CAAI Transactions on Intelligence Technology

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Enhancing Generalisation via Cascaded Inertia SGD With Learnt Hyperparameters

Yongji Guan[1] | Shunbao Zhang[1,2] | Longbang Wang[1] | Erchao Li[3]

[1]School of Information Science and Engineering, Lanzhou University, Lanzhou, China | [2]College of Information Science and Engineering, Jishou University, Jishou, China | [3]School of Automation and Electrical Engineering, Lanzhou University of Technology, Lanzhou, China

Correspondence: Erchao Li (lecyxp@lut.edu.cn)

Received: 3 November 2025 | **Revised:** 22 December 2025 | **Accepted:** 21 January 2026

Keywords: deep learning | deep neural networks | optimisation

ABSTRACT

A central challenge in deep learning lies in achieving strong model generalisation, an area in which conventional optimisers such as stochastic gradient descent (SGD) often exhibit limitations, even though they ensure convergence. This paper introduces cascaded inertia SGD (CISGD), a novel optimisation algorithm specifically designed to address this challenge. The core mechanism of CISGD involves a hierarchical aggregation of historical gradients: it first accumulates past gradients and then compounds these accumulations to form multi-stage inertial terms. This deep integration of gradient history allows the optimiser produce more stable optimisation trajectories, thereby encouraging convergence towards flatter minima that are empirically associated with improved generalisation. To enhance practicality and robustness, we further eliminate manual hyperparameter tuning through a lightweight feed-forward neural network that automatically learns the optimal coefficients of the inertial terms, ensuring adaptability across diverse architectures and tasks. Extensive experiments on benchmark image classification and remote sensing change detection tasks demonstrate that CISGD consistently achieves superior generalisation performance compared with existing optimisers, demonstrating that CISGD achieves consistent, though moderate, improvements over standard optimisers while maintaining computational efficiency. This work offers a practical and theoretically grounded framework for automated and generalisable optimisation, promoting broader applications of adaptive deep learning in complex real-world tasks.

1 | Introduction

In recent years, deep learning has achieved remarkable success in the field of remote sensing image processing. Deep learning-based models have delivered strong performance across diverse tasks, including change detection [1, 2], object detection [3], semantic segmentation [4], scene recognition [5], and land use classification [6] in remote sensing images. Despite these advances, generalisation remains challenging, especially under distribution shifts and adversarial perturbations. As a core component of training, the optimiser strongly influences both convergence behaviour and generalisation.

Stochastic gradient descent (SGD) and its variant, SGD with momentum (SGDM) [7–9], are among the most widely used optimisers in deep learning. These optimisers rely on iterative updates to the model parameters based on the gradient of the loss function, where momentum helps accelerate convergence by smoothing the updates across iterations. However, SGD-based methods exhibit slow convergence and degrade on sparse or noisy gradients, particularly in high-dimensional models [10, 11]. To address these challenges, the Adam optimiser is presented, which adjusts the learning rate for each parameter based on the first and second moments of the gradient. This adaptation facilitates faster convergence and

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

enhanced stability, particularly in the context of non-convex optimisation problems [12]. Adam has become a common default in many modern architectures, including transformer-based models [13–15]. In particular, network models for remote sensing image processing are also often based on Adam during training [16–18]. However, Adam may lead to overfitting and weaker generalisation in certain tasks [19].

To improve upon Adam, variants such as Adai [20] and Adan [21] are presented. Adai modifies the way the learning rate is adapted by incorporating a weight-decay regularisation term that helps reduce overfitting, thereby improving the generalisation of the model. Adan improves stability on sparse gradients by revising moment computations, yielding more robust updates on imbalanced or sparse data. Despite progress, these methods can be sensitive to hyperparameters and may not consistently deliver generalisation gains across architectures or datasets.

Recent studies have underscored the importance of incorporating mechanisms into optimisers that are capable of suppressing gradient noise and ensuring stable error convergence, as these properties are critical for enhancing the generalisation ability of deep neural networks (NNs) [9, 22, 23]. Such requirements align with established principles in optimisation theory, where stability and convergence guarantees are fundamental for ensuring reliable algorithmic behaviour [24, 25]. Motivated by these insights, we extend conventional optimisation paradigms by formulating a strategy that explicitly integrates the accumulated effect of historical gradients into the parameter update process. In particular, whereas momentum-based optimisers typically rely on information from only a limited number of recent iterations, the proposed method aggregates the entirety of past gradient information when recalibrating the network weights at each step. This formulation enables the optimisation process to attenuate the influence of both the instantaneous gradient and the accumulated historical gradients, thereby guiding the model towards flatter minima. Since flatter minima are closely associated with improved robustness and generalisation, the proposed approach offers a theoretically grounded framework for developing more stable and generalisable optimisation algorithms.

