Machine Learning-Based Multi-objective Optimization Framework for Industrial Black Nickel Electroplating

Junhao Ren[1], Qiyu Kang[1], Shuo Feng[1], Yajuan Sun[2], Yong Tech Tan[2], Gaoxi Xiao[1*]

[1*]School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore, 639798, Republic of Singapore.
[2]Singapore Institute of Manufacturing Technology (SIMTech), Agency for Science, Technology and Research (A*STAR), 2 Fusionopolis Way, Singapore, 138634, Republic of Singapore.

*Corresponding author(s). E-mail(s): egxxiao@ntu.edu.sg;
Contributing authors: junhao002@e.ntu.edu.sg; kang0080@e.ntu.edu.sg; shuo004@e.ntu.edu.sg; sun_yajuan@simtech.a-star.edu.sg; tanyongt@simtech.a-star.edu.sg;

Abstract

The optimization of industrial processes, such as the industrial black nickel electroplating (IBNE) process, is challenging due to the intricate structure and dynamics involved. Machine learning (ML) methods are proposed to learn from historical data to address this issue. In this study, we propose a novel intelligent process optimization framework based on ML methods to optimize the IBNE process through a sim-to-real approach. The framework consists of a virtual IBNE environment simulator and a deep reinforcement learning-based optimization architecture. The virtual IBNE environment simulator is designed to address three objectives: lightness, uniformity and plating rate, based on a historical plating dataset. Lightness and uniformity are considered in one defect detection problem, where a sample is classified as "Pass" if it meets both criteria, and "Fail" otherwise. In addition, a reward function is formulated to evaluate the plating performance of samples, with the penalty term derived by solving a constrained polynomial optimization problem based on constraints extracted from the dataset. A deep deterministic policy gradient (DDPG) algorithm is presented to learn the optimal current density corresponding to different plating conditions, ensuring that the plating process can achieve optimal performance under specific conditions. Finally, we apply the policy learned in the virtual IBNE environment to a real-world application, and results from laboratory experiments validate the effectiveness of the proposed framework.

Keywords: Black nickel electroplating, Machine learning, Industrial modeling, Deep reinforcement learning, Process optimization

1 Introduction

Optimization is of critical importance in various fields such as industrial (Hou, Li, Hsu, Zhang, & Xu, 2022; Tang, Chai, Yu, & Zhao, 2013; Wang, Li, Zhang, & Xuan, 2024), chemical (Lim, Leow, Pham, & Tan, 2021) and electrical engineering (Yang, Liu, Ma, Yang, & Guerrero, 2023). In many cases, engineering optimization involves the design, implementation, and enhancement of automated systems that streamline production processes, with the aim of enhancing efficiency, maintaining product quality, minimizing costs, and conserving resources (Khodadadi, Soleimanian Gharehchopogh, & Mirjalili, 2022; Rao, 2019).

1.1 Industrial black nickel electroplating process

Industrial electroplating has been extensively employed to protect various substrates from corrosion and abrasion, and to fulfill decorative needs (Durney, 1984; Somasundaram et al., 2018). Black nickel electroplating is among the typical electroplating methods across various industries, including, but not limited to, aerospace and electronics.

IBNE generally includes four stages: pre-cleaning, activation, electroplating, and final cleaning (Teo, Sun, & Tan, 2022). Metal substrates like copper and steel undergo thorough cleaning adopting chemical and ultrasonic techniques, removing any dirt, oil, or oxide layers, which in turn activates the surface. To ensure a contaminant-free surface, a water break test is executed. Subsequently, the substrates are connected to a DC rectifier and immersed “live” (with the current switched on before immersion) into a pre-adjusted plating bath. Upon completion of the prescribed plating duration, the rectifier is deactivated and the sample is extracted from the plating bath. The product is then meticulously cleaned with deionized water and dried using a compressed air gun. A substrate that has undergone the entire electroplating process is called a sample.

The quality of a sample is evaluated using various metrics, depending on industrial requirements. Generally, the metrics to assess a sample in IBNE include thickness (Katirci, Aktas, & Zontul, 2021; Poroch-Seritan et al., 2011), lightness (Poroch-Seritan et al., 2011), and uniformity (discoloration) (Solovjev, Denis, Solovjeva, Inna, & Konkina, Viktoriya, 2019).

1.2 Challenges and motivations

Despite the widespread applications of IBNE, the process is riddled with complications that substantially hinder efficiency and product quality. Such issues arise chiefly from the lack of robust models capable of predicting and optimizing the highly dynamic electroplating process influenced by a plethora of factors. As a result, current IBNE techniques frequently exhibit inconsistent coating properties. The inconsistencies, spanning from thickness and uniformity to the physical appearance of the plated object, impact the overall performance and aesthetics of the sample. As the process is heavily reliant on a variety of parameters, such as current density, temperature, pH, and bath composition, even slight deviations in these parameters may lead to sensitive changes, making it a challenge to sustain consistent, high-quality output. The lack of reliable and comprehensive models that consider all these variables often leads to unsatisfactory performance of the process optimization, resulting in resource wastage and low yield, hindering the ability of effectively scaling the process to large-scale industrial applications.

In general, advanced process optimization strategies that can considerably enhance the efficiency and effectiveness of IBNE process are scarce. Traditional techniques are predominantly based on trial-and-error methods and expert experiences, which may be time-consuming and offer limited potential for process enhancement.

At this juncture, ML comes into the picture. Due to significant advances in computing and artificial intelligence techniques, ML has gained unforeseen capabilities to learn from data (experiences), discovering highly complex patterns and rules without relying on explicit mathematical models (Zhou, 2021).

Although many optimization algorithms have been developed to optimize the industrial process or engineering problems (i.e., MOAEOSC for IoT in Hosseini, Gharehchopogh, and Masdari (2023)), only a few studies have adopted ML models to investigate the IBNE process. In the work of Liu et al. (2022), a comprehensible ML random forest model was introduced to predict battery capacity and analyze coating parameters. Katirci et al. predicted the thickness and percentage of Ni in ZnNi alloy electroplating, employing various ML methods including response surface design (RSD), polynomial regression (PR), and support vector regression (SVR) (Katirci et al., 2021). Even fewer studies have been conducted on IBNE optimization. In Katirci and Danaci (2023), they utilized a non-dominated sorting genetic algorithm (NSGA-II) to improve the ML model for the IBNE process, where in the training process, image data labeled with the Mask RCNN algorithm was used as input for the ML model. In Poroch-Seritan et al. (2011), the authors utilized the RSD method and the central composite experimental design (CED) for statistical modeling and analysis of IBNE. Subsequently, they employed a genetic algorithm (GA) to determine a Pareto optimal set for multiple IBNE parameters. The contributions of these works are compared in Table 1.

There remains a deficiency in the detailed analysis of IBNE from an engineering point of view. Specifically, existing studies typically take IBNE as a chemical process, paying little attention to its optimization as an engineering process. When taking IBNE as an engineering process, it can be realized that dynamically tuning different parameters incurs different costs, some significantly more costly than others. Hence,

Table 1 Comparison of existing researches in the IBNE process field

Reference	Modeling	Optimization	Methods	Multi-Objective	Real-Time
Poroch-Seritan et al. (2011)	✓	✓	Statistical modeling, GA algorithm	✓	✗
Solovjev, Denis et al. (2019)	✓	✓	Mathematical Methods, Local variations method	✗	✗
Katirci et al. (2021)	✓	✗	RSD, PR, SVR	✗	✗
Liu et al. (2022)	✓	✗	RF	✗	✗
Katirci and Danaci (2023)	✓	✓	Mask RCNN, RF, NSGA-II	✗	✗

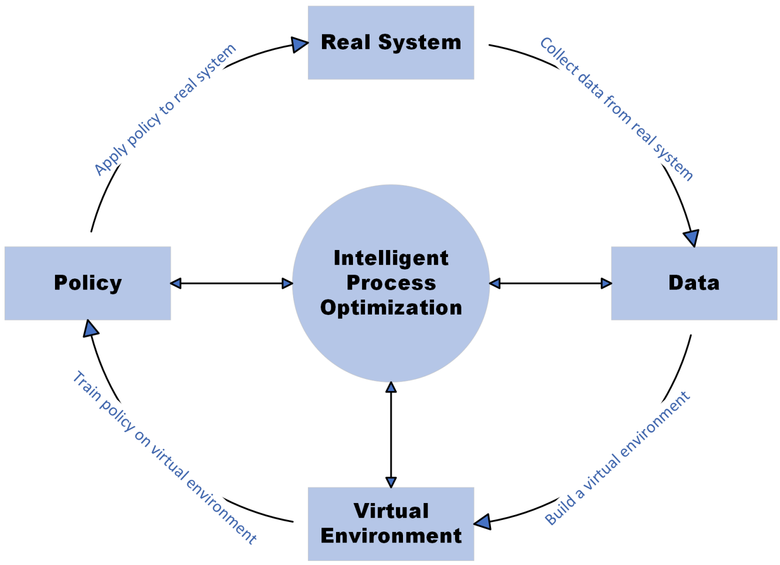

Fig. 1 The workflow of intelligent process optimization

a combination of parameters that optimize the performance of the process may not be applicable in reality. Such observations suggest that conventional optimization approaches such as GA may not be suitable for optimizing the IBNE process.

