Mechanics & Industry **26**, 37 (2025)
© S. Wang et al., Published by EDP Sciences 2025
https://doi.org/10.1051/meca/2025031

Mechanics & Industry

Available online at:
www.mechanics-industry.org

Original Article

Open Access

Precision compensation technology of industrial robot based on OOA-DNN and transfer learning

Shuai Wang, Yuanbao Huang, Yongkang Mo, and Jianwen Guo*

School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, PR China

Received: 26 April 2024 / Accepted: 26 October 2025

Abstract. Addressing the issue of increased positioning errors in industrial robots during grasping and handling tasks, we propose an accuracy compensation technique optimized through the Osprey Optimization Algorithm (OOA) − Deep Neural Networks (DNN). This method leverages the efficient equilibrium between global and local searches characteristic of OOA to optimize the initial weights and biases of the DNN model, thereby enhancing training efficiency and model performance. Experimental results demonstrate significant improvement: based on a model trained with data points under a 60 kg load, the average, maximum, and standard deviation of positioning errors were reduced by 92.1%, 91.5%, and 91.0% respectively, outperforming DNN, Particle Swarm Optimized DNN, and Extreme Learning Machine methods. Furthermore, addressing varying degrees of positioning errors caused by different mass loads, we introduce an accuracy compensation method through transfer learning based on the OOA-DNN algorithm model. Experimental results reveal that adopting a freeze-thaw training strategy for transfer parameters achieves the best accuracy compensation effect; under a 120 kg load with 150 training data points, the mean error, maximum error, and standard deviation were reduced by 91.9%, 89.9%, and 88.4%, respectively. These improvements suggest that our method, requiring significantly fewer data, is especially valuable in data-constrained application scenarios.

Keywords: Industrial robots / precision compensation / osprey optimization algorithm / deep neural networks / transfer learning

1 Introduction

With the in-depth application of industrial robots, they have begun to enter some high-precision manufacturing fields, such as high-precision equipment assembly, high-precision laser welding, and flexible grinding. However, the application of industrial robots in high-precision manufacturing requires further improvement of their positioning accuracy through offline calibration technology. Industrial robot offline calibration technology, executed external to the production line, bifurcates into kinematic and non-kinematic calibration. Kinematic calibration [1] involves the formulation of a kinematic model [2] for the industrial robot, utilizing measurement devices like laser tracker [3] to accrue data on the robot's posture and position. This data underpins the calibration of the developed kinematic model through algorithms for model parameter identification [4], thereby enabling accuracy compensation. Despite kinematic calibration's efficacy in correcting geometric errors, it falls short in addressing non-geometric errors, entails intricate model computations, and necessitates authorization to access the control system of the industrial robot. Conversely, non-kinematic calibration obviates the need for constructing the robot's kinematic model, opting instead to directly estimate errors in the robot's position and posture via an error prediction model, with the accrued error data serving as the basis for accuracy compensation. This technique is adept at compensating for both geometric and non-geometric errors, delivering high precision and adaptability, thereby augmenting the robot's positioning accuracy.

Non-kinematic calibration methods, predicated on disparate error prediction principles, primarily leverage spatial interpolation and neural network methodologies [5,6]. Yuan [7] introduces a compensation technique based on the limit learning machine model, evidenced by precision compensation experiments on an aerial drilling robot that underscore a significant enhancement in the robot tool center point's average and maximal absolute position accuracy post-compensation. Cao [8] offers a calibration methodology predicated on an extended Kalman filter and artificial neural network, which optimizes geometric parameter error in a robust manner against Gaussian noise in nonlinear systems, as evinced by the substantive improvement in robot attitude accuracy.

* e-mail: guojw@dgut.edu.cn

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Utilizing a deep confidence network alongside error similarity, Wang [9] establishes a model mapping the robot's theoretical attitude coordinates to actual attitude errors, thereby facilitating attitude error prediction, with experiments showing a marked reduction in the robot end actuator's maximum absolute position error post-compensation. Ma [10] advocates for an incremental extreme learning machine model to forecast industrial robots' positioning errors, achieving significant error reduction post-compensation. Present research on non-kinematic calibration methods predominantly focuses on the amalgamation of intelligent algorithms for neural network optimization, thereby elevating the accuracy compensation level.

Deep Neural Networks (DNN) [11] are characterized by their multi-layer neural network architecture and their potent non-linear fitting capabilities. With the escalation of computing power, DNNs have found extensive applications. The Osprey Optimization Algorithm [12] (OOA), inspired by ospreys' hunting behavior, epitomizes a swarm intelligence optimization algorithm with exemplary global and local search proficiencies. Transfer learning [13,14], as a machine learning paradigm, endeavors to ameliorate performance across disparate yet related tasks by harnessing knowledge from one task, thereby optimizing model performance, curtailing data requisites, and expediting the training regimen.

This paper addresses the burgeoning issue of positioning errors in industrial robots during intensive load grasping and handling endeavors. It promulgates an accuracy compensation method amalgamating OOA and DNN, denoted as OOA-DNN. Furthermore, acknowledging that varying mass loads precipitate differing degrees of positioning errors in industrial robots, this investigation incorporates a transfer learning technique into the accuracy compensation realm of industrial robots, delineating a methodology to rectify the positioning accuracy amidst heavy-load grasping activities. This approach is designed to diminish the necessity for extensive data collection and to amplify data gathering efficacy. Experiments on a custom-built industrial robot heavy-load grasping test platform affirm the proposed methods' efficacy.

2 Experimental platform and error data analysis

2.1 Experimental setup and parameter configuration

In this investigation, a dedicated experimental setup, as depicted in Figure 1, was developed to facilitate the execution of accuracy compensation experiments. This setup encompasses an industrial robot, specified end-effector loads, a laser tracker, and a designated target sphere for the tracker's targeting purposes. The industrial robot selected for this study is the BRTIRUS3511A model, a product of the domestic Bronte Company. The Brtirus3511A six-axis industrial robot produced by Bronte Company was adopted in this study. Its maximum arm span can reach 3500 mm, the initial attitude height is 2227 mm, the rated load is 100 kg, the maximum load can reach 120 kg, and the repeated positioning accuracy is ±0.2 mm. For the experiments, end-effector loads were meticulously chosen at 60 kg and 120 kg to simulate varying operational conditions. The precision measurements were enabled by the VantageE model laser tracker, sourced from FARO Company, renowned for its leading specifications, which are systematically cataloged in Table 1 for detailed reference.