The main content of this work is outlined in Figure 1. The left part of Figure 1 highlights the advantages of cascaded inertia SGD (CISGD) over existing optimisers by visualisation. The middle part details the update pipeline of CISGD, demonstrating how historical gradients are accumulated into multi-order momentum terms and how the four coefficients are learnt on-the-fly, eliminating manual tuning. The right part summarises the experimental gains.

The main contributions of this work are presented as follows:

- We propose CISGD, an optimiser that explicitly aggregates historical gradients to construct multi-order momentum terms, enabling effective suppression of gradient noise and improved convergence to flatter minima.
- We design a lightweight feed-forward NN (FNN) for learning the hyperparameters of the optimiser, alleviating manual tuning and enhancing robustness across architectures and tasks.
- We provide theoretical analyses of CISGD and comprehensive experiments on image classification and remote sensing change detection demonstrating its practical effectiveness.

The remainder of this work is organised as follows. Section 2 reviews optimisation in deep learning. Section 3 presents the CISGD algorithm and hyperparameter learning strategy, followed by convergence analysis. Section 4 reports experimental results on image classification and remote sensing change detection. Section 5 concludes the paper.

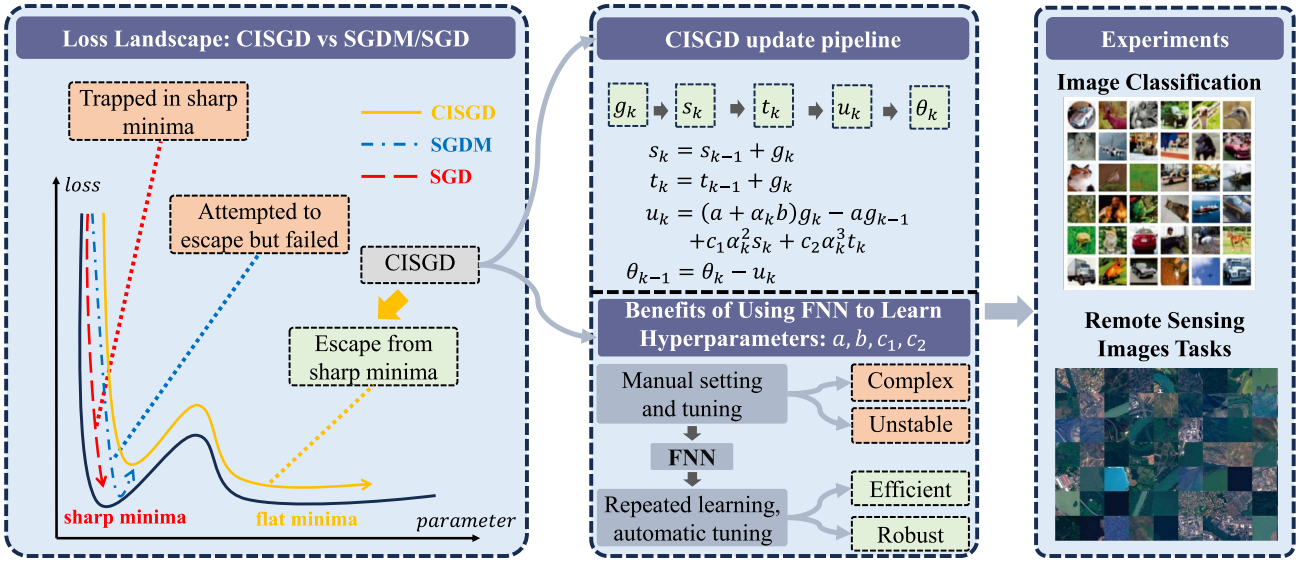


FIGURE 1 | Overview of the research roadmap of this paper.

2 | Preliminaries

This section provides a concise review of optimisation in deep learning.

2.1 | Optimisation in Deep Learning

In deep learning, the optimisation problem is typically defined as finding a set of optimal model parameters that minimise a given loss function. Specifically, given a training dataset $D = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, the goal is to find a parameter set θ that minimises the following loss function:

$$\mathcal{L}(\theta) = \frac{1}{|S|}\sum_{i=1}^{n} l(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta), \tag{1}$$

where S is a subset of D; n is the total number of samples. In mini-batch SGD, S is a random subset of D with size m; $\mathcal{L}$ is the loss function; $\boldsymbol{x}_i$ is the input vector, $\boldsymbol{y}_i$ is the label, and θ is the weight vector of a network. In the early stages of deep learning development, the SGD optimiser stands as a foundational optimisation algorithm, and its parameter update formula is as follows:

$$\theta_{k+1} = \theta_k - \alpha\nabla\mathcal{L}(\theta_k), \tag{2}$$

where α represents the learning rate; $\nabla\mathcal{L}(\theta)$ is the gradient with respect to the parameter θ. We introduce a grid of points $t = t_k$, where $t_k = t_0 + k\tau, k = 0, ..., \hat{k}$ and $\hat{k}$ is number of steps. Assuming that θ_k comes from its continuous function and that the points of each iteration are sampled from multiples of α, θ_k can be written as $\theta(t)$. Let $t = k\alpha$, Equation (2) can be written as:

$$\theta(t+\alpha) = \theta(t) - \alpha\nabla\mathcal{L}(\theta(t)), \tag{3}$$

If the learning rate α is sufficiently small, we obtain the continuous form of SGD as follows:

$$\dot{\theta}(t) = -\nabla L(\theta(t)). \tag{4}$$

Equation (4) can be viewed as an ordinary differential equation (ODE). The trajectory of θ_k is obtained by solving this ODE called the gradient flow.