1.3 Main contributions

In this study, we propose a novel dual-stage optimization framework for the IBNE process, specifically for modeling utilizing ML techniques and optimization applying the DRL algorithm and the constructed virtual environment, respectively. This framework is shown in Fig. 1.

In the modeling stage, we examine different ML algorithms on the dataset collected using the *Beaker Cell*[1] for defect detection and plating rate prediction, respectively. Two multilayer perceptron (MLP) models are employed for building the virtual IBNE environment due to their superior performance compared to other ML algorithms. In the optimization stage, a deep reinforcement learning (DRL) algorithm, deep deterministic policy gradient (DDPG, proposed in Lillicrap et al. (2019)), is used to learn the best policy for current density control in the virtual IBNE environment to adapt to dynamic changes in plating parameters, achieving multi-objective optimization for the IBNE process. The final validation of the optimization framework is conducted in a laboratory setting and affirms the effectiveness of the proposed approach. In this work, our main contributions are as follows:

[1] *Beaker Cell* is a common wet lab set-up for electroplating process experiments in laboratories. It helps engineers analyze the effectiveness of different chemical components under various current densities in the electroplating process.

- The modeling of the virtual IBNE environment is based on few-shot learning, which introduces new challenges such as an imbalanced dataset and overfitting in the training process. In this work, the transfer learning is introduced to address the labeling imbalance in defect detection. Specifically, the defect detection model is pre-trained on a balanced dataset—the *Hull cell* dataset[2]—and then be transferred for use on the *Beaker Cell* dataset. Additionally, to mitigate overfitting, feature selection and reduction are applied to remove redundant features from the dataset.
- For the convenient deployment of the optimization algorithm, we construct a virtual IBNE environment simulator. In this simulator, we integrate multiple objectives (defect detection and plating rate) into a reward function to comprehensively evaluate plating samples based on these objectives. The reward function includes a penalty term to punish and avoid unrealistic responses from the virtual IBNE environment, which may otherwise lead to unexpected and unrealistic high rewards. The penalty term is designed based on constraints observed from the original dataset using constrained polynomial optimization.
- In the optimization stage, we formulate IBNE process as a continuous decision-making process and designate current density as the action needs to be controlled. A DDPG algorithm is trained in this scenario, learning to dynamically adjust the current density to maximize the reward corresponding to different states.

1.4 Organization

The rest of this paper is organized as follows: Section 2 provides a brief introduction to the background of related machine learning methods and formulates the problem for the IBNE process. Section 3 presents the general workflow of the optimization framework, describes the experimental environment, outlines the experimental steps, and details the data collection process for the IBNE process. Section 4 elaborates on the IBNE multi-metric modeling process and performs an analysis on the performance of two proposed ML models. Section 5 establishes a DRL environment using the models discussed in Section 2 and presents an optimization framework for the IBNE process. Section 6 presents real-world experimental results for assessing the effectiveness of the proposed optimization framework. Section 7 concludes this work, states its limitations and potential extensions for future researches.

2 Background

This section provides a brief background of several ML methods and the problem of modeling the IBNE process.

2.1 MLP and DRL

In this paper, we employ supervised learning, comprising both an MLP classifier and an MLP regressor, as well as a deep reinforcement learning algorithm to model and optimize the IBNE process.

[2] *Hull cell* is a common test device in the industry for evaluating electroplating results across a range of current density conditions. It is designed to allow users to observe and evaluate the plating process at different current densities to determine the optimal concentrations of various chemical components.

2.1.1 Multilayer perceptron (MLP)

MLP is a class of feed-forward artificial neural networks. It consists of at least three layers of nodes: an input layer, one or more hidden layers, and an output layer. Each node in a layer is connected with all the nodes in the previous layer. These connections are associated with weights that are adjusted during the learning process. The MLP training procedure needs to iteratively execute two parts: forward propagation and backpropagation. In forward propagation, when a neuron z_j receives an input $(x_1, x_2, \ldots, x_n)$ from the previous layer, it computes the output signal z_j and sends it to the next layer, where $z_j = f(\sum_{i=1}^{n} w_{ji}x_i) = f(W_j^T x)$ and $f(\cdot)$ is a nonlinear activate function. Execute this layer by layer until obtaining the y, the output of the MLP; In backpropagation, we compute the local gradient of the network according to the chosen loss function and adjust the synaptic weights of the network. This training continues until the termination condition is satisfied. The trained network is evaluated on the test dataset using different metrics based on the type of tasks.

2.1.2 Deep reinforcement learning (DRL)

Reinforcement Learning (RL) is an area of machine learning in which an agent learns to make decisions by taking actions in an environment to maximize a reward signal.

Deep Deterministic Policy Gradient (DDPG) is a model-free, off-policy, actor-critic algorithm that is based on the deterministic policy gradient (Lillicrap et al., 2019). It is an extension of Deep Q-learning (see Mnih et al. (2013)) to continuous action spaces, making it applicable to a wider range of problems. The aim of DDPG is to learn a deterministic policy $\pi(s)$ by the actor instead of learning the action-value function $Q(s, a)$. The DDPG algorithm uses four neural networks: a Q-network, a deterministic policy network, and two target networks.

In addition to the applicability to a continuous action space, the DDPG algorithm also enables a more efficient learning ability and easy deployment to high-dimensional state space. Meanwhile, the output of DDPG is more reliable and explainable for the industrial optimal problem, since it learns a deterministic policy and outputs a deterministic action for a given state. The above advantages led us to adopt the DDPG algorithm in this work.

2.2 Problem statement

Industrial electroplating, a critical operation in various manufacturing sectors, is an inherently complex and nonlinear process. Electroplating operation is a function of multiple parameters, such as current density, bath composition, temperature, and plating time, among others. Each of these parameters significantly influences the characteristics and quality of the coated layer.

The nonlinearity of IBNE process arises from the sophisticated chemical reactions, which are related to the sophisticated interactions among experiment parameters. For instance, the rate of deposition, which is a function of the current density, is not linear. Generally, the higher the current density, the higher the deposition rate. However, the deposition rate is also affected by the bath's chemical composition and temperature. For example, the temperature can influence the bath's ion conductivity, and

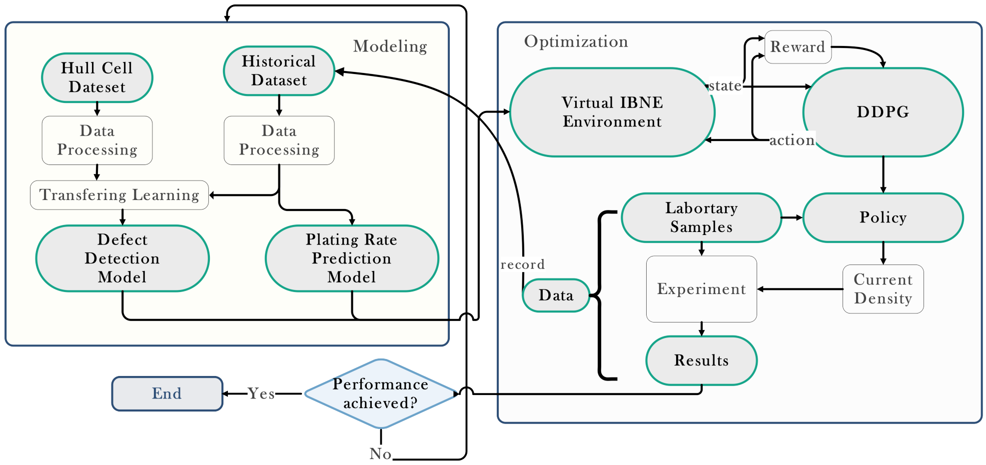

Fig. 2 The flowchart of the proposed optimization framework for IBNE process

subsequently the current distribution, resulting in a nonlinear relationship between the input variables and the plating quality. The process is further complicated by the electrochemical reactions involved in electroplating. These reactions exhibit complex nonlinear dynamics under different operating conditions.