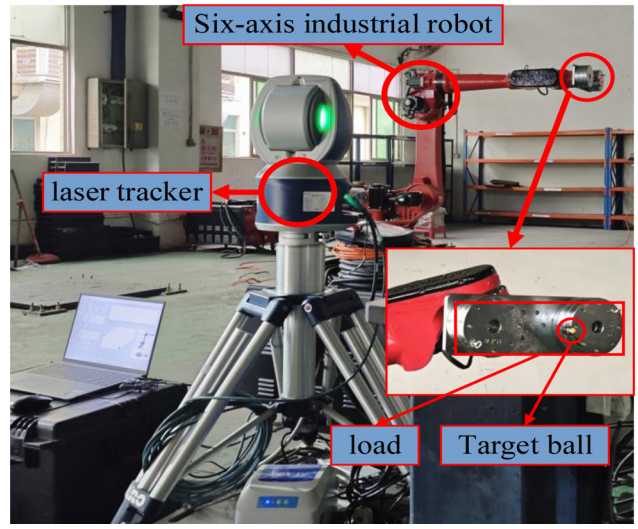


Fig. 1. Industrial robot accuracy compensation experimental platform.

Table 1. Main specifications of the FARO VantageE laser tracker.

Device parameter type	Specification
Maximum measuring radius range	35 m
Horizontal rotation angle	360°
Vertical rotation angle	$-52.1° \sim 77.9°$
Distance resolution	0.5 μm
Distance accuracy	6 μm + 0.8 μm/m

2.2 Coordinate system transformation relationship

In the process of quantifying positional coordinate errors of industrial robots using a laser tracker, it's imperative to delineate a transformational correlation between the laser tracker's coordinate system and the foundational coordinate system of the robot, in addition to determining the tool coordinates for the target sphere. This undertaking, illustrated in Figure 2, is realized through the manipulation of the robot's 1st, 3rd, and 6th axes to generate coordinate data for the robot's reference points during single-axis movements within the coordinate system {T} of the laser tracker. From the coordinates of these reference points, the rotation matrix R and translation matrix T are derived to effectuate the transformation from the robot's base coordinate system to the coordinate points perceived by the laser tracker. Denoting the spatial positioning of the target sphere within the robot's base coordinate system as P^B, and within

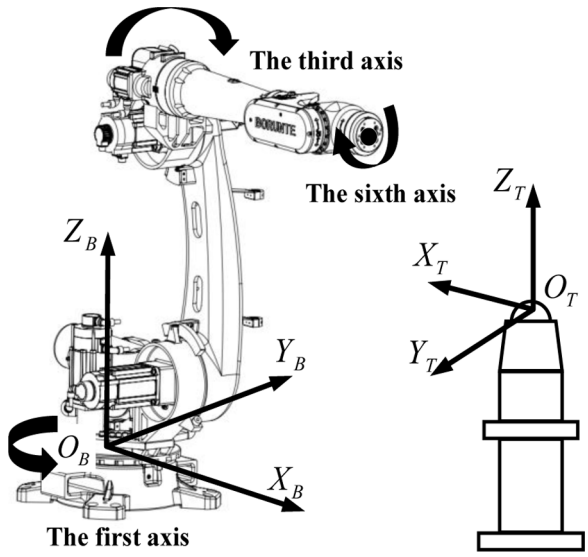


Fig. 2. Industrial robot base coordinate system and laser tracker coordinate system.

the tracker's coordinate system as P^T, the ensuing transformation equation is articulated as follows:

$$P^T = RP^B + T. \tag{1}$$

The coordinate data transformed from the laser tracker coordinate system {T} to the industrial robot base coordinate system {B} is P^{TB}, which can be obtained from equation (1):

$$P^{TB} = R^{-1}\left(P^T - T\right). \tag{2}$$

Then, the error data E in the industrial robot base coordinate system {B} is:

$$E = P^B - P^{TB} = \left[\Delta E_x \Delta E_y \Delta E_z\right]. \tag{3}$$

The comprehensive position error is:

$$\Delta E = \sqrt{\Delta E_x^2 + \Delta E_y^2 + \Delta E_z^2}. \tag{4}$$

2.3 Planning and analysis of robot sampling points

The methodology for accuracy compensation in industrial robots necessitates the collection of the robot's positional (x, y, z) coordinates alongside its orientation angles (u, v, w). In this study, the Latin hypercube sampling method [15]—a stratified sampling approach designed for the equitable selection of sample points across a multidimensional space—is utilized. This method enables a systematic and uniform extraction of data points, vital for comprehensive accuracy analysis. A rectangular sampling domain, depicted in Figure 3 and situated within the operational arena of the robot, was demarcated with dimensions set to 400 mm, 600 mm, and 300 mm along the X, Y, and Z axes of the robot's foundational coordinates, respectively. The scope of the orientation angles (u, v, w) was meticulously calibrated to ±15° for each axis to ensure a broad representation of potential robot positions and orientations. Details regarding the range of each sampling variable within this defined space are methodically compiled in Table 2, ensuring a clear and organized presentation of the sampling framework employed.

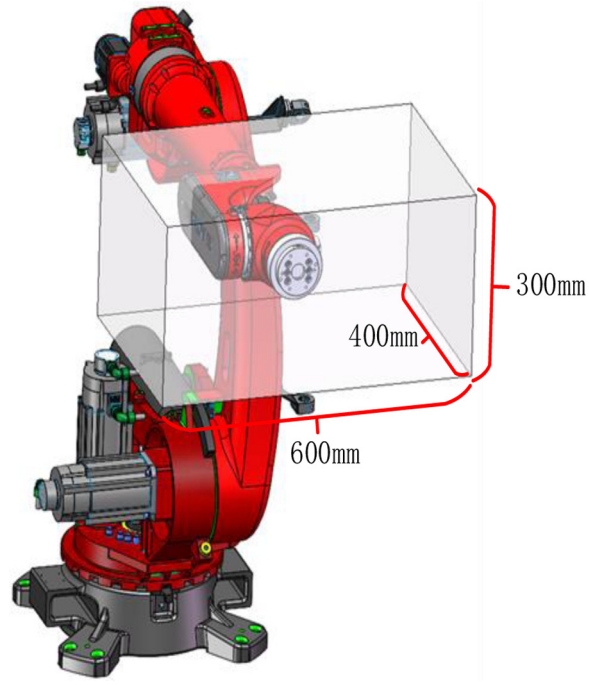


Fig. 3. A Schematic diagram of the cuboid sampling space for industrial robots.

Table 2. Variables of the Latin hypercube sampling space.

