of China 2024. https://doi.org/10.27517/d.cnki.gzkju.2023.002082.
[39] Li GH, Li XF, Li J, et al. PTMB: an online satellite task scheduling framework based on pre-trained Markov decision process for multi-task scenario. Knowl-Based Syst 2024:284111339. https://doi.org/10.1016/j.knosys.2023.111339.