Once the modeling phase is completed, we proceed to optimization based on these ML models. As is known, DRL optimization algorithms, such as DDPG, are heavily based on the reliability and stability of the interacting environment. However, inherent, unaccountable, and inevitable inaccuracies may occur in ML models due to sample errors. These inaccuracies may severely affect the performance of the optimization algorithms and the optimality of results due to excessive exploration in the unstable state space of the ML models. To mitigate the optimization errors caused by these inaccuracies, a response analysis is needed on these ML models. By examining the response surfaces of the models and identifying the reliable areas within the state space, we can design an appropriate reward function for an environment integrated with two ML models. This would ensure that the optimization algorithm receives a positive reward when the output is reliable; and a negative reward otherwise.

3 Optimization framework and experiment platform

In this section, we will introduce the workflow of the proposed framework, as well as the experiment platform. We shall also present the hardware/software setup for algorithm deployments.

3.1 The workflow of the optimization framework

The workflow of the proposed optimization framework is shown in Fig. 2. In the proposed framework, it consists of the modeling stage and the optimization stage. In the modeling stage, we will introduce the data processing and modeling for defect

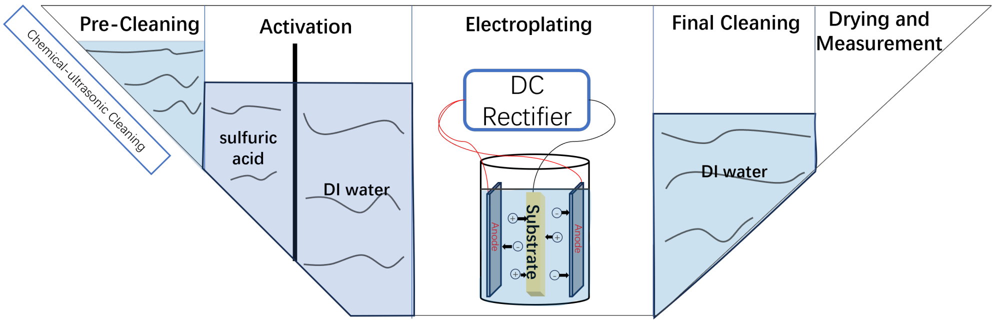

Fig. 3 The procedure of IBNE

Table 2 Parameters in the IBNE process

Parameter	**Explanation**
2 faces surface area	The surface area of substrate in contact with the solution
Initial Mass at RTP	The mass of the substrate before electroplating
Initial pH	The initial pH value before electroplating
Current	The flow of electrical charge carriers per unit time
Current density	The flow of electrical charge carriers per unit time per unit cross-sectional area
Time	The electroplating time
Temperature	The temperature of the solution
Agitation	Stirring of the plating bath to maintain uniform solution composition during electroplating
Accumulated ampere hour	A metric used to assess the age of the bath solution
C_{Zn2+}	The concentration of Zn2+ in the bath solution
C_{SCN-}	The concentration of SCN- in the bath solution

detection and plating rate prediction, respectively. As to the optimization stage, it contains constructing a virtual IBNE environment based on the defect detection model and the plating rate prediction model, optimization algorithm training in the virtual IBNE environment and the experiment validation of the optimization algorithm.

3.2 IBNE experiment platform

In this work, all historical data and the data of the experiment validation are collected based the real electroplating experiments conducted in the same laboratory environment. The main procedure of IBNE is illustrated in Fig. 3. The detailed raw experimental data are recorded in this process. All the recorded parameters are listed and explained in Table 2.

3.2.1 Electroplating preparation and composition

Before starting electroplating, we prepare the cleaned sample and adjusted solution in the bath. Adjustments to the bath solution may change its pH value and concentration of multiple ions. In our experiments, a 1-liter glass beaker was used as a plating bath

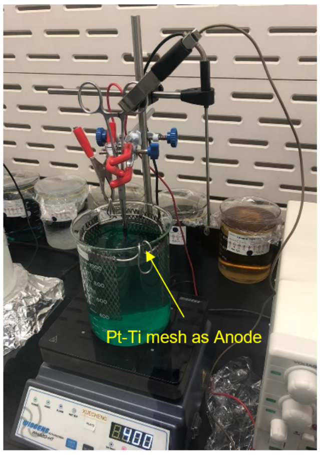

Fig. 4 The setting of the plating bath container

Table 3 Black nickel plating solution components

Component	Amount (g/L)
Ammonium Nickel Sulphate Hexahydrate ($(NH_4)_2SO_4.NiSO_4.6H_2O$)	60
Zinc Sulphate Heptahydrate ($ZnSO_4.7H_2O$)	Depends on the experiment
Sodium Thiocyanate (NaSCN)	15
pH Adjustment (Ammonia solution to increase pH, H_2SO_4 to decrease pH)	Depends on the experiment

container and a Pt-Ti mesh served as the anode. The cathode is the substrate. (See Fig. 4)

In the plating solution preparation process, to reach the specific ion concentration, we calculated the molar mass and the weighed quantities of chemicals according to the volume of the solution. The chemicals are added to the solution and stirred well. The initial pH is then adjusted by the addition of acid or base, and measured using the pH meter (Metrohm 826 pH mobile). The chemicals required for this process are listed in Table 3.

In the substrate preparation process, the substrate is some copper panels of size $100mm \times 65mm \times 3mm$. Before plating, each copper panel was weighed and then masked with tape to control the surface area for plating. The plating region was around 0.4 to 0.5 dm^2 in this work. The copper panel was then cleaned in a hot alkaline solution at 80°C temperature for 10 minutes, before being rinsed in DI water. The copper panel was then activated by immersion in 10% sulfuric acid and then be thoroughly rinsed with DI water. This activation step is important for removing the copper oxide on the sample surface to ensure good-quality plating in subsequent steps. Water-break test is done to ensure the surface is contaminant-free.

3.2.2 Electroplating process and plating characterization

After these preparation processes, the substrate is connected to a DC rectifier (Extech Quad Output DC Power Supply) and immersed into the plating bath with turned-on current for desired duration (20 min). After plating, the rectifier is turned off, and the sample is removed from the plating bath, thoroughly cleaned with DI water, and dried with an air gun. The plated sample is then weighed again to obtain the mass of the plating deposit. The pH of the plating solution after plating is also recorded.

In this work, we characterize their appearance and their deposited plating rate. Appearance-wise, we looked out for presence of coating defects, followed by measuring the color of the coating characterized by the lab color values using an X-rite Ci60 spectrophotometer. Plating rate-wise, we calculate it based on the plating time and the mass difference of the sample before and after plating.

After electroplating, a sample can fall into one of four categories: passing both the lightness and uniformity tests, passing only the lightness test, passing only the uniformity test, or failing both tests. To simplify the modeling and optimization process, we classify samples into two categories: "Pass" if they pass both the lightness and uniformity tests, and "Fail" otherwise. Additionally, another objective is to predict a key metric in the IBNE process: the plating rate. All these results are recorded together with the corresponding parameters aforementioned in the Excel documents. Thus, the problem involves three metrics represented by two models: a classification model for defect detection (Pass/Fail) outcomes and a regression model for plating rate prediction.

3.3 Setup for algorithm deployments

We train and evaluate all modeling and optimization algorithms on an AMD Ryzen Threadripper PRO 3955WX 16-Core 3.9 GHz CPU, with NVIDIA GeForce RTX 3090 GPUs.

For software, we primarily use NumPy 1.21.5 (Harris et al., 2020), PyTorch 2.0.0 (Paszke et al., 2019), Optuna 3.1.1 (Akiba, Sano, Yanase, Ohta, & Koyama, 2019), and Scikit-learn 1.2.2 (Pedregosa et al., 2011) for implementation of various algorithms.

4 Modeling for IBNE process based on MLP algorithm

In this section, we model the IBNE process based on two metrics that are most relevant to this process in reality: defects and plating rate. All parameters are called "features" for convenience in this section. All features are normalized and then preprocessed by the feature importance analysis and the feature reduction. The processed data are fed into two MLP models to provide prediction results.

