

Research on a CNC Machine Tool Machining Error Analysis and Prediction Method Based on Digital Twin and ISSLO-CNN

Zelin Cao

The Higher Educational Key Laboratory for Flexible Manufacturing Equipment Integration of Fujian Province (Xiamen Institute of Technology)

Lihua Chen

`chenlihua52039@163.com`

Jimei University

Zhenbei Lin

The Higher Educational Key Laboratory for Flexible Manufacturing Equipment Integration of Fujian Province (Xiamen Institute of Technology)

Yibin Zhuo

Xiamen yangsen NC equipment Joint Stock Co., Ltd.

Article

Keywords: CNC machine tool, Digital twin, Convolutional neural network, Improved stochastic social learning optimization algorithm, Hyperparameter optimization, Machining error prediction

Posted Date: September 21st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10947358/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Research on a CNC Machine Tool Machining Error Analysis and Prediction Method Based on Digital Twin and ISSLO-CNN

Zelin Cao¹, Lihua Chen², Zhenbei Lin¹, Yibin Zhuo³

1.The Higher Educational Key Laboratory for Flexible Manufacturing Equipment Integration of Fujian Province (Xiamen Institute of Technology), Xiamen 361021, China; 2.Navigation College, Jimei University, Xiamen 361021, China; 3.Xiamen yangsen NC equipment Joint Stock Co., Ltd., Xiamen 361021, China.

Email: caoslin52055@163.com

Corresponding author: "Chen Lihua"

Email: chenlihua52039@163.com

Abstract: To address the insufficient hyperparameter optimization accuracy and low convergence efficiency of the stochastic social learning optimization algorithm (SSLO) in CNC machine tool machining error prediction, caused by the fixed distribution of learning factors, lack of elite guidance in learning targets, and static knowledge exchange rate, an improved stochastic social learning optimization algorithm (ISSLO) and digital twin-driven CNN error prediction model are proposed. First, physical measurement signals and virtual simulation signals are synchronously acquired based on a digital twin quadruple model, and a multisource fused feature matrix is generated using a linear fusion strategy as the input samples of the convolutional neural network (CNN). Subsequently, a new improved SSLO algorithm is designed. On the one hand, an elite-guided adaptive learning factor mechanism based on iterative cosine decay and fitness ranking is introduced to dynamically regulate the search step size and guide individuals toward elite solutions. On the other hand, an adaptive perturbation strategy based on historical successful-direction memory and a dynamic knowledge exchange rate adjustment mechanism based on the update success rate are constructed to enhance the improvement efficiency of disadvantaged individuals and information exchange within the population. Finally, ISSLO is employed to iteratively optimize key CNN hyperparameters, including convolution kernel size, number of kernels, learning rate, number of neurons in the fully connected layer, and batch size, and the CNN is trained using the optimal configuration to perform error prediction. Experimental results on the NCMT-DT-Error dataset show that ISSLO-CNN achieves a root mean square error of $3.84 \pm 0.16 \mu\text{m}$, a mean absolute error of $2.72 \pm 0.12 \mu\text{m}$, and a coefficient of determination of 0.979 ± 0.003 , while its physical consistency metrics comprehensively outperform those of comparison models such as SSLO-CNN. The proposed method has 1.97 M parameters and a single-sample inference time of 3.62 ms, and ablation experiments verify the effectiveness of the two improvement strategies. The results demonstrate that ISSLO-CNN significantly improves prediction accuracy while maintaining low complexity and real-time inference capability, effectively meeting the engineering requirements of online monitoring and compensation of CNC machine tool machining errors and providing a reliable technical approach for intelligent machining quality control.

Keywords: CNC machine tool; Digital twin; Convolutional neural network; Improved stochastic social learning optimization algorithm; Hyperparameter optimization; Machining error prediction.

1. Introduction

Machining errors in CNC machine tools are jointly affected by multiple sources, including thermal deformation, servo tracking errors, and tool wear. Complex coupling relationships exist

among different error sources, making traditional modeling methods based on a single error source or static mapping relationship inadequate for fully characterizing the dynamic evolution of errors during complex machining processes ^[1-3]. Digital twin technology enables dynamic mapping among physical entities, virtual models, and real-time operational data, thereby supporting online machining-state perception, error modeling and prediction, and feedback compensation ^[4-5]. Therefore, developing a machining error analysis method that integrates multisource digital twin data with intelligent prediction models is of great significance for improving the machining accuracy of CNC machine tools and enhancing intelligent quality control.

Regarding CNC machine tool machining error modeling and digital twin-based prediction, researchers have conducted studies on physical-virtual data fusion, online thermal error monitoring, and prediction model optimization. Lu, Q. et al. fused theoretical thermal errors obtained from a digital twin with prediction results generated by an LSTM using measured data; however, their study provided relatively limited integration of multisource information such as cutting force, vibration, and tool condition ^[6]. Sun, G. et al. developed a digital online monitoring system for thermally induced volumetric errors of CNC machine tools, achieving a prediction accuracy exceeding 95%; however, the method mainly focused on thermally induced volumetric errors and involved limited collaborative modeling of multiple machining error features ^[7]. Sa, G. et al. designed a dual-stage attention CNN-LSTM model and optimized temperature measurement-point placement using a genetic algorithm, maintaining high thermal error prediction accuracy while reducing the number of sensors; however, further improvement remains possible in the collaborative representation of physical measurement and virtual simulation data and the unified prediction of multiple machining errors ^[8]. Overall, existing digital twin-based error prediction studies can already realize physical-virtual information interaction and online prediction of specific error sources, but further coordinated optimization is still required among multisource heterogeneous data fusion, complex error feature extraction, and real-time prediction efficiency.

With the development of deep learning and swarm intelligence optimization algorithms, convolutional neural networks (Convolutional Neural Network, CNN) have increasingly been used to extract multisource features from machining processes, while intelligent optimization algorithms are employed to adjust prediction model parameters, gradually becoming an important approach for improving machining error prediction performance. Bai, W. proposed a machining error prediction and compensation system based on CNN and an improved genetic algorithm, using the improved genetic algorithm to optimize compensation parameters and achieving favorable machining error prediction and compensation performance; however, the adaptive joint optimization of the CNN convolutional architecture and training hyperparameters remained limited ^[9]. Zhao, R. et al. constructed a mechanism–data hybrid-driven digital twin model for the feed system of a CNC machine tool and employed WOA-CNN-LSTM-Attention to compensate for residual errors of the mechanism model; however, the whale optimization algorithm may still be limited by search direction and local exploitation capability in complex high-dimensional hyperparameter spaces ^[10]. Ye, J. et al. proposed the Stochastic Social Learning Optimization (SSLO) algorithm, which demonstrated favorable optimization performance on benchmark functions, engineering optimization, and feature selection problems; however, its learning factors and knowledge exchange process did not fully exploit elite-individual guidance and historical information on successful search directions ^[11]. Therefore, integrating multisource digital twin

data fusion, CNN-based machining error prediction, and an improved social learning optimization mechanism is expected to further enhance the accuracy, stability, and adaptability of machining error prediction under complex operating conditions.

To address the above issues, this study proposes a digital twin-driven ISSLO-CNN method for CNC machine tool machining error analysis and prediction. First, a digital twin quadruple-based data acquisition and fusion framework is constructed to synchronously organize physical measurement signals, virtual simulation signals, and machining-process state information, and a multisource feature matrix is generated through linear fusion. Second, based on the original SSLO search mechanism, an elite-guided adaptive learning factor mechanism, a historical successful-direction memory perturbation strategy, and a dynamic knowledge exchange rate adjustment mechanism are designed to enhance the global exploration and local exploitation capabilities of the algorithm. Finally, ISSLO is employed to jointly optimize key CNN parameters, including convolution kernel size, number of convolution kernels, learning rate, number of neurons in the fully connected layer, and batch size. Comparative experiments, ablation experiments, and physical consistency analyses are conducted on the NCMT-DT-Error dataset to verify the effectiveness of the proposed method.

2. Methods

2.1 SSLO algorithm

The Stochastic Social Learning Optimization (SSLO) is a novel metaheuristic algorithm that integrates social learning theory with the barrel effect [12-13]. By simulating the complete cognitive chain of “imitation learning-inspiration burst-weak-individual reinforcement-knowledge exchange” in the process of human knowledge acquisition, SSLO constructs an optimization framework that balances global exploration and local exploitation. The computational steps of SSLO are as follows:

First, SSLO randomly and uniformly initializes candidate solutions within the search space, simulating the initial cognitive states of individuals before the learning process begins. The position component of the i -th candidate solution X_i in the j -th dimension is given by Eq. (1):

$$x_{i,j} = l_j + u_{i,j} \cdot (u_j - l_j), i = 1, 2, \dots, N, j = 1, 2, \dots, D \quad (1)$$

where $x_{i,j}$ denotes the position component of the i -th candidate solution in the j -th dimension;

l_j and u_j denote the lower and upper bounds of the j -th-dimensional search space, respectively;

$u_{i,j}$ denotes a uniformly distributed random number in the interval $[0, 1]$; N denotes the total number of candidate solutions (population size); and D denotes the dimensionality of the search space.

After initialization, the algorithm enters the knowledge accumulation and inspiration burst stage. Inspired by the exponential-tail exploration kernel in quantum-behaved particle swarm optimization, a closed-form sampling step is introduced into the social learning framework. During the conventional learning process, individuals gradually accumulate knowledge by imitating high-quality solutions; when random sampling generates a relatively large step size, an inspiration burst is triggered, enabling cross-domain cognitive breakthroughs. For the i -th

individual, the position update strategy is given by:

$$X_i^{t+1} = \begin{cases} X_i^t + \xi_i^t \square (X_k^t - X_i^t), & \text{if } f(X_k^t) < f(X_i^t) \\ X_i^t + \xi_i^t \square (X_i^t - X_k^t), & \text{otherwise} \end{cases} \quad (2)$$

where X_i^t denotes the position vector of the i -th individual at the t -th iteration; X_k^t denotes the position vector of another individual randomly selected from the current population ($k \neq i$) at the t -th iteration; $f(\cdot)$ denotes the fitness function, and this study considers a minimization problem; $\square$ denotes the Hadamard product (element-wise multiplication).

When the randomly selected individual X_k^t has a better fitness value (i.e., $f(X_k^t) < f(X_i^t)$), X_i^t moves toward X_k^t to absorb its knowledge; otherwise, it moves away from X_k^t to maintain cognitive independence. ξ_i^t denotes a D -dimensional random learning-factor vector, whose j -th component is defined as:

$$\xi_{i,j}^t = \frac{1}{r_{i,j}^t}, r_{i,j}^t \sim U(0,1) \quad (3)$$

where $r_{i,j}^t$ denotes a uniformly distributed random number in the interval (0, 1).

As shown in Eq. (3), $\xi_{i,j}^t$ follows a reciprocal distribution with the probability density function $p(\xi) = \xi^{-2}$ ($\xi > 1$), exhibiting a typical exponential-tail characteristic. Specifically, $\xi_{i,j}^t$ falls within the interval (1, 2] with a probability of approximately 50%, producing frequent small step sizes to support fine-grained local search. Meanwhile, there is approximately a 25.44% probability that $\xi_{i,j}^t$ falls within the interval (2, 4], enabling moderate exploration with medium step sizes. Low-probability events with $\xi_{i,j}^t > 4$ correspond to inspiration bursts, causing individuals to make large-scale jumps and thereby break through local cognitive boundaries. This adaptive step-size distribution mechanism enables the algorithm to maintain a dynamic balance between exploration and exploitation without requiring additional parameter tuning.

On this basis, the algorithm introduces the barrel-effect mechanism to perform targeted reinforcement of the weakest individual in the population. The barrel effect states that the overall performance of a system is constrained by its weakest component; therefore, improving the knowledge level of the weakest individual is crucial to the overall evolution of the population. Let

X_w^t denote the position vector of the individual with the worst fitness at the t -th iteration. Its update strategy is divided into two sub-stages according to the iteration process. In the early stage

of iteration ($t < 0.5T$), the weakest individual actively searches for the most suitable learning mode through self-exploration or a reflection mechanism, thereby gradually improving its knowledge level. Its position update formula is given by:

$$X_w^{t+1} = X_w^t + \alpha_w^t \square D_w^t \quad (4)$$

$$D_w^t = \text{sign}(r_w^t) \square |X_w^t| \quad (5)$$

where X_w^{t+1} denotes the updated position vector of the weakest individual; D_w^t denotes the exploration direction vector, constructed by the Hadamard product of the sign function $\text{sign}(\cdot)$ and the absolute value $|X_w^t|$ of the current solution; r_w^t denotes a D -dimensional random vector whose components take +1 or -1 with equal probability, determining the update direction in each dimension; and α_w^t denotes a D -dimensional dynamically adjusted parameter vector, whose j -th component is defined as:

$$\alpha_{w,j}^t = \exp\left(-\frac{t}{T}\right) \cdot \exp\left(\frac{F_w^t}{1+F_w^t}\right) \cdot |z_{w,j}^t| \quad (6)$$

where t denotes the current iteration number; T denotes the maximum number of iterations; F_w^t denotes the cumulative number of failed updates of the weakest individual up to the t -th iteration; and $z_{w,j}^t$ denotes a random number sampled from the standard normal distribution $N(0, 1)$.

$\exp\left(-\frac{t}{T}\right)$ denotes an exponential decay control factor that gradually reduces the search step size as the iteration proceeds, thereby ensuring convergence accuracy in the later stage of the algorithm. $\exp\left(\frac{F_w^t}{1+F_w^t}\right)$ denotes a “vitality factor”, which automatically increases the step size

when the weakest individual fails to update consecutively, thereby enhancing its ability to escape from local optima.