Similarly, the proposed CISGD extends this mechanism within the optimiser design, achieving gradient-noise suppression and improved generalisation by explicitly accumulating historical gradients. Traditional adaptive optimisers such as SGDM and Adam maintain only first-order momentum or second-order moment estimates [26, 27]. In contrast, the proposed method explicitly constructs multi-order integration terms to accumulate the full history of gradients, providing theoretical guarantees for gradient noise suppression.

3 | Methodology

This section presents the CISGD optimiser and an FNN-based strategy to learn its hyperparameters, followed by a convergence analysis.

3.1 | CISGD Optimiser

Rewriting above equation and using the learning rate α_k instead of τ, we obtain a new optimiser CISGD. Starting from the iterative relations and introducing step-size α_k, we write CISGD as

$$\begin{cases} \boldsymbol{s}_k = \boldsymbol{s}_{k-1} + \boldsymbol{g}_k \\ \boldsymbol{t}_k = \boldsymbol{t}_{k-1} + \boldsymbol{s}_k \\ \boldsymbol{u}_k = (a + \alpha_k b)\boldsymbol{g}_k - a\boldsymbol{g}_{k-1} + c_1\alpha_k^2\boldsymbol{s}_k + c_2\alpha_k^3\boldsymbol{t}_k \\ \theta_{k+1} = \theta_k - \boldsymbol{u}_k. \end{cases} \tag{5}$$

where $\boldsymbol{g}_k = \nabla L(\theta_k)$ denotes the stochastic gradient at step k; $\boldsymbol{s}_k = \sum_{i=0}^{k}\boldsymbol{g}_i$ accumulates first-order historical gradients, and $\boldsymbol{t}_k = \sum_{i=0}^{k}\boldsymbol{s}_i$ accumulates second-order historical gradients; $\boldsymbol{u}_k$ combines instantaneous and historical signals. The pseudo-code of CISGD is given in Algorithm 1. CISGD encompasses standard SGD as a limiting case when $a = 1$ and $b = c_1 = c_2 = 0$. The coefficients play distinct roles: a and b control the instantaneous and first-order inertial updates, whereas c_1 and c_2 weight the first- and second-order accumulated gradients, acting as a multi-scale filter to suppress high-frequency noise and bias optimisation towards flatter minima. Clearly, the CISGD has multiple hyperparameters, and determining the optimal values for these hyperparameters is a topic worthy of discussion.

ALGORITHM 1 | CISGD

Require: Step sizes α_k, iterative number $\hat{k}$, coefficients a, b, c_1, c_2.
Initial parameter θ.
Initial $\boldsymbol{s}_0 = 0, \boldsymbol{t}_0 = 0$.
Ensure: Optimal θ
for $k = 0$ **to** $\hat{k}$ **do**
  $k = k + 1$
  $\boldsymbol{g}_k \leftarrow \nabla_\theta L_k(\theta_{k-1})$
  $\boldsymbol{s}_k \leftarrow \boldsymbol{s}_{k-1} + \boldsymbol{g}_k$
  $\boldsymbol{t}_k \leftarrow \boldsymbol{t}_{k-1} + \boldsymbol{s}_k$
  if $k = 1$ **then**
    $\theta_k \leftarrow \theta_{k-1} - \alpha_k\boldsymbol{g}_k$
  else
    $\boldsymbol{u}_k = (a + \alpha_k b)\boldsymbol{g}_k - a\boldsymbol{g}_{k-1} + c_1\alpha_k^2\boldsymbol{s}_k + c_2\alpha_k^3\boldsymbol{t}_k$
    $\theta_k = \theta_{k-1} - \boldsymbol{u}_k$
  end if
end for

3.2 | Learnt Hyperparameters in CISGD

In light of the tuning complexity introduced by the four hyperparameters in CISGD during deep learning training, we propose an FNN to automatically learn these coefficients. To provide a principled way of setting the four coefficients, we borrow inspiration from a noise-suppressing neural-dynamics framework. Specifically, denoting I^n as the n-fold integral, we consider a double integration-enhanced neural-dynamics model (DIEND) [28–30]:

$$a\dot{\delta}(t) + b\delta(t) + c_1 I^1 \delta(t) + c_2 I^2 \delta(t) - \sigma(t) = 0. \tag{6}$$

The objective of DIEND is to converge the error $\delta(t)$ to zero by means of Equation (6). The negative gradient in the deep learning process is analogous to the error in the control system. The goal of the optimiser is to minimise the gradients of the loss function, a process analogous to the convergence of error to zero by DIEND. The DIEND model serves as the theoretical foundation for our optimiser. Intuitively, we map its dynamics to the optimisation process: the term with coefficient a responds to instantaneous gradients, b accumulates first-order gradient history, and terms with c_1 and c_2 capture higher-order integration effects. This hierarchical structure allows the optimiser to suppress high-frequency noise while maintaining responsiveness.