4.1 Data processing based on domain-knowledge

Data are sampled from the laboratory environment and subsequently collected and stored in a dataset. At the preprocessing step, we need to consolidate the collected data

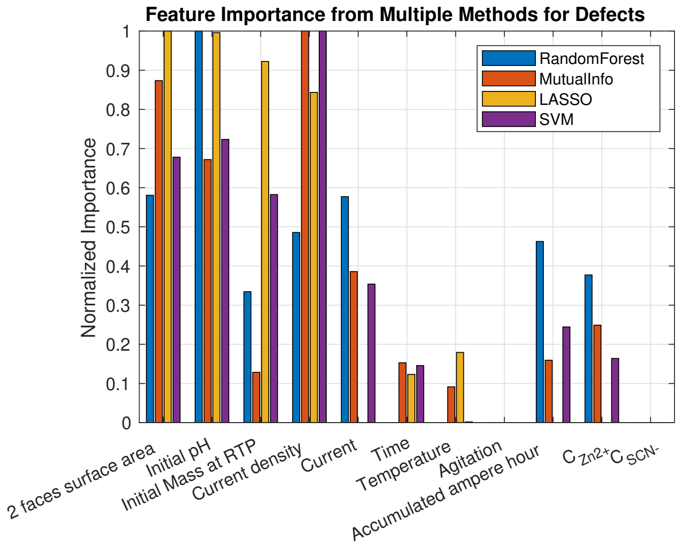

Fig. 5 Feature importance for defects

Table 4 Most importance features after feature selection

	Defects	Plating Rate
Features	Initial pH at RTP Current density C_{Zn2+}	Initial pH at RTP Current density C_{Zn2+}

and eliminate any missing values that could be associated with operational failures and errors. The data processing workflow encompasses two major steps: feature importance analysis (alternatively, feature reduction) and subsequent data processing.

To analyze feature importance for defect detection, we employed multiple statistical and machine learning methods, including Random Forest feature importance, Mutual Information scores, LASSO regression coefficients, and linear Support Vector Machine (SVM) weights. For plating rate prediction, feature importance was evaluated by calculating the Pearson correlation coefficients, and the results were visualized using a correlation heatmap. The feature importance for defects and plating rate is presented in Figs. 5 and 6, respectively.

The most significant features used to model the IBNE process are determined by a combination of statistical methodology and domain knowledge. The results are presented in Table 4. It is worth mentioning that the feature selection is partially based on domain knowledge, rather than solely by using statistical methods. For example, current density is the ratio of current to surface area. Consequently, removing relatively insignificant features like current and surface area can significantly reduce information redundancy in the feature space, thereby enhancing model performance. Nevertheless, there are still some shortages in the data set for feature space after reduction:

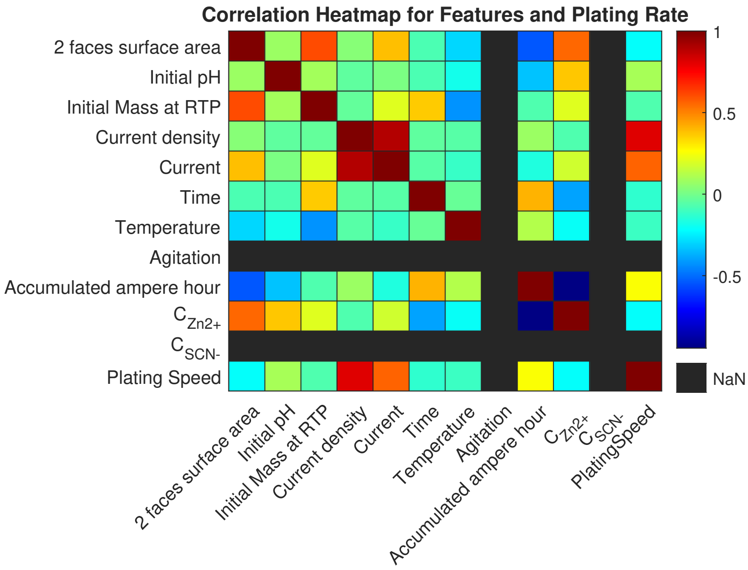

Fig. 6 The correlation heatmap of feature importance for the plating rate

- Our dataset shows a high degree of sparsity, with a majority of the feature values being identical or displaying a lack of sample diversity. The sparsity level of our data significantly influences our modeling choices. However, due to the time and cost involved in collecting diverse data, it is challenging to gather a dataset with large varieties in a short term (say, within a few months).
- The class distribution of the defects in our dataset is highly imbalanced, with the minority class (samples with defects) representing only around 25% of the observations. This class imbalance can pose challenges for standard machine learning algorithms, as these algorithms may bias towards the majority class and thus fail to adequately learn the distinguishing characteristics of the minority class.

These statistical characteristic of the dataset are shown in Fig. 7. For the sample sparsity problem, feature reduction, to some extent, can relieve (though not totally solve) the negative influences caused by sparsity. Such an approach, however, does pose an extra burden on model performance, which requests an extra solution, as will be discussed in detail later. For tackling the imbalance problem, many methods have been developed. These methods typically can be classified into resampling methods and synthetic data generation methods (see He, Bai, Garcia, and Li (2008)). However, all of them are still subject to the limitations imposed by the original dataset, which may cause potential overfitting or loss of information. In a small dataset, such as *Beaker Cell* dataset, the situation may be worse. To tackle the problem, in this work, we adopt the transfer learning method to introduce an external dataset from a similar task, which potentially provides richer and more diverse data and helps mitigate overfitting and information loss.

Remark 1. *It is noted that, to address severe imbalance in target dataset, the transferring dataset should be sufficiently large and diverse to provide a solid learning*

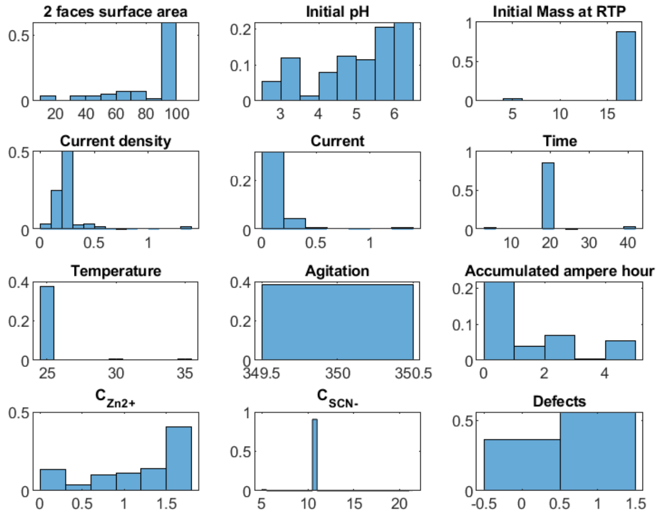

Fig. 7 The statistical distribution of the dataset for sparsity and imbalance problems

foundation and similar enough to the target domain to ensure effective knowledge transfer. Meanwhile, considering the domain difference is subtle, we transfer the entire pre-trained model based on transferring dataset to target dataset.

4.2 Modeling for IBNE process using MLP models

We adopt two different MLP models, to address the tasks related to defects classification and plating rate regression, respectively. Such an approach stems from the MLP's capability to model intricate, non-linear relationships inherent in the data. The specific process is referred to the modeling part in Fig. 2.

4.2.1 MLP for defect classification

Defect detection is approached as a binary classification task. Our objective is to classify the samples based on the following input features: Initial pH, Current density, C_{Zn2+} (see Table 4) .

One challenge we faced was the limited size of the dataset, which could potentially impact the performance of the model. To counteract this, we pre-train an MLP using the *Hull cell* dataset. Specifically, the MLP classifier is designed with two hidden layers. The ReLU (Rectified Linear Unit) function is employed as the activation for these hidden layers, while the output layer utilizes a Sigmoid activation, apt for binary classification problems. Our training employs binary cross-entropy loss (BCELoss) as the loss function for binary classification.

For hyperparameter optimization, we use the Optuna optimization library, targeting the optimal configurations for parameters such as the number and size of hidden layers, the learning rate, and the maximum iterations. The effectiveness of

Table 5 The definitions of performance metrics for ML

Metric	Expression
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
F1-Score	$2 \times \frac{\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}$
Recall	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
MSE	$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
R^2	$1 - \frac{\mathrm{SS_{res}}}{\mathrm{SS_{tot}}}$
SS_{res}	Sum of Squares of the residuals
SS_{tot}	Total Sum of Squares

each configuration is gauged using stratified 5-fold cross-validation. After optimization, the hidden layers are configured with 120 neurons. Adam optimizer is used for weight updates during training. Recognizing the potential pitfalls of relying solely on accuracy, especially with imbalanced datasets, we also leverage the F1 score as an evaluation metric. At the end of the training, this model is transferred to the *Beaker Cell* dataset for further learning in a similar way.