Sampling space variable	Range
x	2024.333 mm~2424.333 mm
y	−427.136 mm~172.864 mm
z	1912.894 mm~2212.894
u	−15°~ 15°
v	−15°~ 15°
w	−15°~ 15°

Applying the Latin hypercube sampling method, 1500 sampling points were strategically determined, facilitating the undertaking of experiments to examine the positioning error associated with sampling points when subjected to 60 kg and 120 kg loads. Scatter plots, illustrating the aggregate positioning errors for both 60 kg and 120 kg loads as presented in Figure 4, elucidate the error distribution patterns under varying load conditions. Notably, despite the differential in loads, these scatter plots reveal a degree of uniformity, hinting at a fundamental consistency in the overall positioning errors experienced by industrial robots across disparate loading scenarios. Moreover, the line

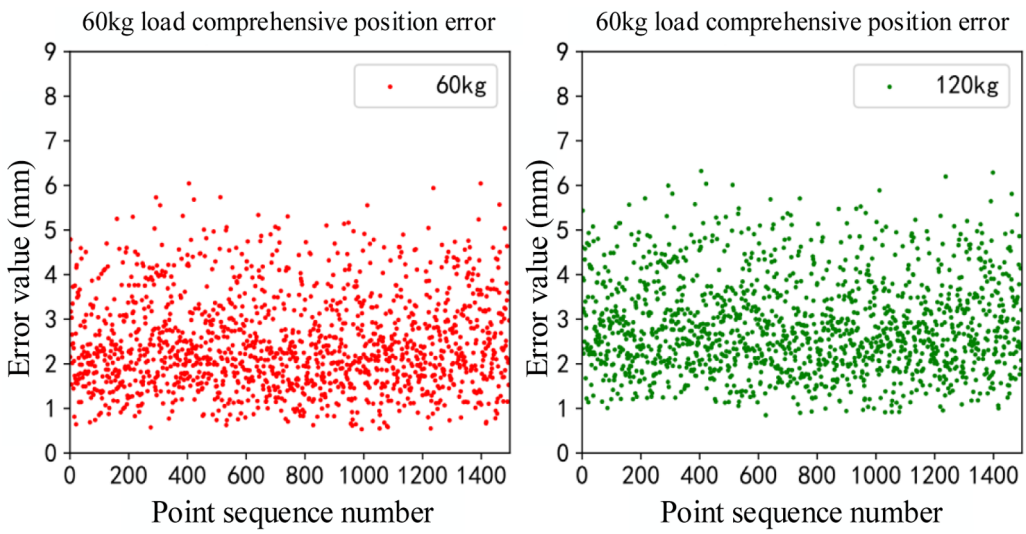


Fig. 4. Integrated plot of 60 kg and 120 kg load points.

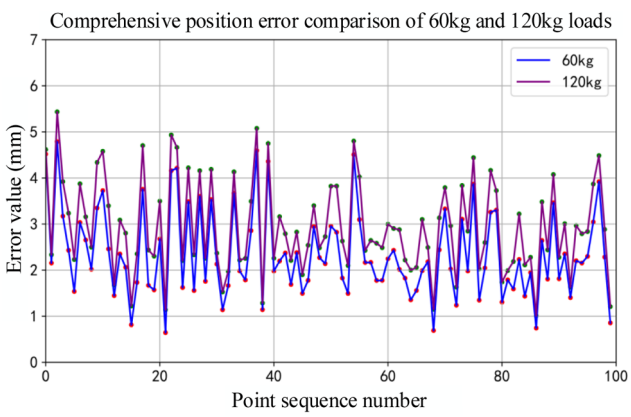


Fig. 5. Line diagram of the integrated position error of the 60 kg and 120 kg load points.

charts portraying the aggregate positioning errors associated with 60 kg and 120 kg loads, depicted in Figure 5, delineate analogous trends in the evolution of positioning errors across 100 distinct points under the two varied load states. This pattern suggests that, although an increase in load generally precipitates a rise in the magnitude of errors, the nature of error fluctuation amongst different points exhibits a remarkable uniformity. Such congruence in error distribution significantly bolsters the investigation into the adaptability of industrial robots, particularly in the context of transfer learning, highlighting their potential to maintain precision and reliability across a spectrum of operational demands.

Table 3 encapsulates the statistical analysis of the data derived from 1500 sampling points. Notably, when comparing the effects of load variations from 60 kg to 120 kg on the industrial robot's positioning accuracy, there is a discernible increment in the mean error by 0.395 mm, the standard deviation by 0.007 mm, and the maximum error by 0.279 mm. This statistical outcome underscores a clear trend: the overall positioning error of industrial robots escalates with the increase in operational load. Such data underscore the inherent challenges posed by heavier loads, confirming the hypothesis that load intensification adversely affects the robot's positioning precision.

Table 3. Comprehensive positioning error data statistics for sampling points.

Category	60 kg Load	120 kg Load
Mean	2.450 mm	2.845 mm
Standard deviation	1.037 mm	1.044 mm
Error range	0.534 mm~6.044 mm	0.848 mm~6.323 mm

3 Accuracy compensation method based on OOA-DNN

3.1 Osprey optimization algorithm (OOA)

The OOA, proposed by Dehghani [12] in 2023, simulates the hunting behavior of ospreys and represents a swarm intelligence optimization algorithm. In the OOA, each osprey's position represents a set of candidate solutions. Through iterative optimization, the positions of the ospreys are updated continually, aiming for the optimal solution. The osprey population can be expressed using matrix notation as in equation (5), with each osprey's position being randomly initialized as shown in equation (6).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,m} \end{bmatrix}. \quad (5)$$

$$x_{i,j} = lb_j + r_{i,j}(ub_j - lb_j), i = ,..., N, j = 1,2,...,m. \quad (6)$$

Here, matrix X is the population matrix of osprey positions, N is the number of ospreys, and X_i is the position of the i osprey. m is the number of variables in the optimization problem, j represents the j variable, $r_{i,j}$ is a random number between [0,1], and lb_j and ub_j are the lower and upper bounds, respectively, of the j variable's search space.

The target function set of the osprey population can be expressed using matrix F in equation (7).

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}. \quad (7)$$

F_i represents the target function value of the i osprey, with the best value corresponding to the optimal candidate solution. The positions of the ospreys are updated iteratively within the search space based on the values of the target function.

First Phase: Global Search. This phase models the global search aspect of the OOA, simulating the natural behavior of ospreys hunting fish. To adapt and update flexibly in search of the best solution, each osprey considers the position of other ospreys with better target function values as "fish" underwater, determined by equation (8).

$$FP_i = \{X_k | k \in \{1, 2, ..., N\} \wedge F_k < F_i\}\{X_{best}\}. \quad (8)$$

Here, FP_i is the position set of the i th osprey, and X_{best} is the position of the best osprey. Based on simulating the osprey's movement towards the fish for an attack, the new position of each osprey is calculated using equations (9) and (10). If the new position yields a better target function value, it replaces the osprey's previous position as indicated in equation (11).

$$x_{i,j}^{P1} = x_{i,j} + r_{i,j}(SF_{i,j} - I_{i,j}x_{i,j}), \quad (9)$$

$$x_{i,j}^{P1} = \begin{cases} x_{i,j}^{P1}, lb_j \le x_{i,j}^{P1} \le ub_j \\ lb_j, x_{i,j}^{P1} < lb_j \\ ub_j, x_{i,j}^{P1} > ub_j \end{cases}, \quad (10)$$

$$X_i = \begin{cases} x_i^{P1}, F_i^{P1} < F_i \\ X_i, else \end{cases}. \quad (11)$$

Here, x_i^{P1} represents the new position of the i th osprey in the first stage, $x_{i,j}^{P1}$ is the size of its j th problem variable to be optimized, and F_i^{P1} is the objective function value of the Osprey position. $SF_{i,j}$ is the fish chosen by the i th osprey, $r_{i,j}$ is the random number within the interval [0,1], and I $I_{i,j}$ is the number randomly chosen from the set {1,2}.