In addition, the theory of DIEND posits that the integral term can suppress the noise $\sigma(t)$ so that the error function $\delta(t)$ converges to zero. Assuming that the rate of change of the network parameters and the negative gradient in the optimiser correspond to the noise and the error in the neural dynamics, respectively, then the gradients are guaranteed to converge to zero. Define $\boldsymbol{g}(t) = \nabla\mathcal{L}(\theta(t))$. Let $\delta(t) = -\boldsymbol{g}(t)$, $\dot{\theta}(t) = \sigma(t)$, rewrite Equation (6) as follows:

$$-a\dot{\boldsymbol{g}}(t) = b\boldsymbol{g}(t) + c_1 I^1 \boldsymbol{g}(t) + c_2 I^2 \boldsymbol{g}(t) + \dot{\theta}(t). \tag{7}$$

rearranging Equation (6), we get the $\dot{\theta}(t)$ as follows:

$$\dot{\theta}(t) = -a\dot{\boldsymbol{g}}(t) - b\boldsymbol{g}(t) - c_1 I^1 \boldsymbol{g}(t) - c_2 I^2 \boldsymbol{g}(t). \tag{8}$$

CISGD update formula could be transformed as follows:

$$\begin{aligned}\boldsymbol{u}_k = a(\boldsymbol{g}_k - \boldsymbol{g}_{k-1}) + \tau b\boldsymbol{g}_k + c_1\tau^2 \sum_{i=0}^{k} \boldsymbol{g}_i \\ + c_2\tau^3 \sum_{i=0}^{k}\sum_{j=0}^{i} \boldsymbol{g}_j.\end{aligned} \tag{9}$$

Consider the update formula of CISGD over a continuous time interval τ, which can be expressed as

$$\boldsymbol{u}_k = a\tau\dot{\boldsymbol{g}}(t) + \tau b\boldsymbol{g}_k + c_1\tau I^1 \boldsymbol{g}(t) + c_2\tau I^2 \boldsymbol{g}(t). \tag{10}$$

From the discrete parameter update rule of CISGD, the discrete parameter variation can be derived as $\theta_{k+1} - \theta_k = -u_k$. According to the definition of the continuous update rate Equation (2), substituting the above equation into it yields $\dot{\theta}(t) = \lim_{\tau\to 0}\frac{-u_k}{\tau}$. This can be rearranged to obtain the continuous approximation of the discrete update term:

$$u_k \approx -\tau\dot{\theta}(t) \quad (\tau \to 0). \tag{11}$$

Note that the time constant τ is effectively absorbed into the learning rate α and the coefficients during discretisation, ensuring dimensional consistency between the continuous ODE analysis and the discrete update rule. By combining Equations (8), (9) and (11), it can be concluded that the update rule of CISGD is fully consistent with a DIEND system.

The relationship between deep NN architectures and discretisation of differential equations is explored by Weinan et al. in [31, 32]. For example, for the dynamical system $\dot{\boldsymbol{z}} = \mathcal{F}(t, \boldsymbol{z}(t))$, its discretisation, which is the Euler discretisation, corresponds to the following deep NN structure:

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + h\mathcal{F}(t_k, \boldsymbol{z}_k), \tag{12}$$

where $\boldsymbol{z}_{k+1}$ is the feature of the k-th layer of the network that corresponds to the state in the dynamical system. The Equation (12) is known as the residual connection in ResNet [33].

For Equation (8), since θ corresponds to the noise in the DIEND system, assuming that the noise in the dynamic system is time-varying linear noise and the rate of change of the network weight θ does not exceed linearity, that is, $\dot{\theta}(t) = \boldsymbol{u}t$, we have

$$\boldsymbol{u}t = -a\dot{\boldsymbol{g}}(t) - b\boldsymbol{g}(t) - c_1 I^1 \boldsymbol{g}(t) - c_2 I^2 \boldsymbol{g}(t). \tag{13}$$

After discretising the aforementioned equation and setting aside the term $o(\tau^2)$, we have

$$\begin{aligned}\boldsymbol{u}k\tau = -a\frac{\boldsymbol{g}_k - \boldsymbol{g}_{k-1}}{\tau} - b\boldsymbol{g}_k - c_1\tau\sum_{i=0}^{k}\boldsymbol{g}_i \\ - c_2\tau^2\sum_{i=0}^{k}\sum_{j=0}^{i}\boldsymbol{g}_j.\end{aligned} \tag{14}$$