4.2.2 MLP for plating rate prediction

For predicting the plating rate, the model infuses four input features: initial pH, current density, surface area, and concentration of Zn^{2+}. Our MLP regressor is structured with two hidden layers, each of which comprising 120 neurons as determined through hyperparameter tuning. Like that in the classification model, the ReLU function serves as the activation for hidden layers. The output layer, designed for continuous value predictions, utilizes a linear activation. Training is guided by the Mean Squared Error (MSE) loss, given its effectiveness in capturing the average squared discrepancies between the predicted and ground truth values. For the evaluation of the model on validation and test data sets, we use the R^2 score, due to its interpretability in regression contexts.

4.3 Performance analysis

The two models have undergone evaluation and comparison against various ML algorithms using different performance metrics. Definitions of these performance metrics can be found in Table 5. Using these metrics, we assess our MLP models and compare them with other ML algorithms.

The performance of the ML model for defect classification is shown in Table 6. From the table, it is noted that, unlike other ML models where there is a gap between accuracy and F1-score, the MLP model maintains the highest scores across accuracy, F1-score and recall, suggesting that it achieves a better performance than others. Hence, MLP appears to be the most suitable model for defect detection on the dataset.

Table 6 The performance of ML classifiers for defect detection

Model Name	Accuracy	F1-Score	Recall
C-Support Vector Classification	0.772	0.683	0.772
K-nearest Neighbors Classifier	0.789	0.777	0.789
Random Forest Classifier	0.811	0.795	0.811
MLP	0.875	0.9	0.865
SGD Classifier	0.783	0.688	0.783
Gaussian Naive	0.750	0.727	0.750

Table 7 The performance of ML regressor for plating rate prediction

Model Name	MSE	RMSE	R^2
C- Support Vector Machine	3.669	1.915	0.508
K-nearest Neighbors Regressor	3.011	1.735	0.597
Random Forest Regressor	1.681	1.296	0.775
MLP	1.211	1.101	0.838
Linear Regression	1.639	1.28	0.780
XGboost	1.421	1.192	0.810
AdaBoost	1.487	1.219	0.801

The performance of the ML model for the plating rate is given in Table 7. In predicting the plating rate, once again MLP stands out as the top-performing model based on evaluation metrics. Specifically, it yields the lowest prediction errors and accounts for the highest R^2 score, indicating its strong ability in effectively capturing the relationship between the inputs and the target variables. In summary, it is evident that the MLP model is suitable for predicting the plating rate in the IBNE process.

In this section, we develop two MLP models: one for defect classification and the other for predicting the plating rate, both tailored for the IBNE process. It is demonstrated that the proposed MLP models surpass the performance of other ML models. In the following discussions, we shall employ these two MLP models to enhance the efficiency of the IBNE process.

5 Optimization framework for the IBNE process

In this section, an optimization framework based on the DRL algorithm is introduced to optimize the IBNE process. Before delving into the optimization framework, we first establish an RL environment for the IBNE process, drawing on the two MLP models detailed in Section III.

5.1 Virtual IBNE environment

An RL environment serves as the foundation for facilitating the interactions and learning of RL algorithms. Three core elements – *State*, *Action*, and *Reward* – make up its architecture. As RL algorithms cannot interact directly with pre-built models for defect detection and plating rate prediction, it is crucial to construct a specialized RL environment. This design is illustrated in Fig. 8.

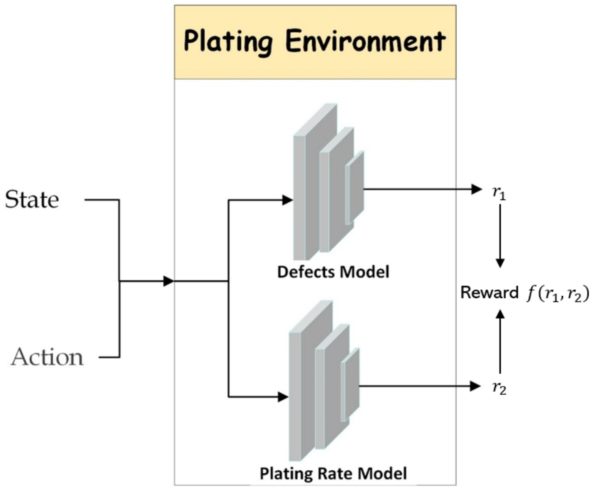

Fig. 8 The RL environment for the IBNE process

5.1.1 State and action spaces

In the context of the IBNE process, the feature space is divided into two critical segments: the state space and the action space. Our feature space includes four features: initial pH at RTP, current density, surface area, and C_{Zn2+} .

To facilitate the learning of RL algorithms, we designate *current density* as the action to be taken in the environment. The remaining features are defined as observable states. This design choice is influenced by considerations of both convenience and cost-effectiveness.

5.1.2 Designing the reward function

The reward, usually a scalar value, serves as an immediate feedback mechanism for the actions performed by the RL algorithms. Designing an appropriate reward function is not straightforward, and it can significantly impact the performance of the learning algorithm.

For our application, we desire the DRL algorithm to select an action that ensures not only defect-free plating but also a high rate of operation. Ideally, our reward function $f(r_1, r_2)$ is constructed to yield a positive value when $r_1 = 0$ (indicating defects-free). In this context ($r_1 = 0$), the faster the plating rate r_2, the higher the reward value.

Due to limitations such as dataset size and the inherent flaws such as sample sparsity, there exist regions of instability in MLP-based predictive models for the plating process. These unstable regions are manifested notably on the model response surfaces. Fig. 9 shows that defect-free samples are mainly clustered in regions with low current densities, low concentration of Zn^{2+} and high initial pH. In Fig. 10, it is demonstrated that there exists a positive correlation between plating rate and current density. Furthermore, Fig. 11 indicates that high plating rates consistently coincide with defects.

Remark 2. *From a chemical engineering point of view, the response surfaces of the defect detection and plating rate prediction models in Figs. 9 and 10 are interpretable. In the electroplating process, the workpiece, a copper plate in this study, is the cathode. Nickel and zinc ions are reduced at the cathode according to the reactions* $Ni^{2+} + 2e^- \to Ni$ *and* $Zn^{2+} + 2e^- \to Zn$*. However, when the concentration of* Ni^{2+} *is low or the current density is high, the deposition of* Zn^{2+} *and* Ni^{2+} *may not occur uniformly across the substrate surface at the right ratio, leading to coating defects such as 'rainbow' or 'uneven' appearances. Additionally, Faraday's law of electrolysis (Ehl & Ihde, 1954) states that the amount of electricity flowing through an electrolyte solution is directly proportional to the mass of the precipitated substance. An increase in current density elevates the amount of electricity traversing the electrolyte solution per unit of time, thereby accelerating the rate of metal ion reduction and deposition.*

As discussed above, to ensure prediction reliability, an optimal reward function should maximize the plating rate within the credible interval $(0, \alpha)$ and decrease sharply, or even turn negative, for rates above α. According to Fig. 11, we can find that the plating is always failed with defects for plating rate over 6. Hence, we set $\alpha = 6$ hereafter. The preliminary reward function, denoted as $f(r_1, r_2)$, is formulated as:

$$f(r_1, r_2) = (1 - r_1) \times (\beta + P_{r_2}), \tag{1}$$

where β is the reward given when the plating is successful and P_{r_2} is a penalty function depending on the plating rate. In fact, the search for the penalty function can be formulated as a constrained nonlinear parameter optimization problem (say, polynomial parameter optimization problem) as follows.

$$J_P(x) = \sum_{i=1}^{n} \left(P_{r_{2,i}}(x) - P_i\right)^2 \tag{2}$$

$$\text{s.t.} \quad dP_{r_2}(x)/dr_2 \geq 0, \quad \forall r_2 \in (0, \alpha] \tag{3}$$

$$dP_{r_2}(x)/dr_2 \leq 0, \quad \forall r_2 \in (\alpha, +\infty) \tag{4}$$

where $(r_{2,i}, P_i)$ is predefined coordinate of penalty function. We present an algorithm to solve the parameter optimization problem based on sequential least squares programming (SLSQP, see Fracas, Camarda, and Zondervan (2023)).