Second Phase: Local Search. Simulating the behavior of ospreys carrying fish to a suitable location for consumption, a new random position is calculated as the "suitable for eating fish" position for the osprey, as shown in equations (12) and (13). If this new position improves the target function value, it supersedes the osprey's prior position according to equation (14).

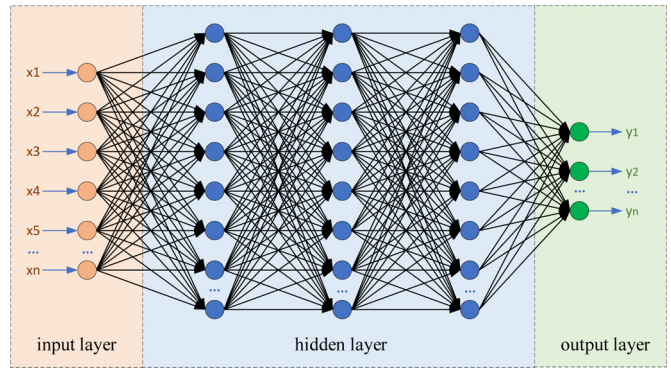


Fig. 6. Network structure of the DNN.

$$x_{i,j}^{P2} = x_{i,j} + \frac{lb_j + r_{i,j}(ub_j - lb_j)}{t}, \quad (12)$$

$$x_{i,j}^{P2} = \begin{cases} x_{i,j}^{P2}, lb_j \le x_{i,j}^{P2} \le ub_j \\ lb_j, x_{i,j}^{P2} < lb_j \\ ub_j, x_{i,j}^{P2} > ub_j \end{cases}, \quad (13)$$

$$X_i = \begin{cases} x_i^{P2}, F_i^{P2} < F_i \\ X_i, else \end{cases}. \quad (14)$$

Here, x_i^{P2} represents the new position of the ith osprey in stage 2, $x_{i,j}^{P2}$ is the size of the j th problem variable to be optimized, F_i^{P2} is the objective function value of the new position of the osprey, and $r_{i,j}$ is the random number in the interval [0,1]. And t is the iteration counter of the algorithm, and T is the total number of iterations.

3.2 OOA-DNN algorithm

Deep Neural Networks (DNN) are models with multiple layers of neural networks known for their robust non-linear fitting capability. The structure of a DNN network, as shown in Figure 6, comprises input, hidden, and output layers. Each layer contains several neurons, with neurons between layers fully connected to form a comprehensive network. The network is optimized through backpropagation, updating weights and biases between neurons to refine the DNN model.

The initialization of weights and biases in a neural network is crucial for the algorithm's convergence. Proper initialization can facilitate rapid model convergence, while poor initialization may lead to information loss in forward propagation or gradient vanishing in backpropagation, making the model difficult to train. Therefore, this paper introduces the OOA-DNN algorithm, which employs the OOA to obtain optimal initial weights and biases for the DNN model, enhancing training effectiveness and convergence speed, and boosting predictive accuracy. Flowchart of the OOA-DNN algorithm is shown as in Figure 7.

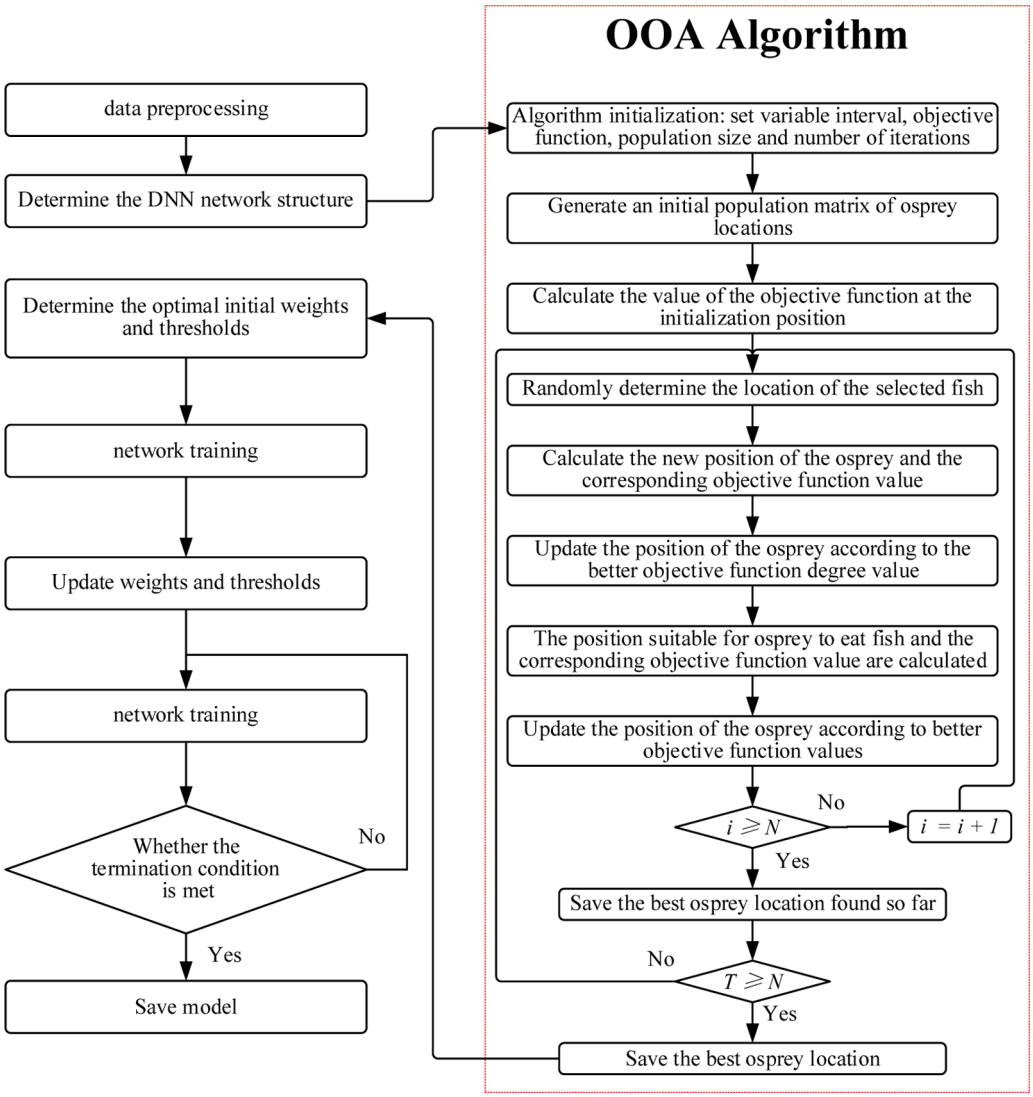


Fig. 7. Flowchart of the OOA-DNN algorithm.