In the later stage of iteration ($t > 0.5T$), the weakest individual instead learns from the current best individual X_b^t in the population and uses the newly acquired knowledge to perform deep exploitation of the search space:

$$X_w^{t+1} = X_b^t + \alpha_w^t \square (X_b^t - X_w^t) \quad (7)$$

where X_b^t denotes the position vector of the individual with the best fitness at the t -th iteration.

This mechanism enables the weakest individual to use the best individual as a reference anchor, contracting toward the vicinity of the optimal solution while introducing random

perturbations, thereby maintaining a certain degree of exploration capability while ensuring convergence speed. If the fitness of the weakest individual fails to improve over L consecutive iterations, it indicates that the current learning strategy has become ineffective. In this case, the algorithm introduces an unconventional learning strategy to break cognitive stagnation, namely by applying relatively large random perturbations to each dimension of X_w^t so that it can escape from the current stagnant region.

After the above updates, some dimensions of the newly generated candidate solutions may exceed the predefined boundary range. To maintain solution feasibility, SSLO adopts a midpoint-replacement boundary handling strategy to correct the out-of-bound components:

$$x_{i,j} = \begin{cases} \frac{l_j + u_j}{2}, & \text{if } x_{i,j} < l_j \text{ or } x_{i,j} > u_j \\ x_{i,j}, & \text{otherwise} \end{cases} \quad (8)$$

where $x_{i,j}$ denotes the j -th-dimensional component of any candidate solution; l_j and u_j denote the lower and upper bounds of that dimension, respectively. This strategy pulls out-of-bound components back toward the center of the feasible region rather than simply clipping them to the boundary, thereby correcting boundary violations without significantly distorting the distribution structure of the solutions and effectively maintaining population diversity.

Subsequently, the algorithm enters the knowledge exchange and innovation stage. At this stage, each individual contributes unique information based on its own knowledge reserve, and information exchange among individuals is achieved through a crossover-mask mechanism, thereby exploring the solution space from a global perspective and reducing cognitive bias. For the i -th individual, a D -dimensional crossover-mask vector M_i is first generated, whose j -th component is controlled by the knowledge exchange rate Cr_i :

$$M_{i,j} = \begin{cases} 1, & \text{if } r_{i,j} < Cr_i \text{ or } j = j_{rand} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where $r_{i,j}$ denotes a uniformly distributed random number in the interval $[0, 1]$; Cr_i denotes the knowledge exchange rate of the i -th individual, defined as $Cr_i = 0.5 + 0.5 \cdot u_i$, where $u_i \sim U(0,1)$; j_{rand} denotes a dimension index randomly selected from $\{1, 2, \dots, D\}$, ensuring that knowledge exchange occurs in at least one dimension.

When $M_{i,j} = 1$, the crossover operation is performed in that dimension; when $M_{i,j} = 0$, the dimension remains unchanged. Based on the crossover mask, another individual X_k^t is randomly selected from the population as the knowledge donor, and the updated position of the i -th individual is given by:

$$X_i^{new} = M_i \square X_k^t + (1 - M_i) \square X_i^t \quad (10)$$

where X_i^{new} denotes the candidate solution after knowledge exchange; $\mathbf{1}$ denotes an all-ones vector; and $(1 - M_i)$ denotes the logical complement of the crossover mask, representing the set of dimensions in which the individual retains its own information. This operation enables X_i to absorb knowledge from X_k in selected dimensions while preserving its own characteristics in the remaining dimensions, thereby achieving local recombination and innovation of knowledge.

Finally, the algorithm determines the solution inherited by each individual for the next generation through a selection strategy and dynamically tracks the number of failed updates. Specifically, the candidate solution X_i^{new} obtained after knowledge exchange is compared with the current solution X_i^t in terms of fitness:

$$X_i^{t+1} = \begin{cases} X_i^{new}, & \text{if } f(X_i^{new}) < f(X_i^t) \\ X_i^t, & \text{otherwise} \end{cases} \quad (11)$$

Meanwhile, the number of failed updates F_i^{t+1} is accumulated according to the following rule:

$$F_i^{t+1} = \begin{cases} F_i^t + 1, & \text{if } f(X_i^{new}) \geq f(X_i^t) \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where F_i^t denotes the cumulative number of failed updates of the i -th individual at the t -th iteration. When an update failure occurs, the algorithm automatically resets the knowledge exchange rate Cr_i of that individual to the value corresponding to its most recent successful update, thereby enabling dynamic adaptive adjustment of the learning strategy.

Through the complete iterative process of “initialization–knowledge accumulation and inspiration burst–barrel-effect reinforcement–boundary handling–knowledge exchange and innovation–evaluation and selection”, SSLO achieves efficient exploitation and exploration of the search space while maintaining population diversity. The computational flow of SSLO is shown in Fig. 1.

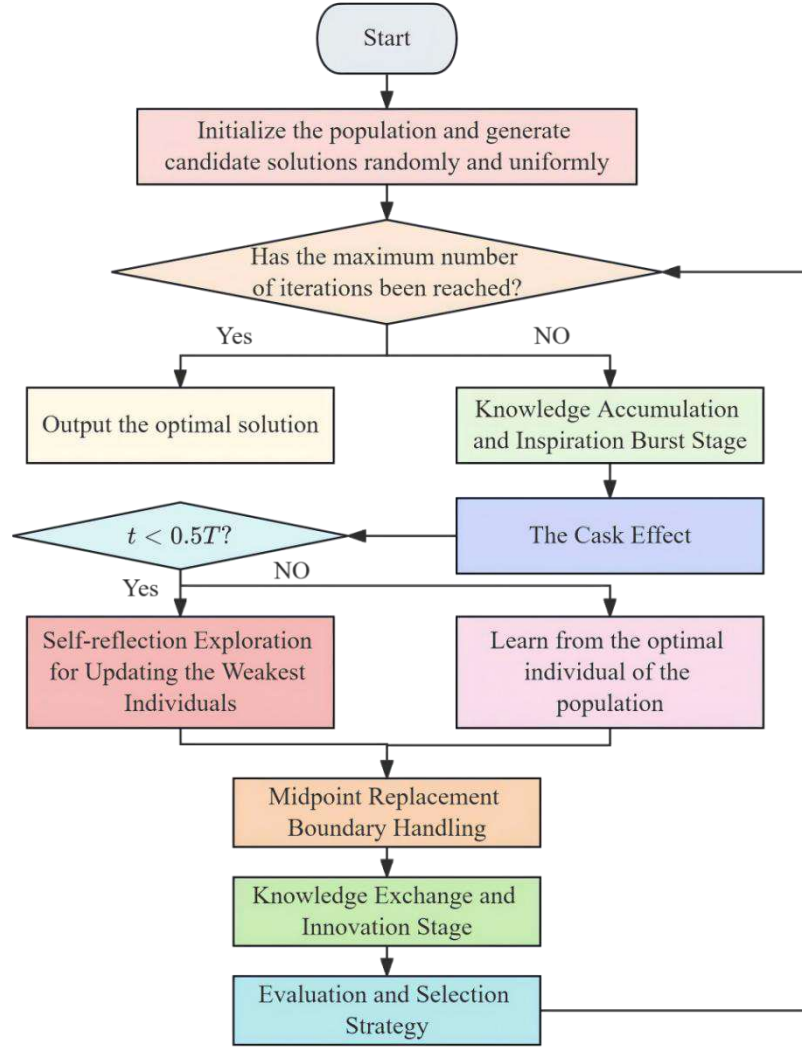


Fig. 1 Computational flowchart of SSLO algorithm

2.2 Limitations of SSLO in CNC machine tool machining error analysis and prediction

Although SSLO introduces a reciprocal-distribution-based learning factor and a barrel-effect reinforcement mechanism to achieve, to some extent, a dynamic balance between global exploration and local exploitation, a closer examination of its iterative update logic reveals that the algorithm still has two structural limitations when applied to complex high-dimensional hyperparameter optimization problems. These limitations directly restrict its solution accuracy and convergence efficiency, as analyzed below ^[14-15].

Limitation 1: The learning-factor distribution is fixed, and the selection of learning targets lacks elite guidance, resulting in insufficient convergence accuracy in the later stages.

During the knowledge accumulation and inspiration burst stage of SSLO, the learning factor $\xi_{i,j}^t$ follows a fixed reciprocal distribution with the probability density function $p(\xi) = \xi^{-2}$. Although this distribution exhibits an exponential-tail characteristic and can provide strong global exploration capability in the early iterations, its distribution form remains unchanged throughout the entire iterative process. Consequently, large-scale jumps may still occur with a non-negligible probability in the later stages. Specifically, the probability of $\xi_{i,j}^t > 4$ is approximately 25%, meaning that even when the algorithm has approached the global optimum region, individuals may still jump out of the current neighborhood with a large step size, causing persistent oscillations around the optimal solution and significantly reducing convergence accuracy. Meanwhile, the learning target X_k^t in Eq. (2) is selected completely at random from the current

population, without utilizing any fitness-ranking information or prior knowledge from elite solutions. For complex multimodal problems, such an indiscriminate random learning strategy can easily lead to a large amount of inefficient or even detrimental knowledge transfer, weakening the directional search capability of the algorithm and reducing its convergence speed.

Limitation 2: The reinforcement strategy for the weakest individual is somewhat blind, and the knowledge exchange rate lacks an adaptive adjustment mechanism.

During the barrel-effect stage of SSLO, the direction vector D_w^t in Eq. (5) during the early iterations is constructed entirely from the Hadamard product of the random sign vector $\text{sign}(r_w^t)$ and the absolute value $|X_w^t|$ of the current solution, without utilizing directional information from successful updates of the individual in previous iterations. This purely random reflection-based search strategy makes the improvement direction of the weakest individual highly uncertain, making it difficult to achieve effective knowledge accumulation. Although Eq. (7) in the later iterations guides the weakest individual to learn from the global best individual X_b^t , if X_b^t has already become trapped in a local optimum, the weakest individual will also be drawn into the same local basin of attraction, losing its ability to escape and further accelerating the loss of population diversity. In addition, during the knowledge exchange and innovation stage, the knowledge exchange rate Cr_i of each individual is randomly generated at initialization and then remains essentially unchanged, except that it is restored to the value corresponding to the most recent successful update when an update fails. This static adjustment mechanism fails to dynamically regulate the information exchange intensity according to the individual's current evolutionary state, such as the history of consecutive successes or failures. As a result, some individuals may maintain excessively low or high crossover rates for extended periods. The former limits knowledge acquisition, whereas the latter may disrupt favorable genetic structures, thereby reducing the overall efficiency of knowledge exchange and the ability to maintain population diversity.

To visually illustrate the two structural limitations of SSLO, Fig. 2 presents the corresponding schematic diagrams. Fig. 2(a) illustrates the persistent large-scale jumps caused by the fixed reciprocal distribution of the learning factor, while Fig. 2(b) illustrates the blindness of the weakest-individual reinforcement strategy and the lack of an adaptive adjustment mechanism for the knowledge exchange rate.

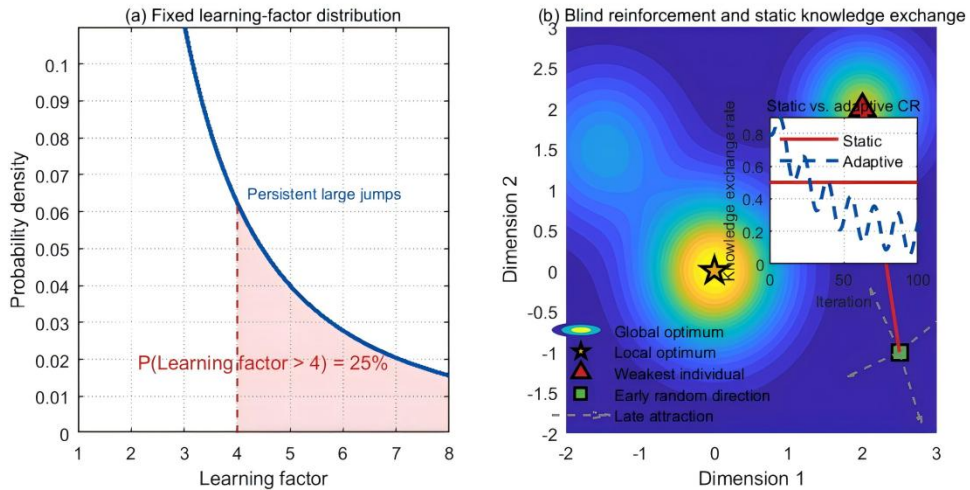


Fig. 2 Schematic illustration of the limitations of SSLO algorithm

2.3 Improved SSLO algorithm

To address the structural limitations of SSLO identified in Section 2.2, including the learning-factor distribution and learning-target selection, the weakest-individual reinforcement strategy, and the adjustment of the knowledge exchange rate, this study designs an elite-guided adaptive learning-factor mechanism and an improved strategy integrating historical successful-memory information with an adaptive knowledge exchange rate. The original algorithm

is systematically reconstructed in terms of search-step regulation, learning-direction guidance, improvement of the evolutionary efficiency of disadvantaged individuals, and optimization of population information exchange quality, thereby proposing an Improved Stochastic Social Learning Optimization algorithm (ISSLO).

Improvement 1: elite-guided adaptive learning-factor mechanism.

To address the problems identified in limitation 1, namely the fixed learning-factor distribution and random selection of learning targets, this study designs an adaptive learning-factor regulation strategy that integrates iteration-progress decay with individual fitness ranking, and introduces an elite-guided learning-target selection mechanism to enhance local exploitation accuracy in the later stages and improve overall search efficiency.