In this work, we choose the set of coordinates as $\{(r_{2,i}, P_i)\}_n$: $\{(0, -5), (0.2, 0), (3.5, 5), (6, 10), (8, 0), (10, -10), (14, -50)\}$ to guarantee that the penalty function satisfies our demands and set degree be equal to 6. Note that the setting of parameters, such as the set of coordinates and degree, is not unique and can be adjusted according to requirements. With $\beta = 20$, we derive the final reward function P_{r_2} by Algorithm 1 and substitute it into (1) to obtain the final reward function:

$$\begin{aligned} f(r_1, r_2) = &(1 - r_1)(20 - 0.0015r_2^6 + 0.0595r_2^5 \\ &- 0.8783r_2^4 + 5.903r_2^3 - 18.3941r_2^2 \\ &+ 26.4115r_2 - 9.503). \end{aligned} \tag{5}$$

A complete plating environment for RL algorithms is therefore constructed. Recall

Algorithm 1 Constrained polynomial optimization of penalty function

Input: The set of coordinates $\{(r_{2,i}, P_i)\}_n$, degree of polynomial, initial coefficients guess x, objective function $J_P(x)$, inequality constraints $g(x)$, tolerance, and max_iterations.
Output: Optimal coefficients x^*.
Initialize: Iteration $= 0$, converged $=$ False.
while not converged and Iteration $<$ max_iterations **do**
  Construct Lagrangian $L(x, \tau) = J_P(x) - \tau g(x)$.
  Calculate gradient of $\nabla J_P(x)$, $\nabla^2_{xx} L(x, \tau)$, $\nabla g(x)$.
  Formulate QP subproblem:
    $\min_d J_P(x) + \nabla J_P(x)^T d + \frac{1}{2} d^T \nabla^2_{xx} L(x, \tau) d$
    s. t.: $g(x) + \nabla g(x) d = 0$.
  Solve QP to get d.
  Update $[x^T_{\text{new}}, \tau^T_{\text{new}}]^T = [x^T, \tau^T]^T + d$.
  if $||x_{\text{new}} - x|| <$ tolerance **then**
    Set converged $=$ True.
  end if
  Update $x = x_{\text{new}}$, Increment Iteration.
end while
Return Optimal coefficients $x^* = x_{\text{new}}$.

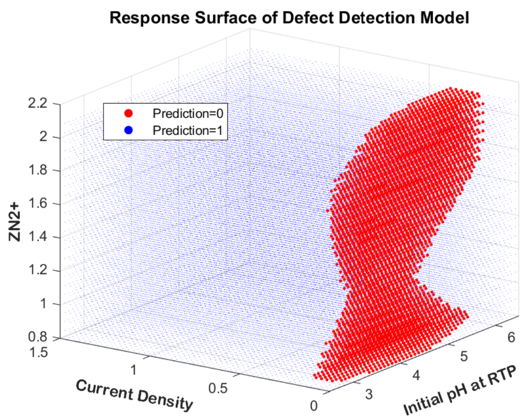

Fig. 9 Response surface of the defect detection model

that the *State* of the plating environment includes initial pH, surface area, and C_{Zn2+}; the *Action* of the plating environment is current density, and the reward function is represented as $f(r_1, r_2)$, as defined in Eq. (5). In the following, a DDPG algorithm is developed to optimize the electroplating process by interacting with the plating environment.

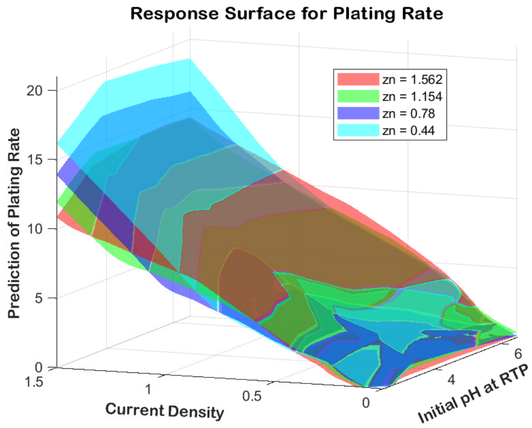

Fig. 10 Response surface of the plating rate prediction model

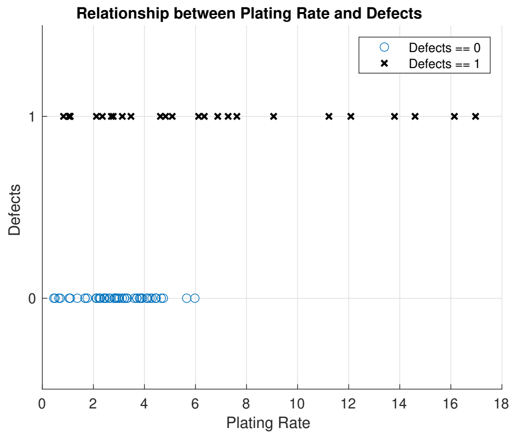

Fig. 11 The relationship between defects and plating rate in the dataset

5.2 The optimization framework based on DDPG

For convenience in the following discussion, we denote the *States* as s, the *Action* as a, and the *Reward Function* as f. To proceed the optimization of the electroplating process, we propose a DDPG-based optimization framework for the plating environment (OFPE-DDPG), shown in Fig. 12. In this work, the average reward per N times the electroplating experiments is taken as the objective to be maximized. Each electroplating experiment is treated as being independent of others. That means, all states are randomly initialized at the beginning of each electroplating experiment (discount factor should be large). Hence, the objective function of OFPE-DDPG is as follows.

$$\max J = \frac{1}{N}\sum_{k=1}^{N} f_k. \tag{6}$$

Specifically, for the critic, its objective function is to minimize the MSE loss function in terms of

$$L = \|Q(s, a) - (f + \gamma \cdot Q'(s', \pi'(s')))\|^2, \tag{7}$$

where $Q(s, a)$ is the estimation of s and a by critic, $\pi'(s')$ is estimation of s' generated by the actor. For the actor, its updating is based on the sampled policy gradient.

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a)|_{(s,a=\pi(s))} \nabla_{\theta^\pi} \pi(s)|_s \tag{8}$$

For the DDPG algorithm, we adopt the following parameter values: batch_size $= N$, hidden_layers $= 3$, number of neurons in each layer $= 32$, actor learning rate $= 1e^{-3}$, critic learning rate $= 1e^{-3}$, discount factor $\gamma = 0.999$, activation function of each hidden layer = Relu, soft update rate $\tau = 0.01$. The pseudo-code of OFPE-DDPG is given in Algorithm 2.

Algorithm 2 The training procedure of OFPE-DDPG

Initialize:
Environment: PlatingEnv
Parameters:

- buffer_size: The size of experience replay buffer
- batch_size: The number of samples in a batch
- max_iter: The maximum number of iterations
- target_reward: The target average reward

Agents: Agent1, Agent2, ..., Agent5

for each Agent in Agents **do**
    Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\pi(s|\theta^\pi)$ with weights θ^Q and θ^π.
    Initialize experience replay buffer R.
    Initialize target network Q_{target} and π_{target} with weights $\theta^{Q_{\text{target}}} \leftarrow \theta^Q$ and $\theta^{\pi_{\text{target}}} \leftarrow \theta^\pi$.
    for iter = 1 to max_iter **do**
        Select action $a = \pi(s|\theta^\pi) + \mathcal{N}$ according to the current policy and exploration noise.
        Execute action a and observe reward f and new state s'.
        Store transition (s, a, r, s') in R.
        Randomly sample a batch of N transitions from R.
        Update critic by minimizing the loss function L.
        Update actor policy using the gradient.
        Update the target networks.
    end for
end for

Algorithm 3 PlatingEnv: A virtual IBNE environment for RL

1: **Input:** Action a: Current density predicted by the agent.
2: **Output:** Reward $f(r_1, r_2)$.
3: **Initialize** the state $s = (\text{Initial pH at RTP}, \text{C}_{\text{Zn2+}})$.
4: Take the action a corresponding to the current state s.
5: Compute and return the reward $f(r_1, r_2)$.