Let $\beta = a + b\tau + c_1\tau^2 + c_2\tau^3$ and rearranging Equation (14), the iterative update formula for $\boldsymbol{g}$ is derived as follows:

$$\begin{aligned}\boldsymbol{g}_k = \frac{a}{\beta}\boldsymbol{g}_{k-1} - \frac{c_1\tau^2}{\beta}\sum_{i=0}^{k-1}\boldsymbol{g}_i \\ - \frac{c_2\tau^3}{\beta}\sum_{i=0}^{k-1}\sum_{j=0}^{i}\boldsymbol{g}_j - \frac{\boldsymbol{u}\tau^2}{\beta}k.\end{aligned} \tag{15}$$

In a manner analogous to residual connectivity in ResNet, the propagation formula employed for the design of the FNN is hereby presented:

$$\begin{aligned}\boldsymbol{z}_{k+1} = \frac{a}{\beta}\boldsymbol{z}_k - \frac{c_1\tau^2}{\beta}\sum_{i=0}^{k}\boldsymbol{z}_i \\ - \frac{c_2\tau^3}{\beta}\sum_{i=0}^{k}\sum_{j=0}^{i}\boldsymbol{z}_j - \frac{\boldsymbol{u}\tau^2}{\beta}(k+1).\end{aligned} \tag{16}$$

The structure of the specific FNN corresponding to Equation (16) is illustrated in Figure 2, where $\boldsymbol{m}_k$ denotes $\sum_{i=0}^{k}\boldsymbol{z}_i$ and $\boldsymbol{v}_k$ denotes $\sum_{i=0}^{k}\sum_{j=0}^{i}\boldsymbol{z}_j$. This FNN comprises multiple layers of identical blocks. Each block is equipped with three inputs and outputs, which corresponds to each iteration of Equation (16).

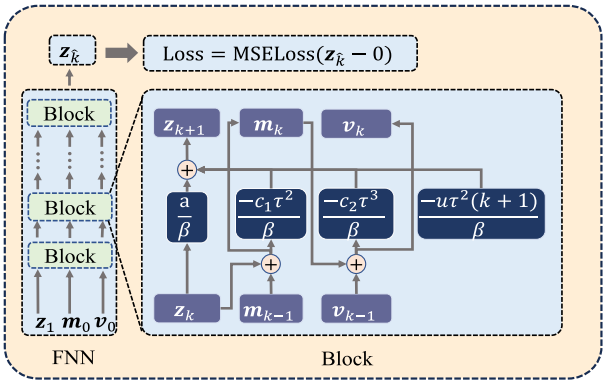


FIGURE 2 | An architecture of FNN designed to learn coefficients, where a, b, c_1, c_2 are set as learnable parameters during network training.

The mean square error loss function is employed. The input data consist of a set of random numbers drawn from a standard normal distribution with a mean of 0 and a variance of 10. The label corresponding to this input data is designated as $\mathbf{0}$.

It has been established that the input of the FNN corresponds to the gradient in the training process of the deep learning model, as well as the error in the DIEND system. The objective of this FNN is to make its output, $\boldsymbol{z}_{\hat{k}}$, equal to zero. In the FNN, a, b, c_1, and c_2 are initialised as trainable parameters with initial values of $1, 0, 0$, and 0, respectively. Under this initialisation, CISGD degenerates to standard SGD. Subsequent to the training process, the hyperparameters that have been obtained can demonstrate enhanced generalisability beyond the confines of the training cases that were specific to the training process. The objective is to ensure that the optimisation algorithm is capable of performing robustly across a range of deep learning models and real-world application scenarios, rather than being over tailored to a single dataset or task. The FNN is lightweight and is trained offline using random noise input. Training on random noise ensures that the learnt coefficients capture intrinsic robust optimisation dynamics based on the statistical properties of gradient noise, rather than being biased by specific dataset features. Since the FNN architecture is orders of magnitude smaller than standard backbone networks and the learnt coefficients are frozen after the offline phase, they are transferred to downstream tasks without re-tuning. Consequently, the proposed method introduces effectively zero computational overhead during the main model training.

Through the training of the designed FNN, four hyperparameters a, b, c_1, and c_2 in the CISGD are obtained as $0.288, 1.476, 0.062$, and 0.062, respectively.

3.3 | Convergence Analysis

This section provides a convergence analysis to validate the effectiveness of the CISGD optimiser and demonstrate that its gradient converges exponentially to zero on a global scale.