First, the learning factor $\xi_{i,j}^t$ in Eq. (3) is dynamically scaled by introducing an adaptive adjustment function λ_i^t :

$$\xi_{i,j}^t = \frac{\lambda_i^t}{r_{i,j}^t}, r_{i,j}^t \sim U(0,1) \quad (13)$$

$$\lambda_i^t = \lambda_{\min} + (\lambda_{\max} - \lambda_{\min}) \cdot \frac{1 + \cos\left(\frac{\pi t}{T}\right)}{2} \cdot \left(1 - \frac{R_i^t}{N}\right)^\beta \quad (14)$$

where λ_i^t denotes the learning-factor scaling coefficient of the i -th individual at the t -th iteration; $\lambda_{\max}$ and $\lambda_{\min}$ denote the upper and lower bounds of the scaling coefficient, respectively, satisfying $\lambda_{\max} > \lambda_{\min} > 0$; t denotes the current iteration number; T denotes the maximum number of iterations; R_i^t denotes the rank of the i -th individual at the t -th iteration after sorting the fitness values in ascending order, where the best individual is ranked 1 and the worst individual is ranked N ; N denotes the population size; and β denotes the rank influence exponent, which controls the strength of the effect of fitness ranking on the scaling coefficient.

In Eq. (14), the cosine annealing term $\frac{1 + \cos\left(\frac{\pi t}{T}\right)}{2}$ causes the scaling coefficient to smoothly decay from $\lambda_{\max}$ to $\lambda_{\min}$ as the iterations proceed, thereby gradually reducing the learning step size and enhancing local search capability in the later stages. The ranking term $\left(1 - \frac{R_i^t}{N}\right)^\beta$ assigns smaller scaling coefficients to individuals with better fitness for fine-grained exploitation, while individuals with poorer fitness retain larger scaling coefficients to preserve the necessary exploration capability. Through these two adjustment mechanisms, the algorithm achieves differentiated and dynamic control of the learning step size at both the individual and iteration levels.

Subsequently, the selection of the random learning target X_k^t in Eq. (2) is changed from completely random selection to elite-guided selection. Let E^t denote the elite subset at the t -th iteration, consisting of the top $\lceil \gamma N \rceil$ individuals in the current population ranked by fitness, where $\gamma \in (0,1]$ denotes the elite proportion parameter. The selection rule for the learning target X_k^t is given by:

$$X_k^t = \begin{cases} \text{Randomly selected from } E^t, \text{ if } u_e < p_e \\ \text{Randomly selected from the entire population,} \\ \text{otherwise} \end{cases} \quad (15)$$

where u_e denotes a uniformly distributed random number in the interval $[0,1]$; $p_e \in [0,1]$ denotes the elite-selection probability. When $u_e < p_e$, the individual learns from elite solutions with a higher probability, thereby accelerating the dissemination of high-quality knowledge; otherwise, the learning target is still randomly selected from the entire population to preserve a certain degree of exploratory diversity and avoid excessive convergence.

This mechanism achieves a controllable trade-off between elite guidance and population diversity.

Improvement 2: integration of historical successful-memory information and adaptive knowledge exchange rate.

To address the blindness of the barrel-effect reinforcement strategy and the static knowledge exchange rate identified in limitation 2, this study introduces a historical successful-direction memory mechanism to reconstruct the update direction of the weakest individual and designs an adaptive knowledge exchange rate adjustment strategy based on the individual update success rate, thereby improving the evolutionary efficiency of disadvantaged individuals and the quality of information exchange within the population.

First, a historical successful-direction vector S_w^t is introduced during the barrel-effect stage to record the directional information associated with successful updates of the weakest individual in previous iterations. This vector is accumulated and updated using an exponentially weighted moving average:

$$S_w^{t+1} = \rho S_w^t + (1 - \rho) \cdot \text{sign}(X_w^{t+1} - X_w^t) \quad (16)$$

where S_w^t denotes the historical successful-direction vector of the weakest individual at the t -th iteration and is initialized as a zero vector; $\rho \in [0,1)$ denotes the memory decay factor, which controls the retention of historical directional information during updating; and $\text{sign}(X_w^{t+1} - X_w^t)$ denotes the directional sign of each dimension in the most recent successful update. When ρ is relatively large, more historical directional information is retained, resulting in greater continuity in the search direction; when ρ is relatively small, the algorithm places greater emphasis on the direction of the most recent successful update.

Subsequently, the direction vector D_w^t in Eq. (5) during the early iteration stage is reconstructed as:

$$D_w^t = \text{sign}(r_w^t) \square |X_w^t| + \eta S_w^t \quad (17)$$

where η denotes the historical-direction weighting coefficient used to balance the contributions of the random exploration direction and the historical successful direction. A larger η causes the weakest individual to search more strongly along historically successful directions, thereby reducing blindness; a smaller η preserves greater random exploration capability. By introducing S_w^t , the weakest individual can exploit its accumulated successful experience for directed improvement during the early iteration stage, significantly enhancing the search efficiency of the barrel-effect stage.

In the later stage of iteration, to prevent the weakest individual from becoming trapped in the same local optimum region as the global best individual X_b^t , an adaptive perturbation term is introduced on the basis of Eq. (7):

$$X_w^{t+1} = X_b^t + \alpha_w^t \square (X_b^t - X_w^t) + \delta_w^t \square \in \quad (18)$$

$$\delta_w^t = \delta_0 \cdot \left[1 + \frac{F_w^t}{1 + F_w^t} \right] \quad (19)$$

where ϵ denotes a D -dimensional standard normal random vector; δ_w^t denotes the adaptive perturbation amplitude coefficient; δ_0 denotes the baseline perturbation amplitude; and F_w^t denotes the cumulative number of failed updates of the weakest individual up to the t -th iteration. When the weakest individual fails to update consecutively, F_w^t increases, causing δ_w^t to increase accordingly and impose stronger perturbations to help it escape from the stagnant region. When an update succeeds, F_w^t is reset to zero and the perturbation amplitude returns to the baseline level, thereby ensuring convergence stability.

Finally, to address the static nature of the knowledge exchange rate Cr_i , an adaptive adjustment mechanism based on the individual update success rate is designed. Let SR_i^t denote the update success rate of the i -th individual over the most recent W iterations, defined as:

$$SR_i^t = \frac{1}{W} \sum_{\tau=t-W+1}^t I_i^\tau \quad (20)$$

where W denotes the sliding-window length; I_i^τ denotes the indicator function, which takes a value of 1 when the i -th individual is successfully updated at the τ -th iteration (i.e., $f(X_i^{\tau+1}) < f(X_i^\tau)$), and 0 otherwise.

The iterative update formula for the knowledge exchange rate is given by:

$$Cr_i^{t+1} = \text{clip}(Cr_i^t + \Delta Cr_i^t, Cr_{\min}, Cr_{\max}) \quad (21)$$

$$\Delta Cr_i^t = \begin{cases} \kappa_1 \cdot (1 - SR_i^t), & \text{if } SR_i^t < \theta \\ -\kappa_2 \cdot SR_i^t, & \text{otherwise} \end{cases} \quad (22)$$

where $Cr_{\min}$ and $Cr_{\max}$ denote the lower and upper bounds of the knowledge exchange rate, respectively, preventing Cr_i from exceeding the prescribed range; κ_1 and κ_2 denote the adjustment step sizes that control the variation magnitude of Cr_i ; and θ denotes the success-rate threshold. When the update success rate SR_i^t of an individual is lower than the threshold θ , it indicates that the individual's current knowledge structure is difficult to improve effectively. In this case, its knowledge exchange rate Cr_i is increased to promote the acquisition of external knowledge through information exchange. When SR_i^t is higher than the threshold θ , it indicates that the individual is evolving well, and Cr_i is appropriately reduced to preserve its favorable genetic structure. $\text{clip}(\cdot)$ denotes the clipping function, which constrains Cr_i^{t+1} within the interval $[Cr_{\min}, Cr_{\max}]$.

This mechanism enables the knowledge exchange rate to be adaptively adjusted in real time according to the individual's recent evolutionary performance, thereby significantly improving the targeting and efficiency of knowledge exchange.

2.4 Error analysis and prediction system model based on digital twin and ISSLO-CNN

The CNC machine tool machining error analysis and prediction system developed in this study uses digital twin technology as the data foundation, a convolutional neural network (CNN) as the core predictor, and the ISSLO algorithm to optimize key CNN hyperparameters. The overall system architecture consists of three modules: a digital twin data generation and fusion module, a

CNN error prediction module, and an ISSLO hyperparameter optimization module. These three modules form a closed-loop iterative structure.

2.4.1 Digital twin data generation and fusion model

A digital twin is a high-fidelity mapping of a physical CNC machine tool in virtual space and can be formally represented as a quadruple:

$$DT = \{G, P, B, R\} \quad (23)$$

where G denotes the geometric model, describing the structural dimensions and assembly relationships of the machine tool; P denotes the physical model, characterizing the mathematical representations of physical processes such as cutting force, thermal deformation, and vibration; B denotes the behavioral model, reflecting the dynamic response characteristics of the machine tool under different operating conditions; and R denotes the rule model, including machining process parameters, expert knowledge, and constraint conditions. Driven by multidomain simulation and real-time sensor data, the digital twin can output simulated signals synchronized with the physical entity.

Let $s_t \in R^{d_s}$ denote the multisource sensor signal vector collected from the physical machine tool at time t , including vibration amplitude, spindle temperature, cutting force components, and other measurements, and let $\hat{s}_t \in R^{d_s}$ denote the simulation signal vector output by the digital twin at the corresponding time. To fully exploit the complementarity between physical measurement data and virtual simulation data, a linear fusion strategy is adopted to generate the feature vector for error analysis:

$$z_t = W_f [s_t; \hat{s}_t] + b_f \quad (24)$$

where $z_t \in R^{d_z}$ denotes the fused feature vector; $[s_t; \hat{s}_t] \in R^{2d_s}$ denotes the concatenated vector of the measured and simulated signals; $W_f \in R^{d_z \times 2d_s}$ denotes the fusion weight matrix; and $b_f \in R^{d_z}$ denotes the fusion bias vector. The fusion weights can be adaptively adjusted through end-to-end training or based on digital twin confidence to reflect the differentiated contributions of measured and simulated signals to error formation.

The fused feature vectors within a continuous time window are organized into a two-dimensional feature matrix as the input to the CNN:

$$X = [z_{t-m+1}, z_{t-m+2}, \dots, z_t]^T \in R^{m \times d_z} \quad (25)$$

where m denotes the time-window length, i.e., the number of historical time steps contained in each input. Each row of X corresponds to the fused feature vector at one time step, while each column represents the evolution sequence of one feature dimension within the time window.

The structural schematic of the digital twin data generation and fusion model is shown in Fig. 3.

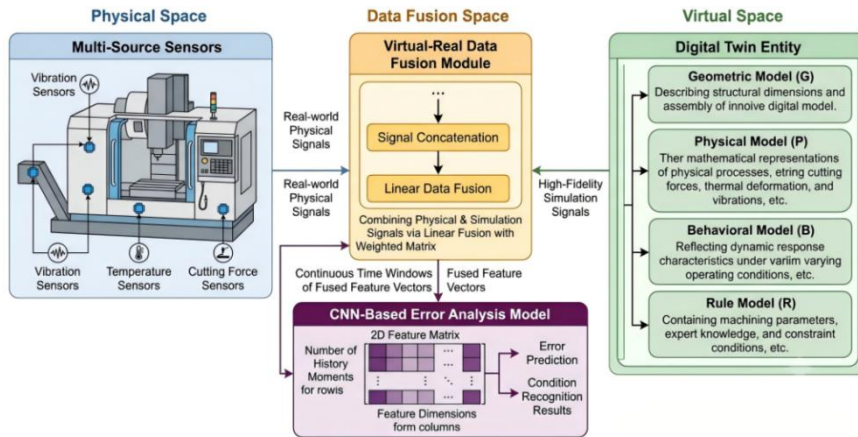


Fig. 3 Structural schematic of the digital twin data generation and fusion model

2.4.2 CNN error prediction model

Through local receptive fields and weight sharing, CNN can effectively extract spatiotemporal correlation features from X , thereby enabling accurate prediction of machining errors. In this study, a cascaded structure consisting of L convolutional layers, L pooling layers, and several fully connected layers is adopted. Let $H^{(l-1)} \in R^{H_{l-1} \times W_{l-1} \times C_{l-1}}$ denote the input feature map of the l th convolutional layer; then, its output feature map $H^{(l)}$ is given by:

$$H^{(l)} = \sigma(W^{(l)} * H^{(l-1)} + b^{(l)}) \quad (26)$$

where $W^{(l)} \in R^{k_l \times k_l \times C_{l-1} \times C_l}$ denotes the convolution kernel tensor of the l -th layer, where k_l denotes the spatial size of the convolution kernel; C_l denotes the number of output channels, i.e., the number of convolution kernels; $*$ denotes the two-dimensional convolution operation; $b^{(l)} \in R^{C_l}$ denotes the bias vector; and $\sigma(\cdot)$ denotes the nonlinear activation function, with the ReLU function $\sigma(x) = \max(0, x)$ adopted in this study. The convolution operation extracts local spatial features from the input feature map through a sliding window, while the C_l convolution kernels capture different types of feature patterns.