The OOA-DNN represents DNN weights and biases with the spatial positions of ospreys, using the mean squared error of the OOA-DNN training set as the error function of the OOA-DNN algorithm, as shown in equation (15).

$$f = \frac{1}{n}\sum\left(y_{true} - y_{pred}\right)^2, \tag{15}$$

Where y_{true} is the actual output of the training set, y_{pred} is the predicted output, and n is the number of training set items.

Steps of the OOA-DNN Algorithm:

- Normalize sample data and divide it into training and test samples.
- Initially determine the DNN network structure.
- Represent DNN weights and biases as the spatial positions of ospreys to obtain optimal initial weights and biases.
- Conduct iterative training and testing of the DNN.
- Save the predictive model if end conditions are met.

3.3 Verification and comparative analysis of the OOA-DNN model compensation effect

3.3.3 Establishment of the error prediction model

This study uses the position coordinates (x, y, z) and orientation angles (u, v, w) of the industrial robot as inputs, with positioning error E (ΔEx, ΔEy, ΔEz) as the output, to establish a prediction model mapping theoretical posture data of industrial robots to positioning errors. The number of hidden layers and neurons in the neural network is critical for model design, directly affecting model performance and learning capability.

To identify the optimal combination of hidden layer numbers and neuron counts, Bayesian optimization is employed for automatic parameter tuning, using the mean squared error of the neural network on the training set as the optimization target function. After 1000 iterations and sorting the obtained data by the target function values, the top 100 combinations of hidden layer numbers and neuron

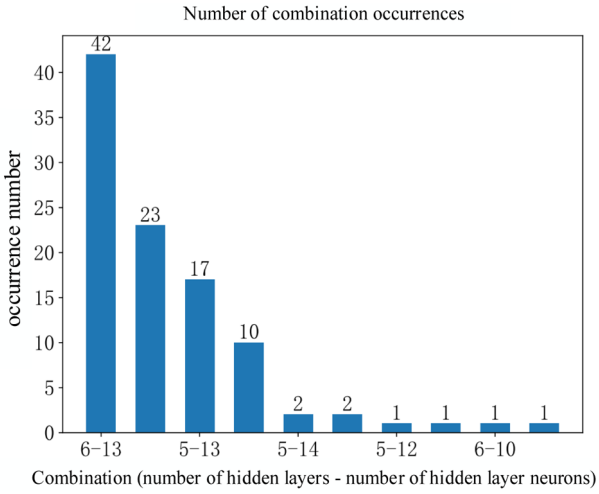


Fig. 8. Statistics of the number of occurrences of combinations.

Table 4. Algorithm parameters.

Parameter	Setting
Population size	50
OOA iterations	200
Osprey position range	[−1,1]
Target optimization algorithm	Adam algorithm
Activation function	ReLU function
Initial learning rate	0.01
DNN iterations	1000
Target function	Training set mean squared error

counts were analyzed to determine the optimal combination. As shown in Figure 8, ten combinations emerged, with the most frequent combination (6 hidden layers − 13 neurons) accounting for 42% of the cases. Based on this statistical analysis, the optimal network structure for the error prediction model was determined to be 6 hidden layers with 13 neurons each.

3.3.2 Training of the error prediction model

To verify the accuracy compensation effect of the OOA-DNN algorithm's error prediction model for industrial robots, data from 1500 sampling points generated using the Latin hypercube sampling method in the robot's sampling space under a 60 kg load were used as training data for the algorithm. The parameters of the algorithm are as listed in Table 4, with the model evaluated based on the R^2 score of the training set to determine the optimal model.

3.3.3 Comparative analysis of model compensation effect

To demonstrate the superiority of the OOA-DNN algorithm proposed in this paper, an accuracy compensation experiment was conducted on 100 point data within the industrial robot's working space. Four models were compared: OOA-DNN model, DNN model, Particle Swarm Optimized DNN model, and Extreme Learning Machine model. The accuracy compensation effects of different error prediction models in practical applications were evaluated. The trained error prediction models were used to forecast errors for 100 points, and the theoretical values of the point data were adjusted inversely with their predicted positioning errors to obtain the coordinates after accuracy compensation. Finally, the compensated point coordinates were used as positioning commands for the industrial robot, controlled via the robot's controller, and the post-compensation positioning errors were measured using the laser tracking system.

The changes in comprehensive positioning errors before and after compensation are shown in Figure 9, with the statistical data of comprehensive positioning errors before and after compensation listed in Table 5. After applying the OOA-DNN algorithm model for accuracy compensation, the mean error of 100 points on the industrial robot was reduced by 92.1%, the maximum error by 91.5%, and the standard deviation by 91.0%. The mean error, error range, and standard deviation were smaller compared to those of the other three algorithm models after compensation, indicating that after accuracy compensation using the OOA-DNN algorithm model, the positioning errors of the industrial robot across 100 points were reduced, and the distribution of positioning errors became more concentrated and stable, showcasing the superior performance of the OOA-DNN algorithm model in accuracy compensation for industrial robots.

4 Accuracy compensation method for OOA-DNN model via transfer learning

Analyzing the positioning errors of industrial robots under different loads reveals a high similarity in the distribution patterns and trends of positioning errors. Based on this characteristic, this paper proposes a transfer learning method based on model parameters. The OOA-DNN algorithm error prediction model trained under a 60 kg load serves as the source domain model for transfer learning. A DNN algorithm establishes the error prediction model as the target domain model for transfer learning. As shown in Figure 10, by transferring the parameters of the source domain model to the target domain model and then fine-tuning the target domain model based on sampling point data obtained from the industrial robot under a 120 kg load, the model is adapted to the target task.

4.1 Training strategy of the error prediction model based on transfer learning

After transferring the source domain model parameters to the target domain model, the training strategy for the target domain is crucial for improving model performance and adaptability. This paper investigates three training strategies based on whether the transfer parameters are frozen during the fine-tuning process of the target model: freezing