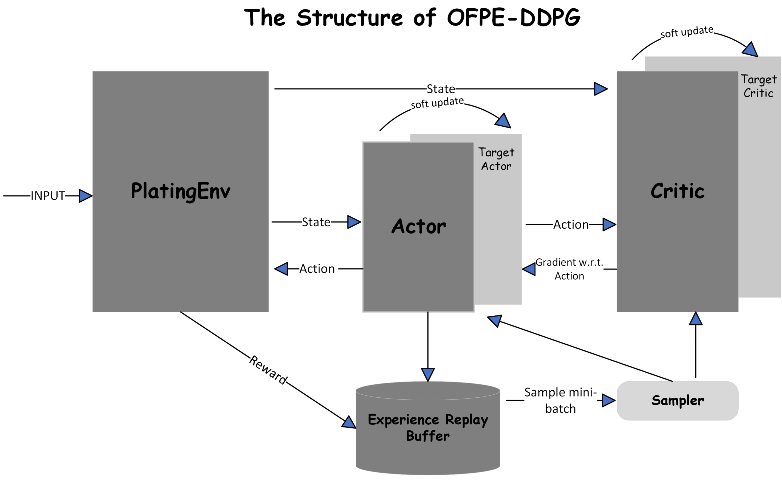

Fig. 12 The structure of OFPE-DDPG

It is noted that the soft update is adopted in this work to avoid and mitigate issues related to non-stationary target function. Mathematically, the soft update is expressed as

$$\theta_{\text{target}} = \tau \cdot \theta + (1 - \tau) \cdot \theta_{\text{target}}, \tag{9}$$

where θ and θ_{target} are the parameters of the primary and target networks respectively, and τ is the soft update rate. Compared to the hard update that replaces the weights of the target network with those of the primary network, soft update provides enhanced stability in the training process because it allows the weights of the target network to shift to the weights of the primary network more smoothly.

5.3 Performance of OFPE-DDPG

The training curve of average reward per N trials of OFPE-DDPG over the episodes is given in the Fig. 13. The results show that 5 agents of DDPG achieve good performance in convergence rate and stability. According to the designed reward function (5), it is known that the ideal average reward should be converged at around 30. However, the final average reward is not high enough. This is because we initialize the environment and randomly generate the *States* before each of the N trials. From Fig. 9, it is observed that most parameter combinations can only generate samples with defects. Namely, such initializing operation will certainly cause *States* a high

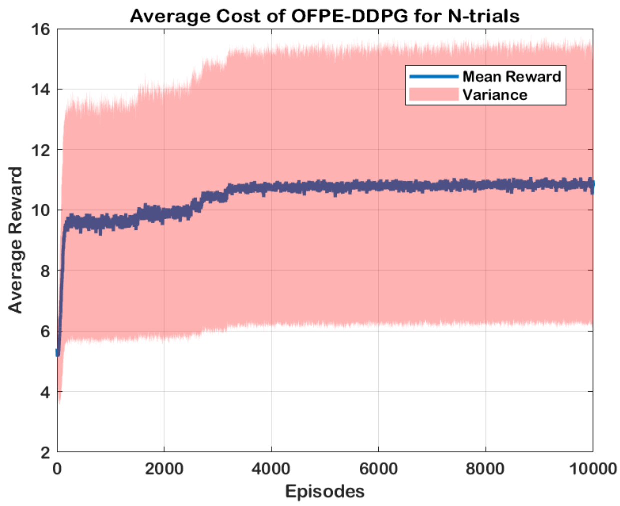

Fig. 13 The learning curve of 5 Agents of OFPE-DDPG

probability of generating defects and the majority rewards of the N trials generate prediction 0. Numerical results of OFPE-DDPG overall speaking demonstrate reasonably good effectiveness, robustness and interpretability, while further experiment-based validation is still needed.

6 Experiment validation

In this section, we carry out the experiment validation to demonstrate the effectiveness of the proposed OFPE-DDPG framework for the IBNE process. It should be noted that, apart from pH value, concentration of Zn^{2+}, current density and surface area of samples, all other parameters are kept constant during the experimental validation. For instance, the plating time is set to 20 minutes, the temperature is maintained at $25°C$, and the agitation speed is kept at 350 RPM (revolutions per minute).

6.1 First experiment results and analysis

In the first experiment, we randomly generated state data for 24 samples, and the well-trained DDPG agents provided the corresponding predicted current densities. With the help of the OFPE-DDPG framework, we successfully produced satisfactory plating results for 14 out of the 24 samples, while 10 samples failed to pass the defect detection, as shown in Fig. 14.

It is noted that, for the 10 samples with defects, the virtual IBNE environment also successfully predicted failed results for 3 of them. The primary reason for defects of the 3 samples may be that, the experimental conditions of the IBNE process were randomly initialized within a range that may not satisfy the basic requirements of IBNE, such as ion concentration being either too high or too low. Therefore, for defect detection, the virtual IBNE environment was consistent with the experimental

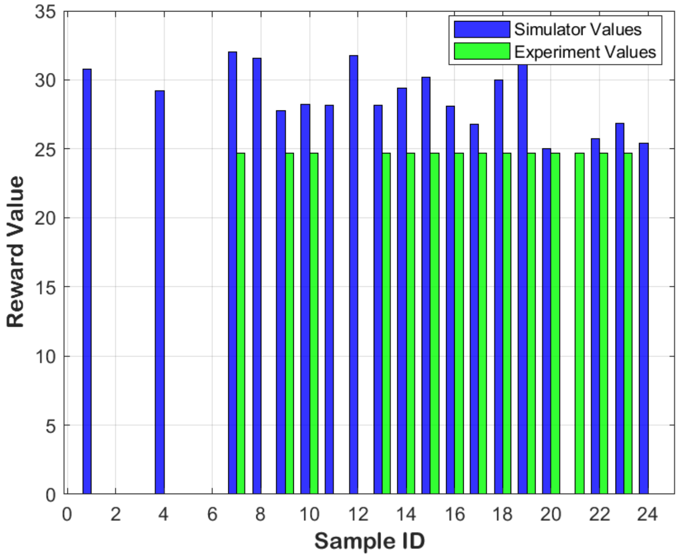

Fig. 14 First experiment: plating results different samples in the virtual IBNE environment and the real experiment with optimal current density given by OFPE-DDPG

results for 70.83% of the samples, while the OFPE-DDPG framework achieved a plating success rate of 66.67% among the effective samples. Additionally, even for samples without defects, a significant discrepancy was observed between the predicted rewards in the virtual IBNE environment and the real values from the experiments. This indicates a noticeable gap between the plating rate prediction model and the actual IBNE process. Specifically, for all samples in this experiment, the average MSE between the simulator's predicted plating rate and the actual value was 4.0336, with a variance of 3.7501.

6.2 Second experiment results and analysis

Based on the results of the first experiment, several weaknesses were identified in the virtual IBNE environment, which naturally affect the quality of the policy learned from it. Therefore, we utilized the new data collected from the first experiment to sequentially fine-tune the models in the virtual IBNE environment simulator and the OFPE-DDPG framework. Following this fine-tuning process, we conducted a second validation experiment.

In the second experiment, state data were randomly generated for 22 samples. The experimental results and the corresponding estimates are shown in Fig. 15. According to Fig. 15, 17 out of 22 samples passed the defect detection test, yielding a plating success rate of 77.28% among the effective samples. The figure also indicates a significant improvement in plating rate prediction accuracy. For most samples, the predicted rewards from the simulator closely matched the real rewards from the experiment, suggesting that the fine-tuning step substantially enhanced the performance of the

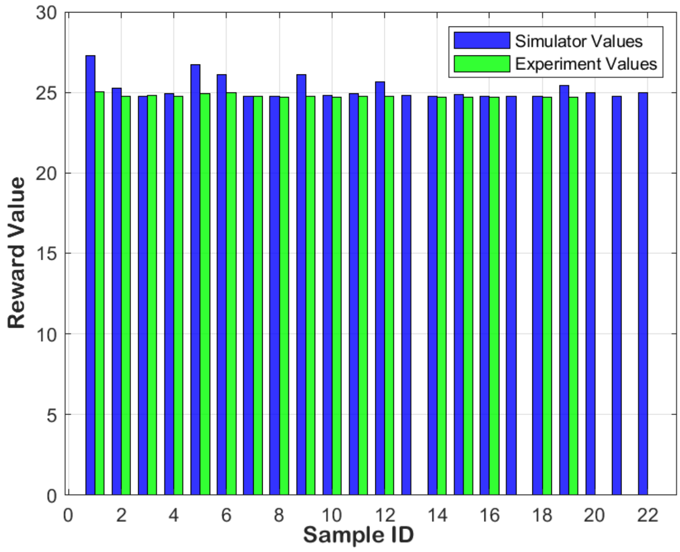

Fig. 15 Second experiment: plating results different samples in the virtual IBNE environment and the real experiment with optimal current density given by OFPE-DDPG

plating rate prediction model. In particular, for all samples in the second experiment, the average MSE between the simulator's predicted plating rate and the actual value was 0.2993, with a variance of 0.10174. Additionally, it should be noted that the plating rate in the second experiment are quite similar to one another, which aligns with the observed consistency in rewards. This is because the reward function was designed to be relatively conservative, given the instability of the virtual IBNE environment simulator developed with a small dataset. As a result, the learned policy was not overly aggressive in pursuing a higher plating rate, which reduced the likelihood of defects. This reflects a trade-off between maximizing the plating rate and minimizing defects, especially in the context of a small dataset.