The pooling layer follows the convolutional layer and is used to reduce the spatial resolution of the feature map while enhancing translation invariance. In this study, max pooling is adopted:

$$P^{(l)} = \text{MaxPool}(H^{(l)}, p_l, s_l) \quad (27)$$

where p_l denotes the pooling window size and s_l denotes the pooling stride. Max pooling selects the maximum value within each $p_l \times p_l$ window as the output, thereby reducing computational cost while preserving salient features.

After L convolution-pooling operations, the feature map output by the final pooling layer is flattened into a one-dimensional vector $h^{(L)} \in R^{d_f}$ and fed into the fully connected layer for nonlinear mapping, ultimately producing the predicted machining error value:

$$\hat{y} = W^{(fc)} h^{(L)} + b^{(fc)} \quad (28)$$

where $\hat{y}$ denotes the predicted machining error, such as dimensional error or shape error; $W^{(fc)} \in R^{1 \times d_f}$ denotes the weight vector of the fully connected layer; and $b^{(fc)}$ denotes the bias scalar. If multiple error indicators need to be predicted simultaneously, the output dimension can be extended to q , where $\hat{y} \in R^q$ and $W^{(fc)} \in R^{q \times d_f}$.

The training objective of the CNN model is to minimize the mean squared error loss function between the predicted error and the actual error:

$$L(\Theta) = \frac{1}{M} \sum_{m=1}^M (y_m - \hat{y}_m)^2 + \lambda \|\Theta\|_2^2 \quad (29)$$

where Θ denotes the set of all trainable parameters in the CNN, including convolution kernel weights, biases, and fully connected layer parameters; M denotes the number of training samples; y_m and $\hat{y}_m$ denote the actual machining error and predicted machining error of the m -th sample, respectively; and λ denotes the regularization coefficient used to suppress overfitting.

The structural schematic of the CNN model is shown in Fig. 4.

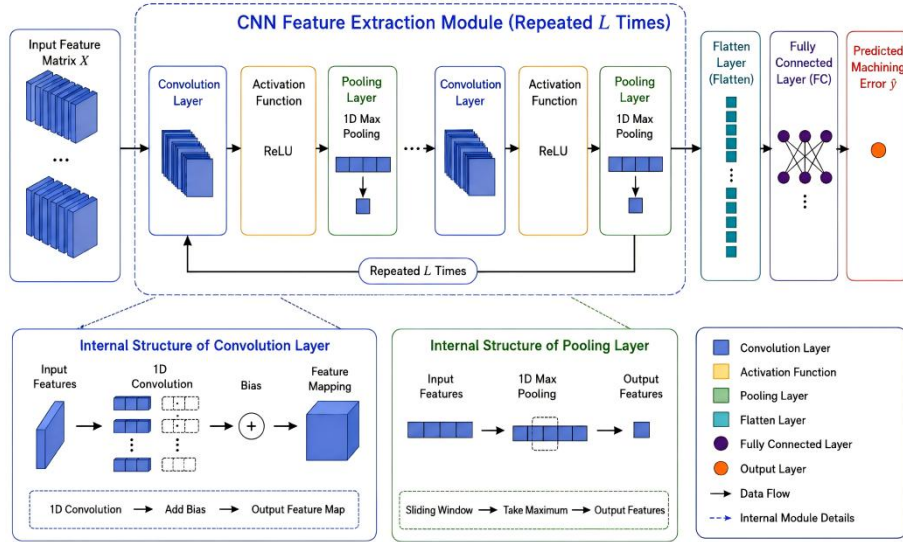


Fig. 4 Structural schematic of the CNN model

2.4.3 Hyperparameter optimization of the CNN model using ISSLO

The prediction performance of the CNN model is highly dependent on its hyperparameter configuration, including convolution kernel size, number of convolution kernels, learning rate, number of neurons in the fully connected layer, and batch size. Complex coupling relationships exist among these hyperparameters, making it difficult to obtain a globally optimal combination through manual tuning. Therefore, the ISSLO algorithm is employed to automatically optimize the key hyperparameters of the CNN. The hyperparameters to be optimized are encoded into a candidate solution vector:

$$X = [k_1, n_1, k_2, n_2, \dots, k_L, n_L, \eta_{lr}, n_{fc}, B_s] \quad (30)$$

where k_l denotes the convolution kernel size of the l -th convolutional layer, selected from a predefined discrete set such as $\{3, 5, 7\}$; n_l denotes the number of convolution kernels in the l -th convolutional layer, selected from a predefined integer range; η_{lr} denotes the learning rate for CNN training; n_{fc} denotes the number of neurons in the fully connected layer; and B_s denotes the batch size. The search space is jointly determined by the upper and lower bounds and the predefined discrete value sets of the hyperparameters.

In each ISSLO iteration, the candidate solution X is decoded into a specific CNN architecture configuration, and the CNN is trained on the training set. The fitness value is then calculated on the validation set. The fitness function is defined as the sum of the root mean square error on the validation set and a model complexity penalty term:

$$f(X) = \sqrt{\frac{1}{M_{val}} \sum_{m=1}^{M_{val}} (y_m - \hat{y}_m(X))^2} + \lambda_c \cdot C(X) \quad (31)$$

where M_{val} denotes the number of validation samples; $\hat{y}_m(X)$ denotes the prediction of the CNN for the m -th validation sample under hyperparameter configuration X ; $C(X)$ denotes the model complexity measure, which can be defined as the weighted sum of the total number of convolution kernels and the number of neurons in the fully connected layer, i.e., $C(X) = \sum_{l=1}^L n_l + n_{fc}$; and λ_c denotes the complexity penalty coefficient used to balance prediction accuracy and model size. By minimizing $f(X)$, ISSLO suppresses excessive model complexity while maintaining prediction accuracy.

The computational flow of ISSLO for optimizing the hyperparameters of the CNN model is shown in Fig. 5.

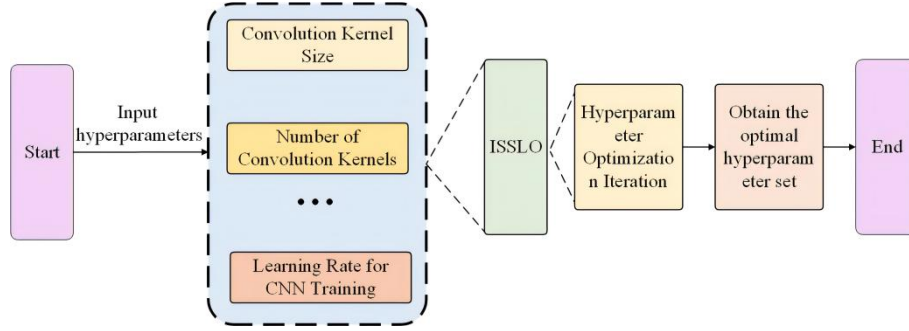


Fig. 5 Computational flow of ISSLO for optimizing CNN hyperparameters

In summary, the proposed error analysis and prediction method based on digital twin and ISSLO-CNN uses digital twin fused features as the input, CNN as the predictor, and ISSLO as the hyperparameter optimizer. The three components work collaboratively to achieve high-accuracy and adaptive prediction of CNC machine tool machining errors. The hyperparameters optimized by ISSLO specifically include the convolution kernel size k_l and number of convolution kernels n_l of each convolutional layer, the CNN training learning rate η_{lr} , the number of neurons in the fully connected layer n_{fc} , and the batch size B_s .

2.5 CNC machine tool machining error analysis and prediction method based on digital twin and ISSLO-CNN

To comprehensively evaluate the temporal prediction performance of the proposed machining error prediction model based on digital twin and ISSLO-CNN under typical practical machining conditions, and to verify its capability to model multisource fused features and complex error evolution patterns, this study designs a closely coupled simulation experiment procedure based on the aforementioned digital twin data fusion strategy and CNN spatiotemporal feature extraction mechanism, thereby systematically validating and analyzing model performance.

Step 1: Construction and feature engineering of the machining error time-series dataset based on digital twin and physical measurement fusion.

The CNC machine tool digital twin platform and multisource sensors deployed on the physical machine tool are used to synchronously acquire signals including vibration amplitude, spindle temperature, cutting force components, and acoustic emission, while the corresponding simulation signals generated by the digital twin under the same operating conditions are simultaneously obtained. The digital twin quadruple model is established according to Eq. (23) to drive synchronous outputs from virtual simulation and physical measurements. The measured and simulated signals are linearly fused according to Eq. (24) to form the feature vector for error analysis. According to Eq. (25), the fused feature vectors within a continuous time window are organized into a two-dimensional feature matrix as the CNN input. Actual machining errors, including dimensional error, contour error, shape error, and surface roughness, measured using a coordinate measuring machine or an online inspection system, are used as prediction labels, covering different operating conditions such as spindle speed, feed rate, cutting depth, tool wear state, and workpiece material. A sliding-window strategy is adopted to construct sample pairs from the fused feature matrices of the preceding T time steps to the machining error sequence at future time steps, and the samples are stratified into training, validation, and test sets according to cutting parameters and tool conditions.

Step 2: Construction of the evaluation metric system.

In addition to root mean square error, mean absolute error, and coefficient of determination for evaluating error prediction accuracy, task-specific metrics are defined to assess the physical consistency of dynamic machining error time-series prediction, including maximum contour error, roundness error, surface roughness prediction error, dimensional tolerance exceedance rate, and error-trend turning-point prediction error. These metrics are used to quantify the model's ability to track key error evolution trends and abrupt changes. Model complexity and inference time are also introduced as auxiliary metrics to evaluate the engineering feasibility of the prediction system.

Step 3: Instantiation of the ISSLO-CNN model.

According to Sections 2.3 and 2.4, the elite-guided adaptive learning-factor mechanism and

the strategy integrating historical successful memory with an adaptive knowledge exchange rate are instantiated. Each ISSLO individual encodes the CNN hyperparameters to be optimized according to Eq. (30), including the convolution kernel size and number of convolution kernels in each convolutional layer, learning rate, number of neurons in the fully connected layer, and batch size. The sum of the validation-set RMSE and the complexity penalty term defined in Eq. (31) is used as the fitness function. Structural hyperparameters such as the number of convolutional layers and pooling method are preset empirically or selected from predefined discrete sets. During the search process, the adaptive learning factor and elite learning-target selection in Eqs. (13)–(15) control the search step size and direction; Eqs. (16)–(19) strengthen the weakest individual using historical successful directions and adaptive perturbations; and Eqs. (20)–(22) dynamically adjust the knowledge exchange rate according to the individual's recent update success rate. These mechanisms work collaboratively to achieve fine-grained search within the hyperparameter space.

Step 4: Model training and robustness assurance.

The training process uses the mean squared error loss defined in Eq. (29) as the objective function and adopts the Adam optimizer for staged training. First, ISSLO performs iterative optimization within the hyperparameter search space. After each candidate solution is decoded, the corresponding CNN is trained and its fitness is evaluated on the validation set. Subsequently, the final CNN model is retrained using the optimal hyperparameters and evaluated on the test set. Early stopping, gradient clipping, and the learning rate and batch size optimized by ISSLO are introduced during training to prevent overfitting and training instability. After each experimental run, comprehensive validation metrics are recorded, and the model parameters corresponding to the minimum validation loss are retained for final testing. Multiple repeated experiments are conducted, and the mean and standard deviation of each metric are reported to ensure the statistical stability of the conclusions.

Step 5: Comparative experiments and ablation analysis.

To independently evaluate the contributions of Improvement 1 and Improvement 2, ablation experiments are constructed as follows. The original SSLO-optimized CNN is used as the baseline, and two variants are constructed: one introducing only the elite-guided adaptive learning-factor mechanism (SSLO + Improvement 1), and the other introducing only the strategy integrating historical successful memory with an adaptive knowledge exchange rate (SSLO + Improvement 2). These variants are compared with the complete ISSLO-CNN model on the same test set. Comparison between the baseline SSLO and SSLO + Improvement 1 quantifies the improvements in convergence accuracy and late-stage local exploitation capability brought by elite guidance and iteration-decaying learning factors. Comparison between the baseline SSLO and SSLO + Improvement 2 identifies the gains of historical successful-direction memory and the adaptive knowledge exchange rate in improving the evolutionary efficiency of the weakest individual and the quality of population information exchange. The complete model is used to evaluate the combined benefits of the two improvements working collaboratively.

Step 6: In-depth analysis of typical machining error scenarios and application implications.

Representative machining error sequences under different cutting parameters, tool wear stages, workpiece material types, and machining modes, such as high-speed milling, deep-hole drilling, and thin-walled part machining, are selected from the test set to compare the prediction performance of ISSLO-CNN and its ablation variants. Particular attention is paid to key characteristics such as error trends during transitions from steady-state cutting to chatter or intensified thermal deformation, abrupt changes in contour error, surface quality degradation, and dimensional out-of-tolerance warnings. Finally, based on the experimental findings, the potential reference value of the proposed method is discussed for applications including online monitoring of CNC machine tool machining errors, tool wear compensation, thermal error compensation modeling, closed-loop optimization of process parameters, digital twin model calibration, and intelligent quality control.

The computational flow of the CNC machine tool machining error analysis and prediction method based on digital twin and ISSLO-CNN is shown in Fig. 6.