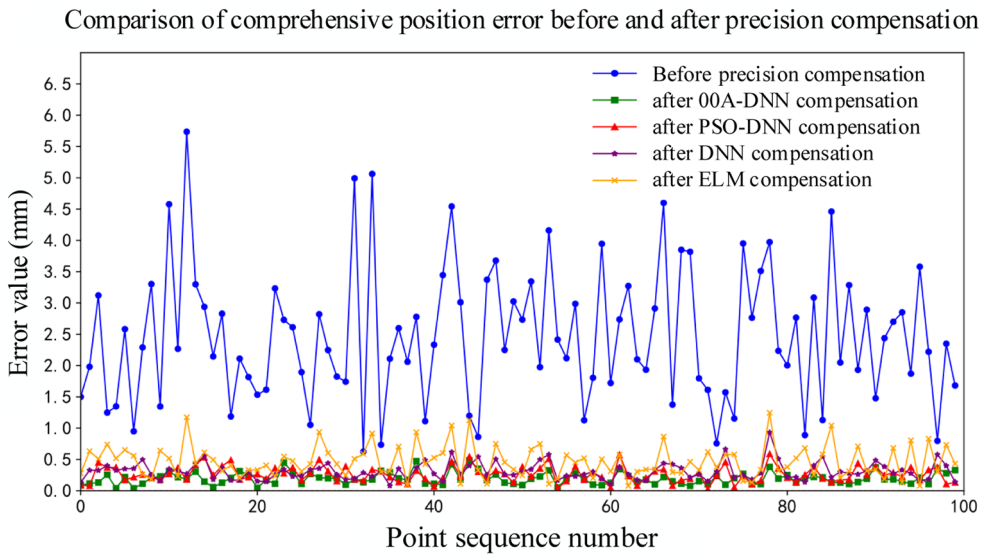


Fig. 9. Comparison of integrated position error before and after model accuracy compensation of the four algorithms.

Table 5. Statistical data of comprehensive positioning errors before and after compensation using four algorithm models.

Category	Before compensation	After OOA-DNN compensation	After PSO-DNN compensation	After DNN compensation	After ELM compensation
Mean value	2.464 mm	0.194 mm	0.262 mm	0.310 mm	0.471 mm
Standard deviation	1.083 mm	0.097 mm	0.128 mm	0.140 mm	0.256 mm
Error range	0.627~5.736 mm	0.039~0.486 mm	0.023~0.589 mm	0.080~0.933 mm	0.078~1.245 mm

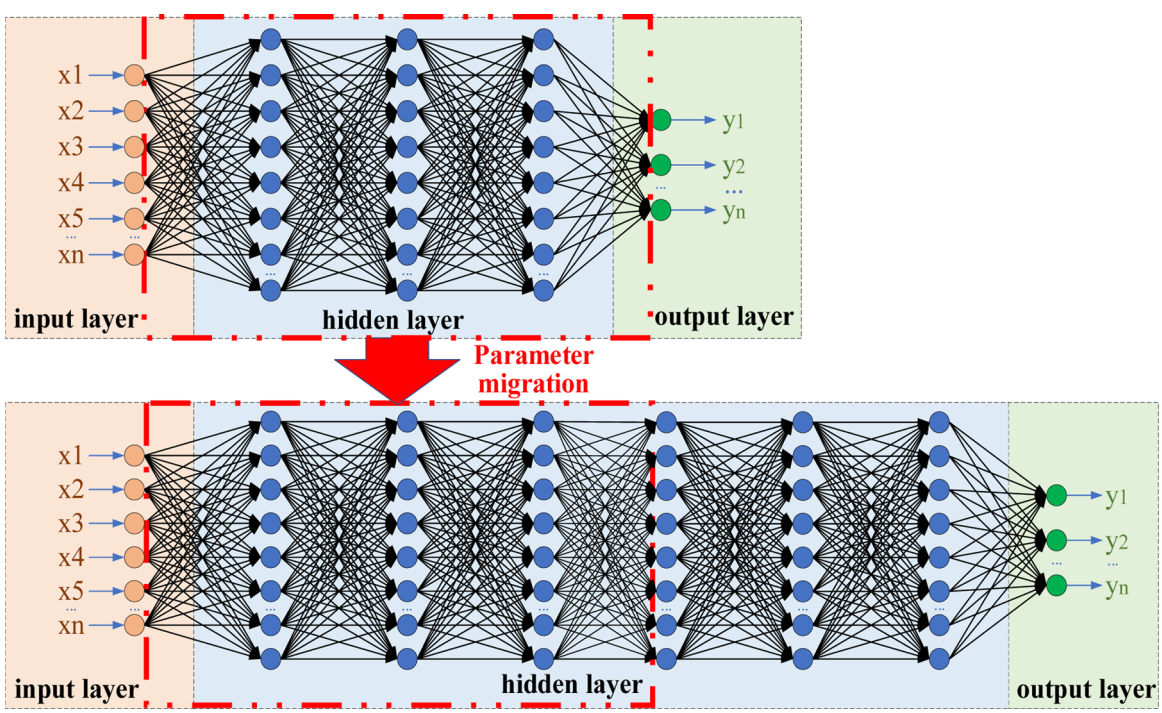
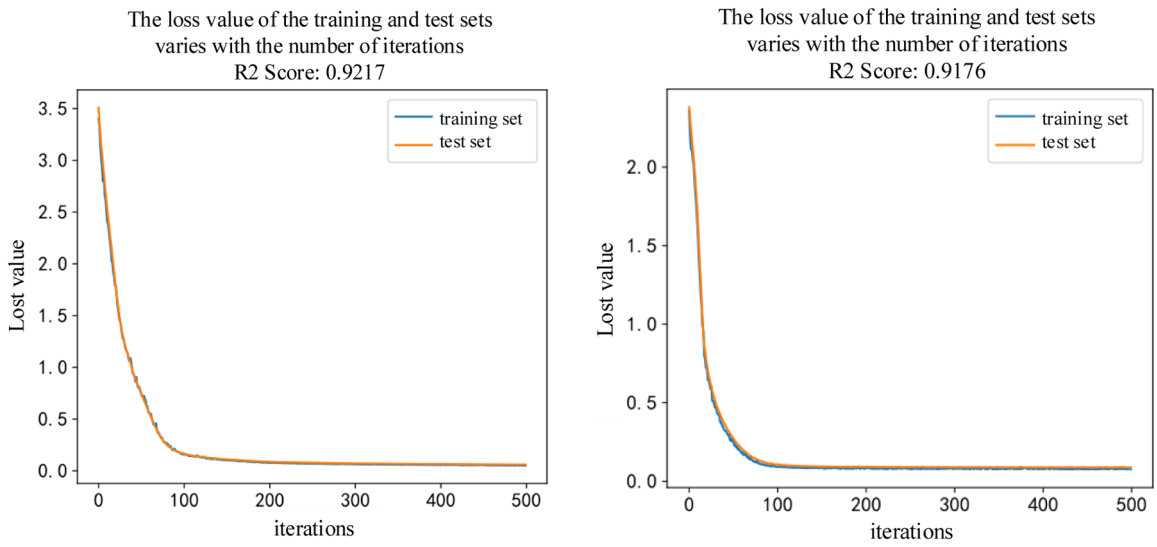


Fig. 10. Parameter migration based on the OOA-DNN error prediction model.



(a) Freeze parameter (b) No freezing of the parameters

Fig. 11. Image of iteration actions of training and test loss values.

transfer parameters, not freezing transfer parameters, and freezing before unfreezing transfer parameters. Freezing transfer parameters means keeping the parameters of the source domain model unchanged during the fine-tuning of the target domain model. The strategy of not freezing transfer parameters allows for adjustments to the parameters of the source domain model during fine-tuning. In the strategy of freezing before unfreezing transfer parameters, the transfer parameters are initially frozen for fine-tuning, and then unfrozen later in the training process for further adjustments.