Theorem 1. *The gradient $\mathbf{g}_k$ of CISGD as the error function in the DIEND model (6) converges globally to zero.*

Proof. Starting from Equation (15) with $o(\tau^2)$ and replacing k with $k-1$ in the Equation (15), we have

$$\begin{aligned}\boldsymbol{g}_{k-1} = \frac{a}{\beta}\boldsymbol{g}_{k-2} - \frac{c_1\tau^2}{\beta}\sum_{i=0}^{k-2}\boldsymbol{g}_i \\ - \frac{c_2\tau^3}{\beta}\sum_{i=0}^{k-2}\sum_{j=0}^{i}\boldsymbol{g}_j - \frac{\boldsymbol{u}\tau^2}{\beta}(k-1) + \boldsymbol{o}(\tau^2).\end{aligned} \tag{17}$$

Using Equation (15) with $o(\tau^2)$ minus Equation (17), we have

$$\begin{aligned}\left(1+\frac{c_1\tau^2}{\beta}\right)\boldsymbol{g}_k = \left(1+\frac{a}{\beta}\right)\boldsymbol{g}_{k-1} - \frac{a}{\beta}\boldsymbol{g}_{k-2} \\ - \frac{c_2\tau^3}{\beta}\sum_{j=0}^{k}\boldsymbol{g}_j - 1 + \boldsymbol{o}(\tau^2).\end{aligned} \tag{18}$$

Replacing k with $k-1$ in the above equation, we have

$$\begin{aligned}\left(1+\frac{c_1\tau^2}{\beta}\right)\boldsymbol{g}_{k-1} = \left(1+\frac{a}{\beta}\right)\boldsymbol{g}_{k-2} - \frac{a}{\beta}\boldsymbol{g}_{k-3} \\ - \frac{c_2\tau^3}{\beta}\sum_{j=0}^{k-1}\boldsymbol{g}_j - 1 + \boldsymbol{o}(\tau^2).\end{aligned} \tag{19}$$

Using Equation (18) minus Equation (19), we have

$$\begin{aligned}\mathbf{0} = \left(1+\frac{c_1\tau^2+c_2\tau^3}{\beta}\right)\boldsymbol{g}_k - \left(2+\frac{c_1\tau^2+a}{\beta}\right)\boldsymbol{g}_{k-1} \\ + \left(1+\frac{2a}{\beta}\right)\boldsymbol{g}_{k-2} - \frac{a}{\beta}\boldsymbol{g}_{k-3} + \boldsymbol{o}(\tau^2).\end{aligned} \tag{20}$$

Let $p = 1 + \frac{c_1\tau^2+c_2\tau^3}{\beta}, q = 2 + \frac{c_1\tau^2+a}{\beta}, r = 1 + \frac{2a}{\beta}, s = \frac{a}{\beta}$, we have

$$p\boldsymbol{g}_k - q\boldsymbol{g}_{k-1} + r\boldsymbol{g}_{k-2} - s\boldsymbol{g}_{k-3} + \boldsymbol{o}(\tau^2) = \mathbf{0}. \tag{21}$$

Rewriting the above equation in the form of a matrix product, we have

$$\begin{bmatrix}\boldsymbol{g}_k \\ \boldsymbol{g}_{k-1} \\ \boldsymbol{g}_{k-2}\end{bmatrix} = \begin{bmatrix}\frac{q}{p} & \frac{-r}{p} & \frac{s}{p} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{g}_{k-1} \\ \boldsymbol{g}_{k-2} \\ \boldsymbol{g}_{k-3}\end{bmatrix}. \tag{22}$$

Let $\boldsymbol{\varphi}_k = \left[\boldsymbol{g}_k^T, \boldsymbol{g}_{k-1}^T, \boldsymbol{g}_{k-2}^T\right]^T$. The state space equation of the above equation can be described as

$$\boldsymbol{\varphi}_k = \boldsymbol{M}\boldsymbol{\varphi}_{k-1} + \boldsymbol{o}(\tau^2), \tag{23}$$

where $\boldsymbol{M}$ is defined as

$$\begin{bmatrix}\frac{q}{p} & \frac{-r}{p} & \frac{s}{p} \\ \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0}\end{bmatrix}. \tag{24}$$

According to the triangle inequality theorem, we have

$$\begin{aligned}\|\boldsymbol{\varphi}_k\|_2 &\le \|\boldsymbol{M}\boldsymbol{\varphi}_{k-1}\|_2 + \|\boldsymbol{o}(\tau^2)\|_2 \\ &= \|\boldsymbol{M}\boldsymbol{\varphi}_{k-1}\|_2 + \boldsymbol{o}(\tau^2) \\ &\le \|\boldsymbol{M}\|_2\|\boldsymbol{M}\boldsymbol{\varphi}_{k-1}\|_2 + \boldsymbol{o}(\tau^2) \\ &\le \|\boldsymbol{M}\|_2\|\boldsymbol{M}\boldsymbol{\varphi}_{k-2}\|_2 + \|\boldsymbol{M}\|_2\boldsymbol{o}(\tau^2) + \boldsymbol{o}(\tau^2) \\ &= \|\boldsymbol{M}\|_2\|\boldsymbol{M}\boldsymbol{\varphi}_{k-2}\|_2 + \boldsymbol{o}(\tau^2) \\ &\le \|\boldsymbol{M}\|_2^2\|\boldsymbol{\varphi}_{k-2}\|_2 + \boldsymbol{o}(\tau^2) \\ &\vdots \\ &\le \|\boldsymbol{M}\|_2^{k-1}\|\boldsymbol{\varphi}_1\|_2 + \boldsymbol{o}(\tau^2).\end{aligned} \tag{25}$$