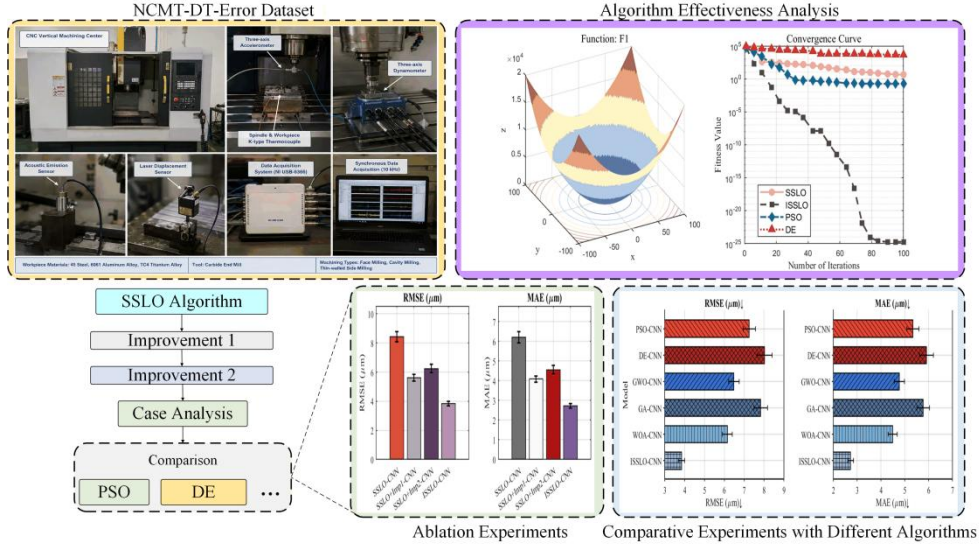


Fig.6 Computational flow of the CNC machine tool machining error analysis and prediction method based on digital twin and ISSLO-CNN

3. Results and discussion

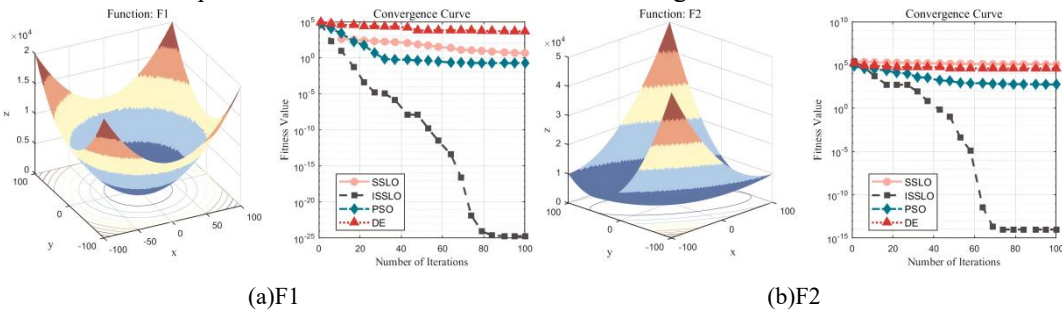
3.1 Effectiveness analysis of the ISSLO algorithm improvements

To systematically evaluate the practical effectiveness of the ISSLO algorithm in hyperparameter optimization tasks, eight representative functions from the CEC2005 benchmark test suite are selected for comparative experiments in this section. The selected functions cover multiple problem types, including unimodal, multimodal, rotated, ill-conditioned, and low-dimensional fixed-dimensional functions, and their basic information is presented in Table 1. The experiments comprehensively compare and analyze the performance of ISSLO against the original SSLO, Particle Swarm Optimization (PSO), and Differential Evolution (DE) in terms of convergence accuracy, convergence speed, statistical significance, and ablation contributions.

Table 1. Basic information of the test functions

No.	Function name	Domain	Search range
F1	Sphere	30	$[-100,100]$
F2	Rosenbrock	30	$[-100,100]$
F3	Schwefel 2.22	30	$[-30,30]$
F4	Quartic with Noise	30	$[-1.28,1.28]$
F5	Rastrigin	30	$[-5.12,5.12]$
F6	Griewank	30	$[-600,600]$
F7	Penalized 1	30	$[-50,50]$
F8	Kowalik	4	$[-5,5]$

To intuitively illustrate the differences in convergence speed and optimization accuracy among the algorithms, the original SSLO, the proposed improved SSLO (ISSLO), Particle Swarm Optimization (PSO), and Differential Evolution (DE) are tested on the above eight benchmark functions. The comparative simulation results are shown in Fig. 7.



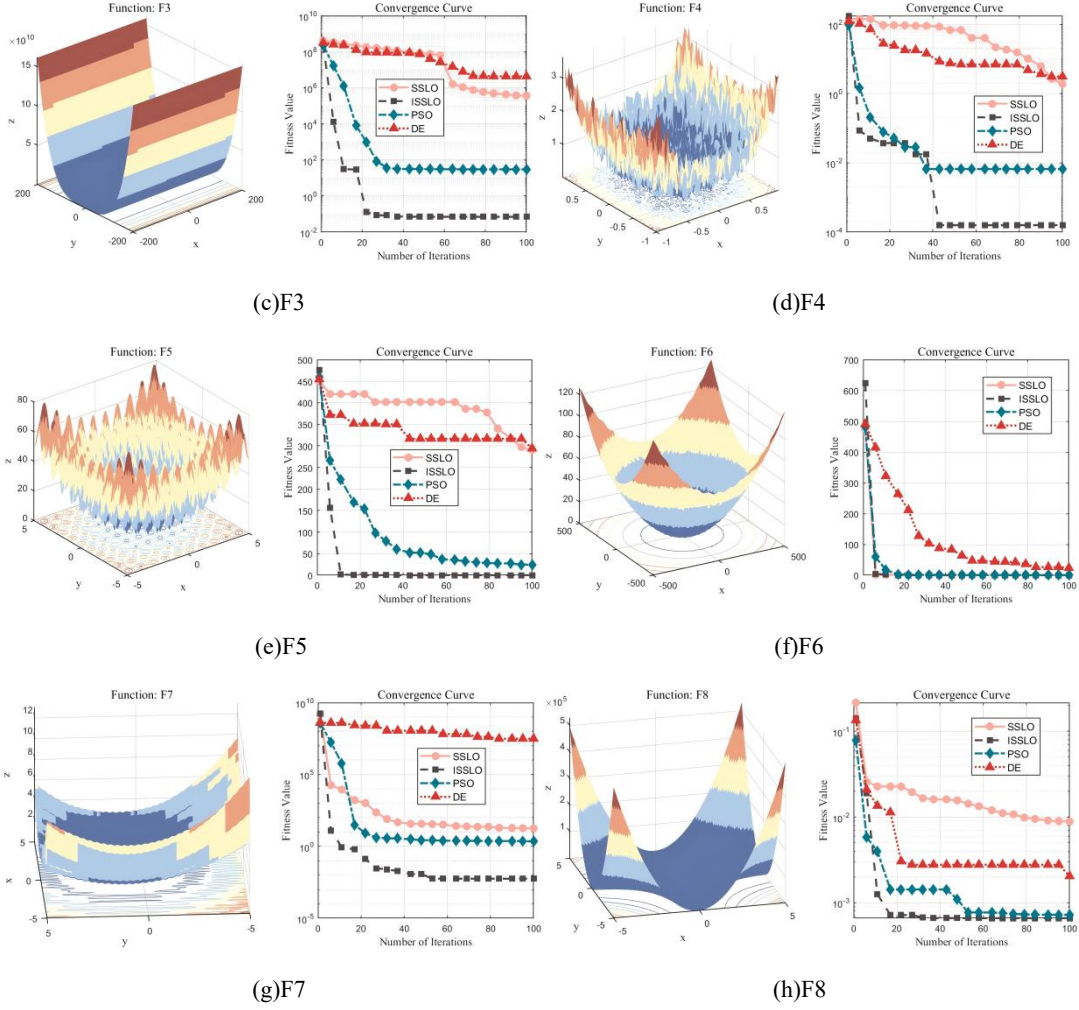


Fig.7 Comparison of convergence results of different metaheuristic algorithms on test functions

Analysis of Fig. 7(a) shows that, on the unimodal Sphere function (F1), all four algorithms exhibit monotonically decreasing convergence trends, but significant differences are observed in convergence accuracy and speed. ISSLO starts from an initial fitness value of approximately 8×10^4 , rapidly decreases to the order of 10^{-6} within the first 20 iterations, and then continues to converge at an accelerated rate, reaching the order of 10^{-24} at around the 80th iteration and finally stabilizing at approximately 10^{-25} , demonstrating extremely strong local exploitation capability. In comparison, SSLO and PSO both stagnate at around the 30th iteration, with their final fitness values remaining only at the order of 10^0 , approximately 2 and 0.5, respectively. Their convergence curves are highly overlapped, indicating that the late-stage exploitation capability of the original SSLO on a simple unimodal problem is comparable to that of the classical PSO. In contrast, DE shows almost no effective convergence and remains at the order of 10^4 - 10^5 throughout the iterations, with a final fitness value of approximately 5×10^4 . These results indicate that the elite-guided adaptive learning-factor mechanism enables ISSLO to continuously reduce its search step size in the later iterations through iterative cosine decay and rank-based scaling, thereby avoiding oscillations around the optimal neighborhood caused by the fixed reciprocal distribution in the original SSLO and achieving an accuracy improvement of more than 20 orders of magnitude on the unimodal function.

Analysis of Fig. 7(b) shows that the performance gap among the algorithms becomes even more pronounced on the ill-conditioned Rosenbrock function (F2). The Rosenbrock function has a long and narrow curved valley landscape, placing extremely high demands on the directional perception capability of optimization algorithms. ISSLO starts from an initial value of approximately 5×10^5 , decreases to the order of 10^{-1} within the first 30 iterations, and then maintains a stable convergence trend, reaching approximately 10^{-13} at around the 75-th iteration before stabilizing. It is the only algorithm capable of effectively traversing the curved valley and

approaching the global optimum. PSO decreases to approximately 10^2 in the early stage and then stagnates, with a final fitness value of approximately 50. SSLO and DE almost completely fail to escape the initial search region, remaining at the order of 10^5 throughout the iterations, with final fitness values of approximately 5×10^5 and 3×10^5 , respectively. The outstanding performance of ISSLO on this function can be attributed to the synergistic effects of the two improvements. The elite-guided learning-target selection mechanism allows individuals to learn with a higher probability from elite solutions ranked highly by fitness, thereby ensuring that the search direction follows the valley. Meanwhile, the historical successful-direction memory mechanism enables the weakest individual to continue moving along historically successful directions, avoiding directional loss caused by the purely random reflection strategy in long and narrow valleys.

Analysis of Fig. 7(c) shows that ISSLO also exhibits a significant performance advantage on the multimodal Schwefel2.22 function (F3). The initial fitness values of this function are on the order of 10^8 - 10^9 . ISSLO rapidly decreases to the order of 10^1 within the first 15 iterations, reaches approximately 0.5 at around the 20-th iteration, and finally stabilizes near 0.3, converging much faster than the other three algorithms. PSO stagnates at approximately 50 after the 20-th iteration. SSLO decreases relatively slowly and reaches approximately 5×10^5 at around the 60-th iteration before leveling off, with a final fitness value of approximately 5×10^5 . DE performs the worst, fluctuating at the order of 10^7 - 10^8 throughout the iterations and ending at approximately 5×10^6 . Notably, the convergence curve of ISSLO on this function exhibits a typical two-stage pattern of “rapid decrease–stable transition”. The sharp decrease during the first 10 iterations corresponds to the global exploration stage driven by the elite-guided mechanism, whereas the fine convergence after the 10th iteration corresponds to the local exploitation stage driven by the decay of the adaptive learning factor, verifying that the proposed improvements achieve an effective balance between exploration and exploitation.

Analysis of Fig. 7(d) shows that, on the noisy Quartic function (F4), ISSLO still exhibits a clear advantage in convergence accuracy, although its convergence speed is relatively close to that of PSO. The initial fitness values of all four algorithms are on the order of 10^2 , ranging from 120 to 200. Both ISSLO and PSO rapidly decrease to the order of 10^{-2} within the first 30 iterations. ISSLO further decreases to approximately 10^{-4} at around the 45-th iteration and then remains stable, whereas PSO stagnates at approximately 8×10^{-3} . SSLO converges the slowest, reaching only approximately 2 by the 80-th iteration and ending at approximately 1.5. DE stagnates at approximately 3 after the 50-th iteration. Because this function contains a random noise term, its fitness landscape includes numerous local perturbations, posing a challenge to the interference resistance of the algorithms. The ability of ISSLO to finally converge to the order of 10^{-4} under noisy conditions indicates that the adaptive perturbation term and historical successful-direction memory mechanism effectively improve robustness against noise, enabling the algorithm to distinguish genuine descent directions from false fluctuations caused by noise.

Analysis of Fig. 7(e) shows that ISSLO demonstrates the most prominent global exploration capability on the highly multimodal Rastrigin function (F5). This function contains hundreds of regularly distributed local optima and is a standard benchmark for evaluating an algorithm’s ability to escape from local optima. Starting from an initial value of approximately 480, ISSLO sharply decreases to approximately 10 within the first 15 iterations, reaches a value close to 0 at around the 20-th iteration, and ultimately converges to 0, indicating that it successfully locates the global optimum. PSO decreases to approximately 30 and then becomes trapped in a local optimum. SSLO converges extremely slowly, remaining above 400 during the first 60 iterations and reaching only approximately 290 by the 100-th iteration. DE performs similarly to SSLO, with a final fitness value of approximately 295. The excellent performance of ISSLO on the Rastrigin function strongly verifies the effectiveness of the two improvements. The elite-guided selection mechanism retains a controllable probability of selecting learning targets from the entire population, thereby avoiding excessive convergence and preserving necessary exploration diversity. Meanwhile, the adaptive perturbation mechanism for the weakest individual automatically increases the perturbation amplitude after consecutive update failures, helping individuals escape from local attraction basins. Their synergistic effect enables ISSLO to maintain strong global search capability in complex multimodal landscapes.