To compare the effects of the three transfer training strategies, a training set size of 300 and 500 iterations per training were set, conducting 20 training sessions for each transfer learning strategy. The best R^2 score models from each group represent the respective transfer learning training strategy, as shown in Figure 11. Both strategies showed effective training results, with the loss curves for training and test datasets decreasing synchronously and converging smoothly, indicating good adaptability of the model to data variations and its ability to generalize well on unseen test data without signs of overfitting or underfitting.

As both strategies converged by iteration 200, in the freezing before unfreezing transfer parameters strategy, transfer parameters were frozen when the iteration number was less than 200, and unfrozen when it was greater than or equal to 200, training all parameters of the model until reaching 500 iterations, as shown in Figure 12. The loss values for the training and test datasets tended to converge between iterations 100 and 200, with no further decrease in loss values. At the convergence stage of freezing transfer parameters, a further drop in loss values can be observed when the iteration number reaches 200, stabilizing around iteration 300. This strategy achieved a better R^2 score than the first two, indicating superior training performance.

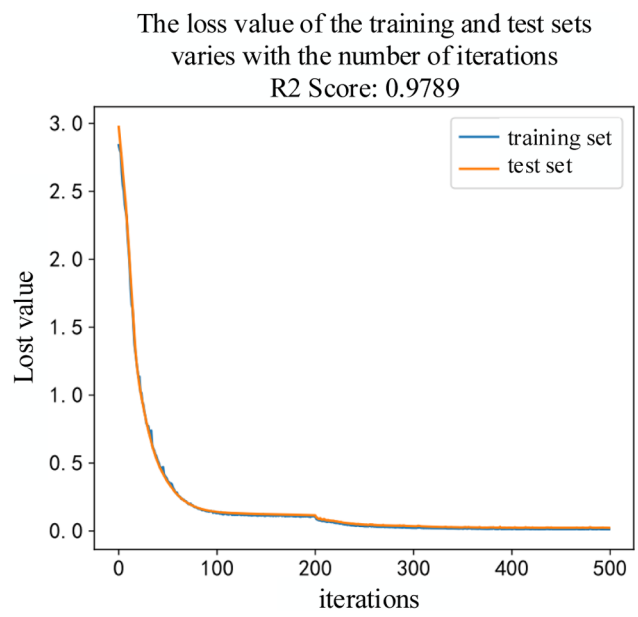


Fig. 12. Freeze before thawing the iteration change images of migration parameter training and test loss values.

4.2 Validation and comparative analysis of the compensation effect of transfer learning model

To investigate the training effects of the transfer learning model with different training set sizes, this paper selected 10 groups of training sets of varying sizes, incrementing by 10 groups each time until reaching a total of 500 groups. Fifty experiments were conducted across different training set sizes, iterating 10 times for each. The best R^2 score model from each iteration was selected as the representative. Results, as shown in Figure 13, display the trend of

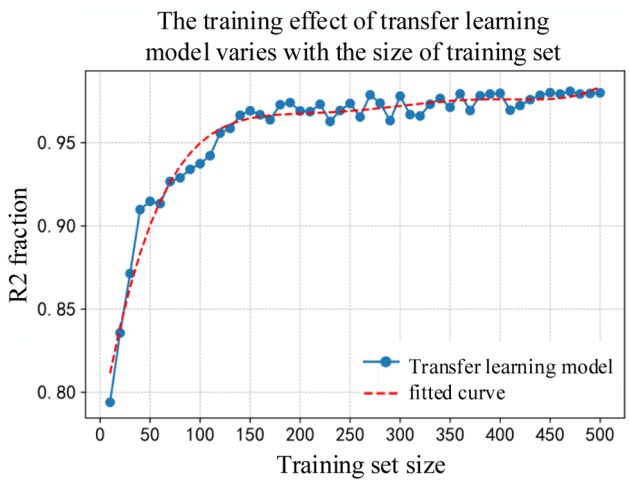


Fig. 13. The training effect of the transfer learning model changes with the training set size.

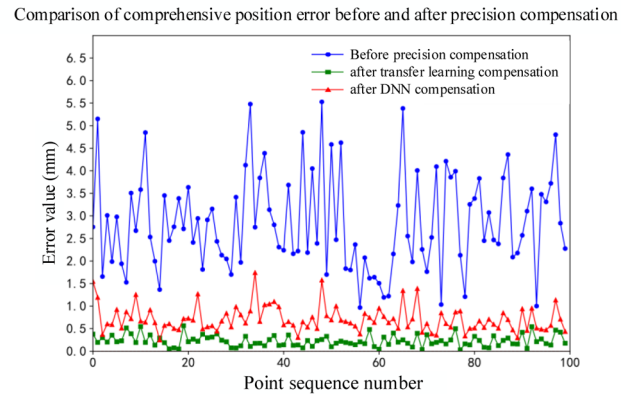


Fig. 14. Comparison of comprehensive position error before and after accuracy compensation of transfer learning model and DNN algorithm model.

training effects of the transfer learning model with the size of the training set. It is observed that when the number of training sets reaches 120 or more, the best R^2 scores can reach above 0.95. A fifth-degree polynomial function fitted using nonlinear least squares describes the data trend, indicating that changes in the best R^2 scores become stable when the size of the training set exceeds 150 groups, suggesting that increasing the size of the training set beyond this stage has a minimal impact on model performance improvement.

To verify the superiority of the transfer learning algorithm proposed in this paper, a model accuracy compensation experiment was conducted on 100 point data within the industrial robot's working space, comparing the transfer learning model trained with 150 data sets against the DNN model, evaluating the accuracy compensation effects of models in practical applications. Using the trained error prediction models, errors for 100 points were predicted, and the theoretical values of the point data were adjusted inversely with their predicted positioning errors to obtain coordinates after accuracy compensation. Finally, the compensated point coordinates were used as positioning commands for the industrial robot, controlled via the robot's controller, and post-compensation positioning errors were measured using the laser tracking system.

Changes in comprehensive positioning errors before and after compensation are shown in Figure 14, with statistical data of comprehensive positioning errors before and after compensation listed in Table 6. After applying the transfer learning algorithm model for accuracy compensation, the mean error of 100 points on the industrial robot was reduced by 91.9%, the maximum error by 89.9%, and the standard deviation by 88.4%. The mean error, error range, and standard deviation were smaller compared to those after DNN model compensation, demonstrating that with only 150 training data sets, the transfer learning algorithm performs better.