Assuming that $1 < a < 2, 2 < b < 3, 2 < c < 3$, and $0 < d < 1$, the absolute value of the eigenvalues of matrix $\boldsymbol{M}$ are less than 1. Then, we have $\lim_{k\to\infty}\|\boldsymbol{M}\|_2^k = 0$ and

$$\|\boldsymbol{\varphi}_k\|_2 \le \lim_{k\to\infty}\|\boldsymbol{M}\|_2^{k-1}\|\boldsymbol{\varphi}_1\|_2 + \boldsymbol{o}(\tau^2) = \boldsymbol{o}(\tau^2). \tag{26}$$

In light of the definitions of $\boldsymbol{\varphi}_k$ and $\boldsymbol{g}_k$, we can conclude that the CISGD is convergent. The proof is thus completed. □

4 | Experiments

In this section, to demonstrate the performance of the proposed CISGD optimiser, we conduct experiments on image classification and remote sensing change detection. The code is based on the PyTorch framework, and all experiments are conducted on a server equipped with 2 NVIDIA RTX 8000 GPUs and 1 RTX 3060 laptop GPU.

4.1 | Image Classification

To evaluate the generalisation of the CISGD, extensive experiments were conducted on image classification tasks utilising the CIFAR-10 and CIFAR-100 [34] benchmark datasets. The ResNet18 [33], ResNet34, ResNet50, VGG11 [35], and DenseNet121 [36] network architectures are selected for training on the CIFAR datasets from scratch. The classification performance of CISGD is compared with those of optimisers such as SGDM, PIDO, Adam, Adai, and Adan. The training epoch is set to 100, and the batch size is configured to 128. To ensure a fair comparison, learning rates were tuned for each optimiser based on its specific characteristics and requirements. The learning rates of CISGD, SGDM, PIDO, Adam, Adai, and Adan are 0.1, 0.1, 0.01, 0.0001, 1, and 0.01, respectively. The mean and standard deviation of test accuracy over five runs are reported in Table 1. We apply standard data augmentation (random crop, horizontal flip, normalisation) and a cosine annealing learning rate scheduler.

As shown in Table 1, CISGD consistently improves test accuracy over SGDM across multiple architectures, and performs on par with adaptive baselines.

4.2 | Change Detection of Remote Sensing Images

We further evaluate CISGD on a remote sensing change detection task. Remote sensing change detection aims to identify changes across bitemporal images acquired in the same geographic area. Changes include the appearance or disappearance of objects and background modifications. The global very-high-resolution landslide mapping (GVLM) dataset [37] is chosen for the experiments. The GVLM dataset includes 17 bitemporal very-high-resolution imagery pairs with a spatial

TABLE 1 | Test accuracies (mean ± standard deviation) on CIFAR-10 and CIFAR-100 for different optimisers with different network models.