Analysis of Fig. 7(f) shows that, on the Griewank function (F6), ISSLO, SSLO, and PSO can all eventually converge to values close to 0, but ISSLO converges significantly faster. Starting from an initial value of approximately 620, ISSLO requires only about 10 iterations to decrease to

a value close to 0 and becomes fully stable at 0 after the 15-th iteration. SSLO starts from approximately 500 and reaches 0 at around the 20-th iteration, while PSO also converges to 0 at approximately the 20-th iteration. DE converges the slowest and still remains at approximately 30 by the 100th iteration, failing to reach the global optimum. The Griewank function contains a large number of densely distributed local minima but has an overall landscape that tends to converge toward the center, placing high demands on early-stage exploration efficiency. ISSLO drops sharply from 620 to approximately 10 within the first five iterations, with a convergence speed approximately twice that of SSLO. This advantage mainly results from the elite-guided learning-target selection mechanism, which accelerates the dissemination of high-quality knowledge during the early iterations and enables the population to rapidly aggregate toward promising regions.

Analysis of Fig. 7(g) shows that ISSLO again demonstrates a significant accuracy advantage on the Penalized1 function (F7). The initial fitness values of this function are on the order of 10^7 – 10^8 . ISSLO decreases to the order of 10^0 within the first 15 iterations, reaches approximately 0.2 at around the 25-th iteration, and finally stabilizes near 0.1. PSO stagnates at approximately 3 after the 20-th iteration. SSLO decreases to approximately 100 and then levels off, with a final fitness value of approximately 100. DE shows almost no convergence and remains at the order of 10^7 throughout the iterations. Compared with SSLO, ISSLO achieves an improvement of approximately three orders of magnitude in accuracy, and compared with PSO, the improvement is approximately 1.5 orders of magnitude. This function divides the search space into multiple feasible regions through penalty terms, placing high demands on boundary handling and directional search capability. The elite-guided mechanism and historical successful-direction memory of ISSLO exhibit good adaptability to such structurally constrained problems.

Analysis of Fig. 7(h) shows that, on the low-dimensional Kowalik function (F8), ISSLO and PSO achieve similar final convergence accuracy, but ISSLO converges faster. This function is a fixed four-dimensional problem with initial fitness values ranging from 0.1 to 0.5. ISSLO starts from approximately 0.15, decreases to approximately 1.2×10^{-3} by the 15-th iteration, reaches approximately 6×10^{-4} by the 25-th iteration, and then remains stable. PSO starts from approximately 0.1 and converges slightly more slowly, reaching approximately 6×10^{-4} at around the 50-th iteration. SSLO converges the slowest and reaches only approximately 9×10^{-3} by the 100-th iteration. DE stagnates at approximately 2.5×10^{-3} after the 20-th iteration. Notably, PSO exhibits strong competitiveness on this low-dimensional problem, achieving a final accuracy comparable to that of ISSLO. However, ISSLO requires only about half as many iterations as PSO to reach the same accuracy, indicating that the convergence-speed improvement brought by the proposed strategies remains consistent across different dimensionalities.

Overall analysis of the eight groups of convergence curves in Fig. 7 shows that the proposed ISSLO algorithm achieves convergence performance superior to, or at least no worse than, that of the original SSLO, PSO, and DE on all eight benchmark test functions, with particularly pronounced advantages on high-dimensional complex multimodal functions such as F2, F3, F5, and F7. In terms of convergence accuracy, ISSLO improves upon the original SSLO by approximately 25, 18, 6, 3, and 3 orders of magnitude on F1, F2, F3, F5, and F7, respectively, and by approximately 1-2 orders of magnitude on F4 and F8. Only on F6 is no clear accuracy difference observed because all algorithms are able to converge to the global optimum. In terms of convergence speed, the number of iterations required for ISSLO to reach stable convergence on most functions is approximately one-third to one-half that required by SSLO. These experimental results verify, from the perspective of numerical optimization, the effectiveness of the two improvement strategies proposed in Section 2.3. The elite-guided adaptive learning-factor mechanism addresses the late-stage oscillation problem caused by the fixed learning-factor distribution in the original SSLO through iteration-dependent decay and rank-based scaling, while elite learning-target selection enhances directional search capability. The improvement integrating historical successful memory and the adaptive knowledge exchange rate enhances the improvement efficiency of the weakest individual and the quality of population information exchange. Their synergistic effect enables ISSLO to achieve a more refined dynamic balance between exploration and exploitation, providing an algorithmic performance foundation for its subsequent application to CNN hyperparameter optimization.

3.2 Data sources for the case study

To verify the effectiveness of the proposed CNC machine tool machining error analysis and

prediction method based on digital twin and ISSLO-CNN in practical machining scenarios, a time-series dataset for CNC machine tool machining errors under multiple operating conditions and multi-sensor fusion is constructed and named NCMT-DT-Error. The dataset is generated synchronously from a three-axis vertical machining center of a specific model and its digital twin. It includes physical measurement signals, digital twin simulation signals, and machining error labels obtained from a coordinate measuring machine and an online inspection system, providing a relatively comprehensive representation of the multisource coupling characteristics involved in error formation and evolution during typical machining processes.

Physical experimental platform: The dataset is collected from a three-axis vertical machining center equipped with a FANUC 0i-MF CNC system. The maximum spindle speed is 12,000 r/min, and the worktable size is 1000 mm × 500 mm. Three representative workpiece materials are used: 45 steel, 6061 aluminum alloy, and TC4 titanium alloy. Carbide end mills with diameters of 6 mm, 8 mm, and 10 mm are employed. The machining operations include face milling, pocket milling, and side milling of thin-walled parts. The sensors deployed on the physical machine tool include a triaxial accelerometer, K-type thermocouples installed at the spindle and workpiece regions, a triaxial dynamometer, an acoustic emission sensor, and a laser displacement sensor. All signals are sampled at 10 kHz and synchronously recorded using an NI data acquisition card. The experimental platform is shown in Fig. 8.

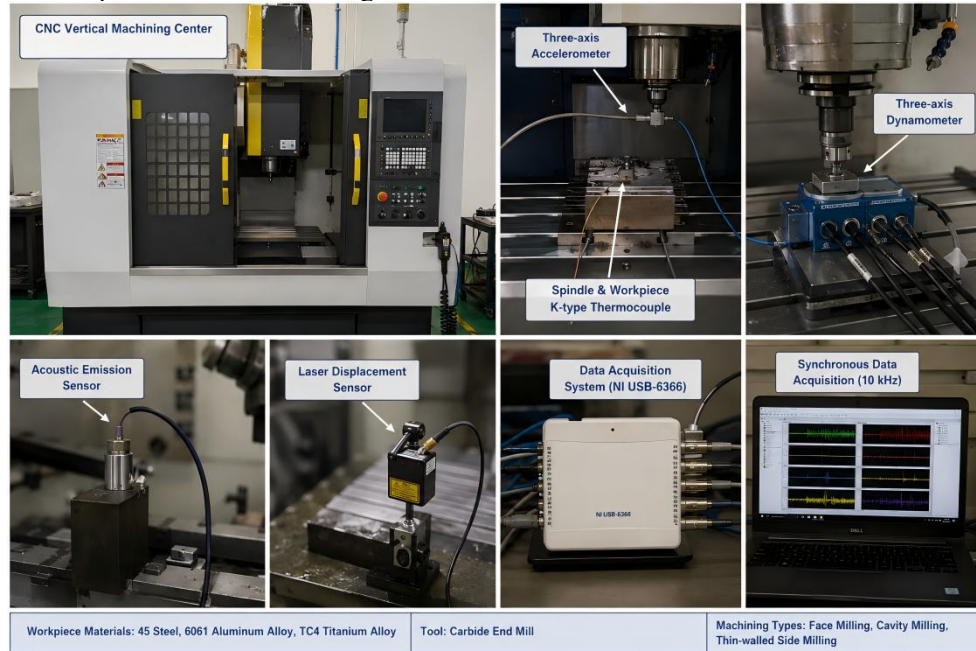


Fig.8 Experimental platform

The digital twin of the machining center is constructed according to Eq. (23). The geometric model is built in SolidWorks based on the actual dimensions and assembly relationships of the machine tool. The physical model, including cutting force, heat transfer and thermal deformation, and tool wear, is implemented through co-simulation in MATLAB/Simulink and ANSYS. The behavioral model characterizes dynamic responses under different operating conditions using parametric simulation and response surface methods, while the rule model incorporates process parameter ranges, expert knowledge, and constraint conditions. Under the same operating conditions and time sequence, the digital twin outputs simulated vibration, temperature, and cutting force signals, which complement the corresponding physical measurements.

To cover representative machining conditions and improve dataset representativeness, an orthogonal experimental design is used to configure key factors including spindle speed, feed rate, cutting depth, tool wear stage, and workpiece material. The final dataset contains 12 representative operating-condition combinations, with three repeated cutting tests for each condition, resulting in 36 complete machining-process records. Each record lasts approximately 25 s on average, and about 18,000 valid samples are obtained after sliding-window sampling. The dataset is stratified by operating condition and divided into training, validation, and test sets at an approximate ratio of 7:1.5:1.5, ensuring a consistent distribution of samples from the same operating condition

across the three subsets. The test set also includes a proportion of previously unseen combinations of tool wear states and workpiece materials to evaluate the model's generalization capability.

3.3 Ablation experiments

To independently assess the contribution of each improvement proposed in Section 2.3 to CNC machine tool machining error prediction, this section strictly follows the ablation scheme described in Step 5 of Section 2.5 and systematically compares four model variants on the NCMT-DT-Error dataset. The four models are: the baseline model in which the original SSLO is used to optimize the CNN (SSLO-CNN); a model incorporating only the elite-guided adaptive learning-factor mechanism (SSLO + Improvement 1-CNN); a model incorporating only the historical successful-memory and adaptive knowledge exchange rate mechanism (SSLO + Improvement 2-CNN); and the full model integrating both improvements (ISSLO-CNN). To ensure a fair and attributable comparison, all models use the same CNN hyperparameter search ranges, early-stopping criteria, and random seed set, with the optimization algorithm configuration being the only variable.

For evaluation, eight metrics are used to comprehensively assess prediction accuracy and physical consistency: root mean square error (RMSE), mean absolute error (MAE), coefficient of determination (R^2), maximum contour error, roundness error, surface roughness Ra prediction error, dimensional tolerance exceedance rate, and error-trend turning-point prediction error. Model parameter count and inference time are also included to assess model complexity and engineering feasibility. The ablation results for these metrics are shown in Fig. 9.

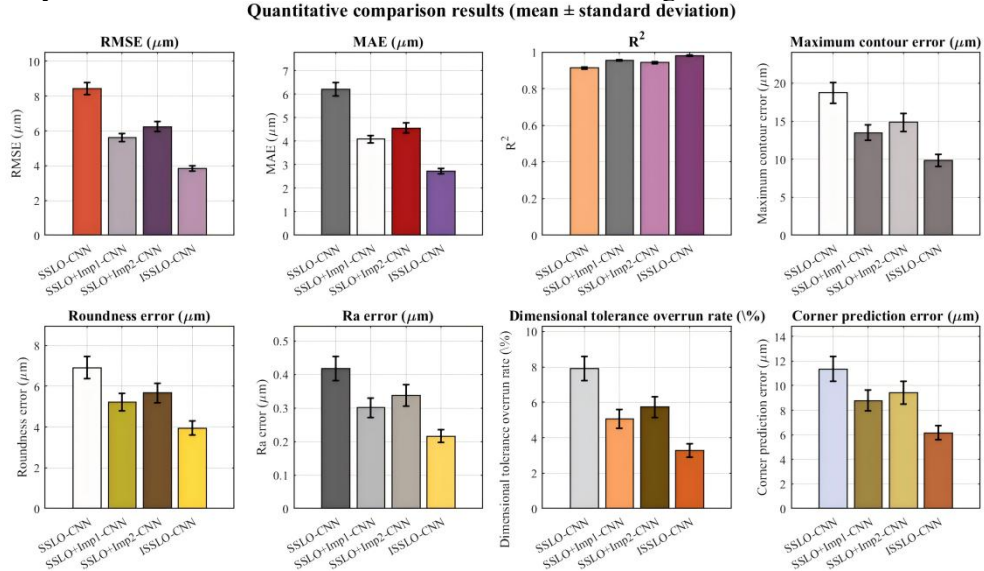


Fig.9 Comparison results of the ablation experiments—prediction accuracy and physical consistency metrics

Analysis of Fig. 9 shows a clear stepwise improvement in the error prediction performance of the four model variants on the NCMT-DT-Error dataset. The complete ISSLO-CNN achieves the best results across all eight evaluation metrics, quantitatively confirming both the individual contributions and the synergistic gains of the two proposed improvements.