5 Conclusion

This paper addresses the issue of increasing positioning errors in industrial robots during heavy-load grasping and handling tasks by proposing the OOA-DNN algorithm. The experimental results show:

- After compensating for accuracy based on a model trained with 1500 sets of data under a 60 kg load, the mean error of the industrial robot during heavy-load grasping and handling tasks was reduced by 92.1%, the maximum error by 91.5%, and the standard deviation by 91.0%. The mean error, error range, and standard deviation were smaller compared to those after compensation using the DNN model, Particle Swarm Optimized DNN model, and Extreme Learning Machine model, indicating superior compensation effects.
- In response to the issue that different mass loads can lead to varying degrees of positioning errors in industrial robots, a transfer learning-based accuracy compensation method for heavy-load grasping and handling tasks using the OOA-DNN algorithm model was proposed. Experimental results show that after accuracy compensation using the transfer learning algorithm model with 150 sets of training data under a 120 kg load, the mean error of the industrial robot was reduced by 91.9%, the maximum error by 89.9%, and the standard deviation by 88.4%. The mean error, error range, and standard deviation were smaller compared to those after compensation using the DNN model, demonstrating that with only 150 sets of training data, the transfer learning algorithm performs better. This significantly reduces the need for data collection and enhances the efficiency of data gathering. The transfer learning-based accuracy compensation method for industrial robots during heavy-load grasping and handling tasks, applied between different loads on the same industrial robot, can be extended in future research to include transfer learning-based positioning accuracy compensation between different industrial robots, further expanding its application scope and reducing the work involved in data collection.

Table 6. Statistical data of comprehensive positioning errors before and after compensation using OOA-DNN algorithm.

Category	Before compensation	After transfer learning compensation	After DNN compensation
Mean value	2.850 mm	0.230 mm	0.713 mm
Standard deviation	1.080 mm	0.125 mm	0.283 mm
Error range	0.967~5.528 mm	0.032~0.565 mm	0.250~1.742 mm

Funding

This work was funded by the National Key Research and Development Program of China (No.2022YFE03170001), Guangdong Basic and Applied Basic Research Foundation (2021B1515120009, 2022A1515012004), and Department of Education of Guangdong in China (2021ZDJS083)

Conflicts of interest

S.W. certifies that he has no financial conflicts of interest (e.g., consultancies, stock ownership, equity interest, patent/licensing arrangements, etc.) in connection with this article. Y.H. certifies that he has no financial conflicts of interest (e.g., consultancies, stock ownership, equity interest, patent/licensing arrangements, etc.) in connection with this article. Y.M. certifies that he has no financial conflicts of interest (e.g., consultancies, stock ownership, equity interest, patent/licensing arrangements, etc.) in connection with this article. J.G. has received funding from the National Key Research and Development Program of China (No.2022YFE03170001), Guangdong Basic and Applied Basic Research Foundation (2021B1515120009, 2022A1515012004), and Department of Education of Guangdong in China (2021ZDJS083)

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contribution statement

Conceptualization, S.W. and J.G. ; Methodology, S.W. and J.G. ; Software, S.W. and Y.H.; Validation, S.W. and Y.M.; Formal Analysis, Y.H.; Investigation, S.W. and J.G. ; Resources, J.G. ; Data Curation, Y.H.; Writing – Original Draft Preparation, Y. H.; Writing – Review & Editing, S.W.; Visualization, Y.M.; Supervision, J.G. ; Project Administration, J.G. ; Funding Acquisition, J.G. ".

References

[1] M.M. Alam, S. Ibaraki, K. Fukuda, Kinematic modeling of six-axis industrial robot and its parameter identification: a tutorial, Int. J. Autom. Technol. **15**, 599–610 (2021)

[2] F. Guo, H. Cai, M. Ceccarelli, T. Li, B. Yao, Enhanced DH: an improved convention for establishing a robot link coordinate system fixed on the joint, Ind. Robot: Int. J. Robotics Res. Appl. **47**, 197–205 (2020)

[3] N. Zhao, S. Ibaraki, Calibration and compensation of rotary axis angular positioning deviations on a SCARA-type industrial robot using a laser tracker, in: Proceedings of International Conference on Leading Edge Manufacturing/ Materials and Processing, American Society of Mechanical Engineers, 2020, V001T02A001

[4] C. Faria, J.L. Vilaça, S. Monteiro, W. Erlhagen, E. Bicho, Automatic Denavit-Hartenberg parameter identification for serial manipulators, in: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2019, pp. 610–617

[5] Y. Cai, P. Yuan, Z. Shi, D. Chen, S. Cao, Application of universal kriging for calibrating offline-programming industrial robots, J. Intell. Robot. Syst. **94**, 339–348 (2019)

[6] Y. Zeng, W. Tian, W. Liao, Positional error similarity analysis for error compensation of industrial robots, Robot. Comput. Integr . Manuf. **42**, 113–120 (2016)

[7] P. Yuan, D. Chen, T. Wang, S. Cao, Y. Cai, L. Xue, A compensation method based on extreme learning machine to enhance absolute position accuracy for aviation drilling robot, Adv. Mech. Eng. **10**, 1687814018763411 (2018)

[8] H.Q. Cao, H.X. Nguyen, T.N.C. Tran, H.N. Tran, J.W. Jeon, A robot calibration method using a neural network based on a butterfly and flower pollination algorithm, IEEE Trans. Ind. Electron. **69**, 3865–3875 (2021)

[9] W. Wang, W. Tian, W. Liao, B. Li, J. Hu, Error compensation of industrial robot based on deep belief network and error similarity, Robot. Comput. Integr. Manuf. **73**, 102220 (2022)

[10] S. Ma, K. Deng, Y. Lu, X. Xu, Robot error compensation based on incremental extreme learning machines and an improved sparrow search algorithm, Int. J. Adv. Manuf. Technol. **125**, 5431–5443 (2023)

[11] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.R. Müller, Explaining deep neural networks and beyond: a review of methods and applications, Proc. IEEE **109**, 247–278 (2021)

[12] M. Dehghani, P. Trojovský, Osprey optimization algorithm: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems, Front. Mech. Eng. **8**, 1126450 (2023)

[13] R. Ribani, M. Marengoni, A survey of transfer learning for convolutional neural networks, in: 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), IEEE, 2019, pp. 47–57

[14] K. Weiss, T.M. Khoshgoftaar, D.D. Wang, A survey of transfer learning, J. Big data, **3**, 1–40 (2016)
[15] R. Schmidt, M. Voigt, R. Mailach, Latin hypercube sampling-based Monte Carlo simulation: extension of the sample size and correlation control, in: Uncertainty Management for Robust Industrial Design in Aeronautics: Findings and Best Practice Collected During UMRIDA, a Collaborative Research Project (2013-2016) Funded by the European Union, Springer International Publishing, Cham, 2018, pp. 279–289

Cite this article as: Shuai Wang, Yuanbao Huang, Yongkang Mo, Jianwen Guo, Precision compensation technology of industrial robot based on OOA-DNN and transfer learning, Mechanics & Industry **26**, 37 (2025), https://doi.org/10.1051/meca/2025031