Overall, these experimental results demonstrate the strong performance and considerable potential of the proposed OFPE-DDPG framework for tackling optimization problems in the IBNE process, even with a small dataset.

7 Conclusion and future work

In this paper, we have proposed an intelligent optimization framework, OFPE-DDPG, to tackle a multi-objective optimization problem for the IBNE process. This optimization framework consists of a modeling part and an optimization part. The modeling part is primarily based on a small historical dataset collected for defect detection and plating rate prediction, using few-shot learning and transfer learning. Transfer learning is employed to address the dataset imbalance issue caused by the small sample size. Subsequently, a virtual IBNE environment simulator is established based on these models. More importantly, a reward function with a penalty term is designed to eliminate and correct unreasonably high rewards caused by biases in the virtual IBNE environment. This ensures that the policy learned by OFPE-DDPG is safe and converges quickly. Additionally, this policy can dynamically respond with an optimal current density to maximize the reward corresponding to different states. Results of experimental validations indicate that the proposed OFPE-DDPG effectively addresses the optimization challenges inherent in the IBNE process.

Notably, although the optimization framework is theoretically general enough to be applied to other applications, particularly for processes without large datasets for modeling, certain limitations affect its deployment. One of the primary concerns is the quality of the data used for developing the virtual IBNE environment. High-quality data is fundamental to the performance of the entire framework, even for small datasets. In addition, the design of the objective function in multi-objective optimization is also a critical factor. A suitable objective function can help the algorithm converge quickly and reduce its complexity. Another challenge is the sim-to-real transfer. The primary issue here is the gap between the virtual environment and the real environment, which often results in the learned policy in the virtual environment not aligning with its performance in the real environment. This indicates that an extensive fine-tuning process and additional data collection may be needed to bridge this gap. In the future, we will develop more effective optimization frameworks to address these limitations and enhance their applicability to real-world scenarios.

Author contributions. Junhao Ren is responsible for the coding and writing of the manuscript. Qiyu Kang and Shuo Feng help in modeling and refining the codes, respectively. Laboratory experiments are carried out by Yong Tech Tan. Yajuan Sun and Gaoxi Xiao supervise the research and provide the fundamental concept. All authors have read and agreed to the published version of the manuscript.

Data availability. The modeling described in this paper uses a confidential (proprietary) data source obtained from SIMTech. Due to the pending technology disclosure, the data pertinent to the described technology is currently under confidentiality constraints and cannot be freely disseminated.

Code availability. The code developed and used in this study is available upon request.

Declarations

Conflict of interest. The authors declare no competing financial interests.

Funding. This work is supported by the Career Development Fund Project of the Agency for Science, Technology and Research (A * STAR) - "Continuous Process Optimization and Prescriptive Quality Control for Electrochemical Process Using Machine Learning Approach" (C210112027).

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the*

25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (p. 2623–2631). New York, NY, USA.

Durney, L.J. (1984). *Graham's electroplating engineering handbook*. Springer Science & Business Media.

Ehl, R.G., & Ihde, A.J. (1954). Faraday's electrochemical laws and the determination of equivalent weights. *Journal of Chemical Education*, *31*(5), 226, https://doi.org/10.1021/ed031p226

Fracas, P., Camarda, K.V., Zondervan, E. (2023). Shaping the future energy markets with hybrid multimicrogrids by sequential least squares programming. *Physical Sciences Reviews*, *8*(1), 121–156, https://doi.org/doi:10.1515/psr-2020-0050

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T.E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362, https://doi.org/10.1038/s41586-020-2649-2

He, H., Bai, Y., Garcia, E.A., Li, S. (2008). Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (p. 1322-1328).

Hosseini, F., Gharehchopogh, F.S., Masdari, M. (2023). Moaeosca: an enhanced multi-objective hybrid artificial ecosystem-based optimization with sine cosine algorithm for feature selection in botnet detection in iot. *Multimedia Tools and Applications*, *82*(9), 13369–13399, https://doi.org/10.1007/s11042-022-13836-6

Hou, Z., Li, Z., Hsu, C., Zhang, K., Xu, J. (2022). Fuzzy logic-driven variable time-scale prediction-based reinforcement learning for robotic multiple peg-in-hole assembly. *IEEE Transactions on Automation Science and Engineering*, *19*(1), 218-229, https://doi.org/10.1109/TASE.2020.3024725

Katirci, R., Aktas, H., Zontul, M. (2021). The prediction of the ZnNi thickness and Ni% of ZnNi alloy electroplating using a machine learning method. *Transactions of the IMF*, *99*(3), 162–168, https://doi.org/10.1080/00202967.2021.1898183

Katirci, R., & Danaci, K.I. (2023). The optimization of nickel electroplating process parameters with artificial intelligence methods. *Journal of Applied*

Electrochemistry, 1–13, https://doi.org/10.1007/s10800-023-01892-1

Khodadadi, N., Soleimanian Gharehchopogh, F., Mirjalili, S. (2022). Moavoa: a new multi-objective artificial vultures optimization algorithm. *Neural Computing and Applications*, *34*(23), 20791–20829, https://doi.org/10.1007/s00521-022-07557-y

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2019). *Continuous control with deep reinforcement learning.*

Lim, J.X.-Y., Leow, D., Pham, Q.-C., Tan, C.-H. (2021). Development of a robotic system for automatic organic chemistry synthesis. *IEEE Transactions on Automation Science and Engineering*, *18*(4), 2185-2190, https://doi.org/10.1109/TASE.2020.3036055

Liu, K., Niri, M.F., Apachitei, G., Lain, M., Greenwood, D., Marco, J. (2022). Interpretable machine learning for battery capacities prediction and coating parameters analysis. *Control Engineering Practice*, *124*, 105202, https://doi.org/10.1016/j.conengprac.2022.105202

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. (2013). *Playing atari with deep reinforcement learning.*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830,

Poroch-Seritan, M., Gutt, S., Gutt, G., Cretescu, I., Cojocaru, C., Severin, T. (2011). Design of experiments for statistical modeling and multi-response optimization of nickel electroplating process. *Chemical Engineering Research and Design*, *89*(2), 136-147, https://doi.org/10.1016/j.cherd.2010.05.010

Rao, S.S. (2019). *Engineering optimization: theory and practice.* John Wiley & Sons.

Solovjev, Denis, Solovjeva, Inna, Konkina, Viktoriya. (2019). Mathematical modelling and optimization of the electroplating process with a rotating cathode to reduce the non-uniformity of the coating thickness. *MATEC Web of Conferences*, *298*, 00014, https://doi.org/10.1051/matecconf/201929800014

Somasundaram, S., Pillai, A.M., Rajendra, A., P., A., Krishna, P.M., Sharma, A. (2018). Space qualification and characterization of high emittance black nickel coating on copper and stainless steel substrates. *Solar Energy Materials and Solar Cells*, *174*, 163-171, https://doi.org/10.1016/j.solmat.2017.08.023

Tang, J., Chai, T., Yu, W., Zhao, L. (2013). Modeling load parameters of ball mill in grinding process based on selective ensemble multisensor information. *IEEE Transactions on Automation Science and Engineering*, *10*(3), 726-740, https://doi.org/10.1109/TASE.2012.2225142

Teo, Z.H.A., Sun, Y., Tan, Y.T. (2022, November). Electroplating of black nickel: Bath age and its effect on coating quality. *Proceeding of 9th International Conference of Asian Society for Precision Engineering and Nanotechnology.* Singapore.

Wang, H., Li, B., Zhang, S., Xuan, F. (2024). Traditional machine learning and deep learning for predicting melt-pool cross-sectional morphology of laser powder bed fusion additive manufacturing with thermographic monitoring. *Journal of Intelligent Manufacturing*, 1–26, https://doi.org/10.1007/s10845-024-02356-9

Yang, J., Liu, H., Ma, K., Yang, B., Guerrero, J.M. (2023). An optimization strategy of price and conversion factor considering the coupling of electricity and gas based on three-stage game. *IEEE Transactions on Automation Science and Engineering*, *20*(2), 878-891, https://doi.org/10.1109/TASE.2022.3171446

Zhou, Z.-H. (2021). *Machine learning.* Springer Nature.