In terms of overall prediction accuracy, the baseline SSLO-CNN yields an RMSE of $8.43 \pm 0.35 \mu\text{m}$, an MAE of $6.21 \pm 0.28 \mu\text{m}$, and an R^2 of 0.913 ± 0.006 . The complete ISSLO-CNN reduces the RMSE to $3.84 \pm 0.16 \mu\text{m}$ and the MAE to $2.72 \pm 0.12 \mu\text{m}$, while increasing R^2 to 0.979 ± 0.003 . Compared with the baseline, the RMSE and MAE are reduced by 54.4% and 56.2%, respectively, and R^2 increases by 0.066, indicating that the two improvements work synergistically to substantially enhance CNN hyperparameter optimization. Improvement 1, namely the elite-guided adaptive learning-factor mechanism, provides a substantial and consistent gain in prediction accuracy. Compared with the baseline SSLO-CNN, SSLO + Improvement 1-CNN reduces the RMSE from $8.43 \pm 0.35 \mu\text{m}$ to $5.62 \pm 0.23 \mu\text{m}$, a decrease of 33.3%, and the MAE from $6.21 \pm 0.28 \mu\text{m}$ to $4.08 \pm 0.17 \mu\text{m}$, a decrease of 34.3%, while increasing R^2 from 0.913 ± 0.006 to 0.954 ± 0.004 , an improvement of 0.041. In terms of physical consistency, the maximum contour error decreases from $18.72 \pm 1.36 \mu\text{m}$ to $13.48 \pm 1.02 \mu\text{m}$, a reduction of 28.0%; the roundness error decreases from $6.92 \pm 0.55 \mu\text{m}$ to $5.21 \pm 0.43 \mu\text{m}$, a reduction of 24.7%; the surface roughness Ra prediction error decreases from $0.418 \pm 0.036 \mu\text{m}$ to $0.301 \pm 0.029 \mu\text{m}$, a

reduction of 28.0%; the dimensional tolerance exceedance rate decreases from $7.92 \pm 0.68\%$ to $5.08 \pm 0.53\%$, a reduction of 35.9%; and the error-trend turning-point prediction error decreases from $11.36 \pm 1.02 \mu\text{m}$ to $8.79 \pm 0.83 \mu\text{m}$, a reduction of 22.6%. These results show that the elite-guided learning-target selection mechanism accelerates the spread of high-quality knowledge and strengthens directional search, while the adaptive learning factor based on cosine decay and rank scaling improves local exploitation accuracy in the CNN hyperparameter space. Improvement 2, which integrates historical successful memory with an adaptive knowledge exchange rate, also improves performance across all metrics, although to a slightly lesser extent than Improvement 1. Relative to the baseline, SSLO + Improvement 2-CNN reduces the RMSE to $6.25 \pm 0.29 \mu\text{m}$, a reduction of 25.9%, and the MAE to $4.56 \pm 0.22 \mu\text{m}$, a reduction of 26.6%, while increasing R^2 to 0.942 ± 0.005 , an improvement of 0.029. The maximum contour error decreases to $14.83 \pm 1.18 \mu\text{m}$, a reduction of 20.8%; the roundness error decreases to $5.67 \pm 0.48 \mu\text{m}$, a reduction of 18.1%; the Ra error decreases to $0.338 \pm 0.032 \mu\text{m}$, a reduction of 19.1%; the dimensional tolerance exceedance rate decreases to $5.74 \pm 0.59\%$, a reduction of 27.5%; and the turning-point prediction error decreases to $9.42 \pm 0.91 \mu\text{m}$, a reduction of 17.1%. A direct comparison shows that SSLO + Improvement 1-CNN outperforms SSLO + Improvement 2-CNN on all eight metrics, with an RMSE lower by $0.63 \mu\text{m}$, an MAE lower by $0.48 \mu\text{m}$, and an R^2 higher by 0.012. This indicates that, in the present hyperparameter optimization task, the elite-guided adaptive learning-factor mechanism provides a stronger improvement in convergence accuracy and directional search capability.

The complete ISSLO-CNN shows clear additional gains over both single-improvement models. Its RMSE is further reduced by 31.7% and 38.6% relative to SSLO + Improvement 1-CNN and SSLO + Improvement 2-CNN, respectively; the MAE is further reduced by 33.3% and 40.4%; and R^2 increases by 0.025 and 0.037. For the physical consistency metrics, the maximum contour error decreases to $9.86 \pm 0.78 \mu\text{m}$, corresponding to reductions of 26.9% and 33.5% relative to the two single-improvement models; the roundness error decreases to $3.95 \pm 0.34 \mu\text{m}$, with reductions of 24.2% and 30.3%; the Ra error decreases to $0.216 \pm 0.019 \mu\text{m}$, with reductions of 28.2% and 36.1%; the dimensional tolerance exceedance rate decreases to $3.28 \pm 0.37\%$, with reductions of 35.4% and 42.9%; and the turning-point prediction error decreases to $6.16 \pm 0.56 \mu\text{m}$, with reductions of 29.9% and 34.6%. Among these metrics, the dimensional tolerance exceedance rate shows the largest reduction relative to the baseline, at 58.6%, indicating that the combined improvements most strongly enhance the model's ability to identify and warn against critical machining-error exceedance events.

In addition, ISSLO-CNN has the smallest standard deviation among the four models for all eight metrics, with an RMSE standard deviation of only $0.16 \mu\text{m}$ and an R^2 standard deviation of only 0.003. This shows that the two improvements not only raise average prediction accuracy but also markedly improve stability and reproducibility across repeated runs. Overall, both improvement strategies contribute positively to the error prediction performance of SSLO-CNN. Improvement 1 provides a larger gain in convergence accuracy and directional search, while Improvement 2 offers complementary benefits in reinforcing disadvantaged individuals and improving information exchange. By combining the two, the complete ISSLO-CNN outperforms both the baseline and the single-improvement models in prediction accuracy, physical consistency, and operational stability, confirming the effectiveness and synergistic value of the proposed improvements for CNC machine tool machining error prediction.

To further assess the deployment potential of the proposed method in practical engineering scenarios, this section compares the model variants in terms of parameter count and single-sample inference time, with the results shown in Fig. 10.

Fig. 10 shows that the four model variants follow a clear progression in both model complexity and engineering feasibility, consistent with the trend observed for prediction accuracy. The full ISSLO-CNN achieves the smallest model size and the shortest inference time, indicating that the proposed improvements enhance prediction accuracy without increasing computational complexity. Instead, they enable simultaneous gains in both accuracy and efficiency.

In terms of model size, the baseline SSLO-CNN contains 2.87 M parameters. This is reduced to 2.36 M with SSLO + Improvement 1-CNN and 2.52 M with SSLO + Improvement 2-CNN, corresponding to reductions of 17.8% and 12.2%, respectively. The full ISSLO-CNN further reduces the parameter count to 1.97 M, which is 31.4% lower than that of the baseline and 16.5% and 21.8% lower than those of the two single-improvement variants, respectively. These results

suggest that the elite-guided adaptive learning-factor mechanism enables a more refined search of the CNN hyperparameter space through cosine decay and fitness-based scaling, allowing the optimizer to identify more compact combinations of convolutional filters and fully connected neurons. The historical successful-direction memory and adaptive knowledge exchange mechanism further improve search efficiency and help eliminate redundant configurations. Together, the two strategies steer ISSLO toward compact CNN architectures with stronger generalization capability, effectively limiting the overparameterization penalized in Eq. (31).

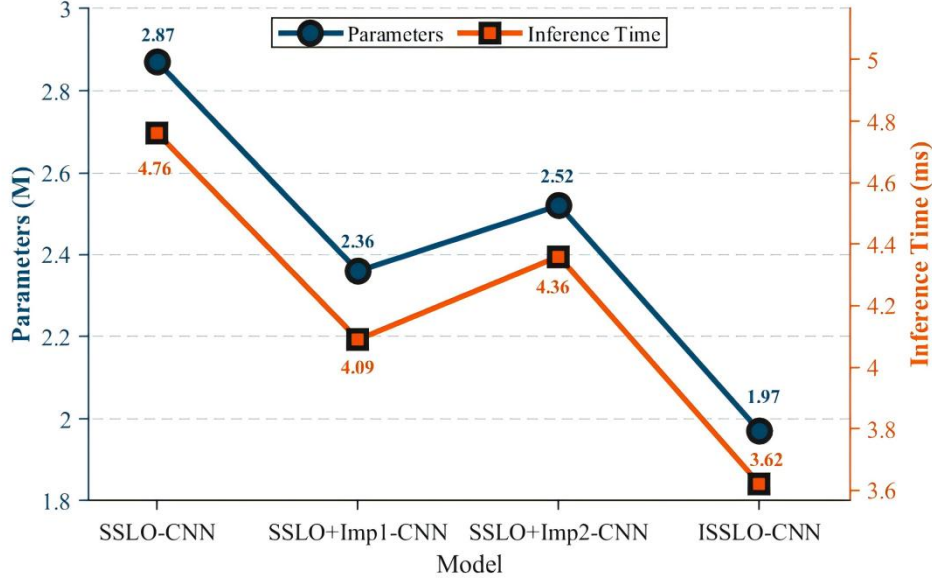


Fig.10 Comparison results of the ablation experiments—model complexity and engineering feasibility metrics

A similar trend is observed for inference efficiency. The baseline SSLO-CNN requires 4.76 ms per sample, whereas SSLO + Improvement 1-CNN and SSLO + Improvement 2-CNN reduce the inference time to 4.09 ms and 4.36 ms, respectively, corresponding to reductions of 14.1% and 8.4%. The full ISSLO-CNN further shortens the inference time to 3.62 ms, representing a reduction of 23.9% relative to the baseline and reductions of 11.5% and 17.0% relative to the two single-improvement models. Notably, SSLO + Improvement 1-CNN outperforms SSLO + Improvement 2-CNN in both parameter count (2.36 M vs. 2.52 M) and inference time (4.09 ms vs. 4.36 ms), consistent with its stronger performance in the prediction-accuracy metrics. This suggests that the elite-guided mechanism is more effective at identifying favorable accuracy–complexity trade-offs in the hyperparameter space, whereas Improvement 2 mainly contributes through enhanced search robustness and information exchange.

Overall, ISSLO-CNN achieves the best prediction performance ($\text{RMSE} = 3.84 \pm 0.16 \mu\text{m}$, $R^2 = 0.979 \pm 0.003$) while also using the fewest parameters (1.97 M) and requiring the shortest inference time (3.62 ms). This shows that the proposed optimization strategy improves accuracy without sacrificing computational efficiency. The advantage arises from the complementary roles of the two improvements: elite-guided learning accelerates the propagation of promising hyperparameter configurations, adaptive learning-factor decay refines local search in later iterations, and historical successful-direction memory together with adaptive knowledge exchange reduces ineffective exploration. As a result, ISSLO is able to identify a CNN configuration that is both accurate and compact within a limited number of iterations. From an engineering perspective, the 3.62 ms single-sample inference time meets the low-latency requirements of online machining error monitoring and real-time compensation, while the 1.97 M parameter count also facilitates deployment on industrial edge-computing devices. These characteristics support the practical application of the proposed method in intelligent quality control, tool wear compensation, and closed-loop process parameter optimization.

3.4 Comparative experiments with different algorithms

To further evaluate the overall performance of ISSLO for CNN hyperparameter optimization and benchmark it against a broader range of competing algorithms, this section selects three representative metaheuristic algorithms—Particle Swarm Optimization (PSO), Differential

Evolution (DE), and the original SSLO—as comparison methods on the NCMT-DT-Error dataset [16–18]. Using the same hyperparameter search space and fitness function, these algorithms are employed to optimize the CNN, resulting in six models: PSO-CNN, DE-CNN, GWO-CNN, GA-CNN, WOA-CNN, and the proposed ISSLO-CNN. Their performance is systematically compared in terms of prediction accuracy, physical consistency, convergence behavior, and engineering feasibility. For all comparison algorithms, the population size is set to 20 and the maximum number of iterations to 100. Each algorithm is independently run 30 times, and the mean and standard deviation are reported. The CNN hyperparameter search ranges, training strategy, and evaluation metrics are kept identical to those used in Section 3.3, with the optimization algorithm being the only difference, thereby ensuring a fair comparison. The comparative results of the different algorithms are shown in Fig. 11.

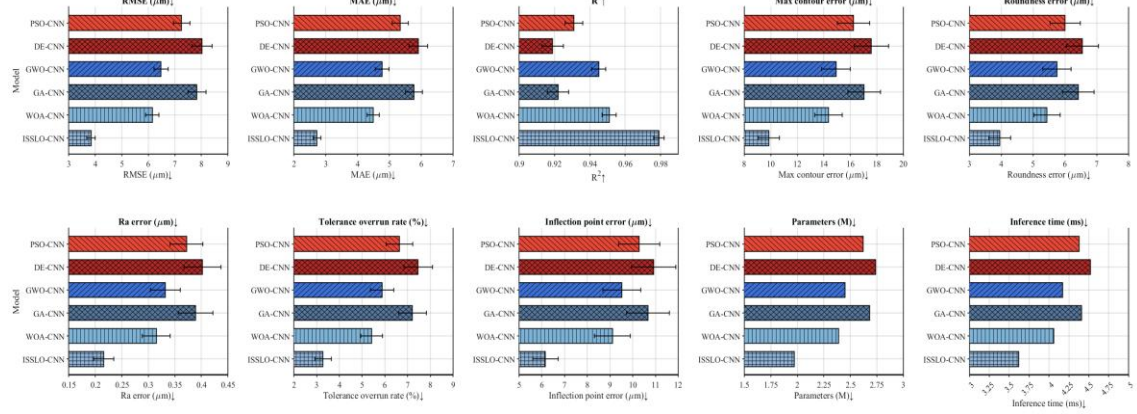


Fig.11 Comparison results of different algorithms

Analysis of Fig. 11 shows a clear performance hierarchy among the six CNN models optimized by different metaheuristic algorithms on the NCMT-DT-Error dataset. ISSLO-CNN significantly outperforms the other five methods across all ten metrics, followed by WOA-CNN and GWO-CNN in the second tier, PSO-CNN and GA-CNN in the middle, and DE-CNN with the weakest overall performance. It should be noted that standard deviations are not reported for parameter count and inference time in the figure. This is because the parameter count is uniquely determined by the CNN structural hyperparameters obtained after convergence, such as the number of convolution kernels and fully connected neurons, and is therefore a deterministic property of the model that does not vary with the randomness of independent algorithm runs. Inference time is measured in a standardized manner for the final optimal model on a fixed hardware platform. Its value is mainly determined by the model size and computational graph structure, while run-to-run variations are primarily caused by system scheduling and workload fluctuations rather than the stochastic nature of the optimization algorithm. Therefore, a deterministic mean value is sufficient for engineering evaluation.

In terms of the core prediction metrics, ISSLO-CNN achieves an RMSE of $3.84 \pm 0.16 \mu\text{m}$, an MAE of $2.72 \pm 0.12 \mu\text{m}$, and an R^2 of 0.979 ± 0.003 . Compared with the second-best WOA-CNN (RMSE = $6.15 \pm 0.25 \mu\text{m}$, MAE = $4.49 \pm 0.19 \mu\text{m}$, $R^2 = 0.951 \pm 0.004$), the RMSE and MAE are reduced by 37.6% and 39.4%, respectively, while R^2 increases by 0.028. Compared with the worst-performing DE-CNN (RMSE = $8.02 \pm 0.38 \mu\text{m}$, MAE = $5.91 \pm 0.29 \mu\text{m}$, $R^2 = 0.919 \pm 0.006$), the corresponding improvements are 52.1%, 54.0%, and 0.060, demonstrating a substantial accuracy advantage. ISSLO-CNN also has the smallest standard deviations among the six models for all three accuracy metrics (0.16 μm , 0.12 μm , and 0.003), indicating the highest reproducibility over the 30 independent runs.

ISSLO-CNN also performs best across all physical consistency metrics. Its maximum contour error is $9.86 \pm 0.78 \mu\text{m}$, 31.3% lower than that of WOA-CNN ($14.36 \pm 1.04 \mu\text{m}$) and 43.9% lower than that of DE-CNN ($17.58 \pm 1.29 \mu\text{m}$). Its roundness error is $3.95 \pm 0.34 \mu\text{m}$, 27.4% lower than that of WOA-CNN ($5.44 \pm 0.41 \mu\text{m}$). The surface roughness Ra prediction error is $0.216 \pm 0.019 \mu\text{m}$, a reduction of 31.4% compared with WOA-CNN ($0.315 \pm 0.026 \mu\text{m}$). The dimensional tolerance exceedance rate is $3.28 \pm 0.37\%$, which is 39.5% lower than that of WOA-CNN ($5.42 \pm 0.48\%$) and 56.0% lower than that of DE-CNN ($7.46 \pm 0.63\%$). This is the largest relative reduction among all metrics, indicating that ISSLO provides the most pronounced improvement in identifying and warning of critical machining-error exceedance events. The error-trend

turning-point prediction error is $6.16 \pm 0.56 \mu\text{m}$, 32.4% lower than that of WOA-CNN ($9.11 \pm 0.79 \mu\text{m}$). In terms of model complexity and engineering feasibility, ISSLO-CNN also achieves the best results, with 1.97 M parameters and a single-sample inference time of 3.62 ms. The parameter count is 17.6% lower than that of WOA-CNN (2.39 M) and 28.1% lower than that of DE-CNN (2.74 M), while the inference time is reduced by 10.8% relative to WOA-CNN (4.06 ms) and by 19.9% relative to DE-CNN (4.52 ms). A clear positive correlation is also observed between inference time and parameter count across the six models, confirming the direct effect of model size on inference latency.

Overall, through the combined effects of the elite-guided adaptive learning-factor mechanism and the strategy integrating historical successful memory with an adaptive knowledge exchange rate, ISSLO achieves a more refined and efficient search of the CNN hyperparameter space than classical metaheuristic algorithms such as PSO, DE, GWO, GA, and WOA. It not only delivers the largest improvements in prediction accuracy and physical consistency, but also reduces model size and inference latency, overcoming the conventional trade-off between accuracy and complexity. The inference latency of 3.62 ms fully meets the low-latency requirements of online CNC machine tool machining error monitoring and real-time compensation, while the 1.97 M parameter count also facilitates deployment on industrial edge-computing devices. These results provide strong engineering support for applying the proposed method to intelligent quality control, tool wear compensation, and closed-loop process parameter optimization.

4. Conclusion

To address insufficient multisource information fusion and the limited hyperparameter optimization capability of SSLO in CNC machine tool machining error prediction, this study proposes a machining error analysis and prediction method based on digital twin and ISSLO-CNN. Its effectiveness is validated through benchmark-function tests, ablation experiments, and comparisons with multiple optimization algorithms. The main conclusions are as follows:

(1) ISSLO effectively improves search accuracy and convergence efficiency for complex optimization problems. Across eight CEC2005 benchmark functions, ISSLO performs better than or at least comparably to SSLO, PSO, and DE. Compared with the original SSLO, its convergence accuracy on F1, F2, F3, F5, and F7 improves by approximately 25, 18, 6, 3, and 3 orders of magnitude, respectively. For most functions, the number of iterations required to reach stable convergence is only about one-third to one-half that of SSLO, indicating that the proposed elite-guided learning, adaptive learning, and historical successful-memory mechanisms effectively enhance both exploration and exploitation.

(2) ISSLO-CNN significantly improves machining error prediction accuracy and physical consistency. On the NCMT-DT-Error dataset, the model achieves an RMSE of $3.84 \pm 0.16 \mu\text{m}$, an MAE of $2.72 \pm 0.12 \mu\text{m}$, and an R^2 of 0.979 ± 0.003 . Compared with WOA-CNN, the RMSE and MAE are reduced by 37.6% and 39.4%, respectively. The maximum contour error, roundness error, Ra prediction error, dimensional tolerance exceedance rate, and trend turning-point prediction error are $9.86 \pm 0.78 \mu\text{m}$, $3.95 \pm 0.34 \mu\text{m}$, $0.216 \pm 0.019 \mu\text{m}$, $3.28 \pm 0.37\%$, and $6.16 \pm 0.56 \mu\text{m}$, respectively, all outperforming the comparison models.

(3) The proposed method achieves a good balance among prediction performance, model complexity, and real-time capability. Ablation experiments show that both improvement strategies independently enhance model performance. Compared with SSLO-CNN, the complete ISSLO-CNN reduces RMSE and MAE by 54.4% and 56.2%, respectively, while increasing R^2 to 0.979 ± 0.003 . At the same time, the model contains only 1.97 M parameters and requires 3.62 ms for single-sample inference, representing reductions of 31.4% and 23.9%, respectively, relative to SSLO-CNN. These results indicate good potential for lightweight deployment and online applications.

The current experiments are still mainly based on a single machining platform and a limited set of representative operating conditions. Future work will further evaluate the method across different machine tools, materials, and more complex operating conditions, and investigate online learning and closed-loop error compensation to improve its generalization capability and engineering applicability.

5. List of abbreviations

Full name

Abbreviations

Computer Numerical Control	CNC
Digital Twin	DT
Convolutional Neural Network	CNN
Long Short-Term Memory	LSTM
Stochastic Social Learning Optimization	SSLO
Improved Stochastic Social Learning Optimization	ISSLO
Root Mean Square Error	RMSE
Mean Absolute Error	MAE
Particle Swarm Optimization	PSO
Differential Evolution	DE
Grey Wolf Optimizer	GWO
Genetic Algorithm	GA
Whale Optimization Algorithm	WOA
Rectified Linear Unit	ReLU

6. Declarations

Availability of data and material: All data included in this study are available upon reasonable request from the corresponding author.

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

Funding: This study was supported by the project “Construction of Adaptive Compensation System for Intelligent Manufacturing Lines and Research on Key Technologies of Digital Twin” (Project ID: 3502Z202573317).

Authors' contributions: Zelin Cao was responsible for the conceptualization, methodology development, research design, project administration, and preparation of the original manuscript. Lihua Chen contributed to algorithm development, software implementation, experimental design, and validation. Zhenbei Lin was responsible for data acquisition, data preprocessing, experimental investigation, and visualization of the results. Yibin Zhuo contributed to formal analysis, result interpretation, validation, and review and editing of the manuscript. All authors discussed the results, contributed to the revision of the manuscript, and approved the final version.

Acknowledgements: The authors gratefully acknowledge the financial support provided by the project “Construction of Adaptive Compensation System for Intelligent Manufacturing Lines and Research on Key Technologies of Digital Twin” (Project ID: 3502Z202573317).

References

- [1] Zhang, Z., Jiang, F., Luo, M., et al. (2024). Geometric error measuring, modeling, and compensation for CNC machine tools: A review. *Chinese Journal of Aeronautics*, 37(2), 163-198. <https://doi.org/10.1016/j.cja.2023.02.035>.
- [2] Sun, J., Zhang, Y., Ma, X., et al. (2026). Multi-source errors evaluation of machine tools: From research gaps to methodologies and applications. *International Journal of Extreme Manufacturing*, 8(2), 022002. <https://doi.org/10.1088/2631-7990/ae1be8>.
- [3] Fu, X., Song, H., Li, S., et al. (2025). Digital twin technology in modern machining: A comprehensive review of research on machining errors. *Journal of Manufacturing Systems*, 79, 134-161. <https://doi.org/10.1016/j.jmsy.2025.01.005>.
- [4] Abed, M., Gameros, A., Mohammad, A., et al. (2023). Swift feedback and immediate error control using a lightweight simulation approach—A case study of the digital-twin-in-the-loop for machining thin-wall structures. *Journal of Manufacturing Systems*, 71, 309-322. <https://doi.org/10.1016/j.jmsy.2023.09.016>.
- [5] Liu, Z., Lang, Z.-Q., Gui, Y., et al. (2024). Digital twin-based anomaly detection for real-time tool condition monitoring in machining. *Journal of Manufacturing Systems*, 75, 163-173. <https://doi.org/10.1016/j.jmsy.2024.06.004>.

- [6] Lu, Q., Zhu, D., Wang, M., et al. (2023). Digital Twin-Driven Thermal Error Prediction for CNC Machine Tool Spindle. *Lubricants*, 11(5), 219. <https://doi.org/10.3390/lubricants11050219>.
- [7] Sun, G., Fan, K., & Yang, J. (2024). Digital-driven in-situ monitoring for thermally-induced volumetric errors of CNC machine tools. *Journal of Manufacturing Processes*, 131, 2000-2015. <https://doi.org/10.1016/j.jmapro.2024.10.029>.
- [8] Sa, G., Jiang, Z., Liu, Z., et al. (2024). An integrated optimization method for measurement points layout and error modeling for digital twin of CNC machine tools. *Precision Engineering*, 90, 1-11. <https://doi.org/10.1016/j.precisioneng.2024.07.013>.
- [9] Bai, W. (2026). Intelligent machining error prediction and compensation system based on convolutional neural networks and improved genetic algorithm. *Scientific Reports*, 16, 18403. <https://doi.org/10.1038/s41598-026-46424-x>.
- [10] Zhao, R., Huang, H., & Mei, L. (2025). A method for constructing digital twins of CNC machine tools feed systems based on hybrid mechanism-data. *Scientific Reports*, 15, 32186. <https://doi.org/10.1038/s41598-025-17587-w>.
- [11] Ye, J., Sunat, K., & Chiewchanwattana, S. (2026). Stochastic social learning optimization: Combining social learning and bucket theory for efficient optimization. *Knowledge-Based Systems*, 341, 115767. <https://doi.org/10.1016/j.knosys.2026.115767>.
- [12] Yang, F., & Lyu, L. (2026). Elite-Guided Collaborative Stochastic Social Learning Optimization for LSTM-Based Carbon Emission Forecasting. *Computers*, 15(7), 441. <https://doi.org/10.3390/computers15070441>.
- [13] Zhang, K., Dong, C., Zhang, T., et al. (2026). Synergy Strategy Bezier Curve Optimization for UAV-Based Moving Target Search in Maritime Search and Rescue. *Journal of Marine Science and Engineering*, 14(14), 1302. <https://doi.org/10.3390/jmse14141302>.
- [14] Wang, J., et al. (2024). Elite-guided equilibrium optimiser based on information enhancement: Algorithm and mobile edge computing applications. *CAAI Transactions on Intelligence Technology*, 9(5), 1126-1171. <https://doi.org/10.1049/cit2.12316>.
- [15] Tatsis, V. A., & Parsopoulos, K. E. (2023). Reinforcement learning for enhanced online gradient-based parameter adaptation in metaheuristics. *Swarm and Evolutionary Computation*, 83, 101371. <https://doi.org/10.1016/j.swevo.2023.101371>.
- [16] Bojappa, K., & Lee, J. (2025). Review on particle swarm optimization: Application toward autonomous dynamical systems. *IEEE/CAA Journal of Automatica Sinica*, 12(9), 1762-1775. <https://doi.org/10.1109/JAS.2024.125028>.
- [17] Reyes-Davila, E., Haro, E. H., Casas-Ordaz, A., et al. (2025). Differential evolution: A survey on their operators and variants. *Archives of Computational Methods in Engineering*, 32(1), 83-112. <https://doi.org/10.1007/s11831-024-10136-0>.
- [18] Wang, D., Hua, N., & Liu, Z. (2026). Multi-strategy improved teaching-learning-based optimization for global optimization and real-world engineering problems. *Symmetry*, 18(6), 942. <https://doi.org/10.3390/sym18060942>.

Biography



Cao Zelin, male, Ph.D., associate professor, primarily engages in teaching and research in industrial digitalization, intelligent manufacturing, and industry-academia-research integration. He has published over 60 papers (including more than 20 core papers), edited or co-edited nine textbooks, filed over 20 patents, and participated in guiding students to win more than 30 awards in various international and domestic competitions.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [Dataprogram.zip](#)