Model	Optimiser	CIFAR-10 (%)	CIFAR-100 (%)	Model	Optimiser	CIFAR-10 (%)	CIFAR-100 (%)
ResNet18	SGDM	$94.77_{\pm 0.13}$	$74.95_{\pm 0.23}$	ResNet34	SGDM	$95.27_{\pm 0.17}$	$77.31_{\pm 0.34}$
	PIDO	$94.69_{\pm 0.42}$	$76.62_{\pm 0.28}$		PIDO	$95.24_{\pm 0.27}$	$77.58_{\pm 0.55}$
	Adan	$93.88_{\pm 0.17}$	$75.59_{\pm 0.86}$		Adan	$94.46_{\pm 0.28}$	$74.79_{\pm 1.11}$
	Adai	$94.40_{\pm 0.15}$	$75.97_{\pm 0.70}$		Adai	$95.28_{\pm 0.17}$	$75.94_{\pm 0.13}$
	CISGD	$\mathbf{94.96_{\pm 0.20}}$	$\mathbf{76.66_{\pm 0.37}}$		CISGD	$\mathbf{95.29_{\pm 0.11}}$	$\mathbf{77.81_{\pm 0.55}}$
ResNet50	SGDM	$94.79_{\pm 0.12}$	$77.31_{\pm 0.34}$	VGG16	SGDM	$92.73_{\pm 0.12}$	$69.99_{\pm 0.34}$
	PIDO	$94.90_{\pm 0.28}$	$77.58_{\pm 0.55}$		PIDO	$93.15_{\pm 0.33}$	$70.30_{\pm 0.63}$
	Adan	$94.55_{\pm 0.11}$	$74.79_{\pm 1.11}$		Adan	$92.65_{\pm 0.32}$	$70.03_{\pm 0.47}$
	Adai	$94.55_{\pm 0.12}$	$75.94_{\pm 0.13}$		Adai	$93.53_{\pm 0.16}$	$70.11_{\pm 0.35}$
	CISGD	$\mathbf{94.96_{\pm 0.15}}$	$\mathbf{77.81_{\pm 0.55}}$		CISGD	$\mathbf{93.55_{\pm 0.12}}$	$\mathbf{70.51_{\pm 0.28}}$
DenseNet121	SGDM	$94.94_{\pm 0.13}$	$76.65_{\pm 0.16}$	WRN28_10	SGDM	$94.10_{\pm 0.12}$	$78.33_{\pm 0.03}$
	PIDO	$94.12_{\pm 0.14}$	$76.02_{\pm 0.14}$		PIDO	$95.17_{\pm 0.46}$	$78.63_{\pm 0.29}$
	Adan	$94.67_{\pm 0.07}$	$76.91_{\pm 0.23}$		Adan	$92.18_{\pm 0.22}$	$76.31_{\pm 0.34}$
	Adai	$94.53_{\pm 0.16}$	$76.88_{\pm 0.37}$		Adai	$93.89_{\pm 0.49}$	$78.86_{\pm 0.25}$
	CISGD	$\mathbf{95.05_{\pm 0.17}}$	$\mathbf{77.51_{\pm 0.34}}$		CISGD	$\mathbf{95.41_{\pm 0.32}}$	$\mathbf{78.64_{\pm 0.13}}$

Note: The first best results are highlighted in bold, respectively.

resolution of 0.59 m acquired via Google Earth service. Each sub-dataset contains a pair of bitemporal images and the corresponding ground-truth map. The dataset is split into 12 training, 2 validation, and 3 testing pairs. No geometric augmentation is applied. The double-attention network (A2Net) [38] and the bitemporal image transformer (BIT) model [39] are chosen as benchmarks for evaluating generalisation.

The experiments involve training the BIT and A2Net models on the GVLM dataset using a variety of optimisers, including SGDM, PIDO, Adam, Adan, Adai, and CISGD. These optimisers are evaluated under identical experimental conditions, with hyperparameters such as learning rate and batch size tuned based on initial trials. Performance is measured using standard metrics, including overall accuracy (OA), precision, mean $F1$-score (m$F1$), and change-class intersection over union (cIoU), which together capture detection accuracy and overall classification performance. The training epoch is set to 100 and the batch size is 16. The initial learning rate of the optimisers is set to be the same as that used in the image classification experiments. Table 2 summarises the results of different optimisers using the BIT and A2Net models on the GVLM test set. The results indicate that CISGD outperforms SGDM and achieves performance comparable to other adaptive optimisers. Figures 3 and 4 provide visual comparisons of results across optimisers on BIT and A2Net, respectively.

5 | Conclusion

This paper has introduced CISGD, an optimisation method that aggregates historical gradients to form multi-order momentum terms for improved stability and generalisation. To alleviate manual hyperparameter tuning, we have developed an FNN-based strategy that learns the optimiser's coefficients, enhancing robustness across models and tasks. We have

TABLE 2 | Comparisons in terms of accuracy, m$F1$, precision, recall, and cIoU on GVLM dataset using BIT and A2Net models.

Model	Optimiser	OA	m$F1$	Precision	Recall	cIoU
BIT	SGDM	0.9822	0.8346	0.8528	0.8174	0.7161
	PIDO	0.9855	0.8715	0.8478	<u>0.8967</u>	0.7723
	Adam	0.9841	0.8768	<u>0.8974</u>	0.8572	0.7807
	Adan	**0.993**	**0.9361**	**0.935**	**0.9372**	**0.8799**
	Adai	<u>0.9848</u>	<u>0.8595</u>	0.8748	0.8447	<u>0.7536</u>
	CISGD	0.9868	0.8794	0.8818	0.877	0.7847
A2Net	SGDM	0.9765	0.7388	<u>0.9459</u>	0.6061	0.5858
	PIDO	0.9734	0.7006	0.9174	0.6541	0.5391
	Adam	0.9776	**0.8114**	0.9156	**0.7285**	**0.6827**
	Adan	0.9734	0.7006	0.9174	0.5666	0.5391
	Adai	<u>0.9795</u>	0.7797	**0.9497**	0.6613	0.6389
	CISGD	**0.9813**	<u>0.8043</u>	0.9455	<u>0.6998</u>	<u>0.6727</u>

Note: The first and second best results are highlighted in bold and underline, respectively.

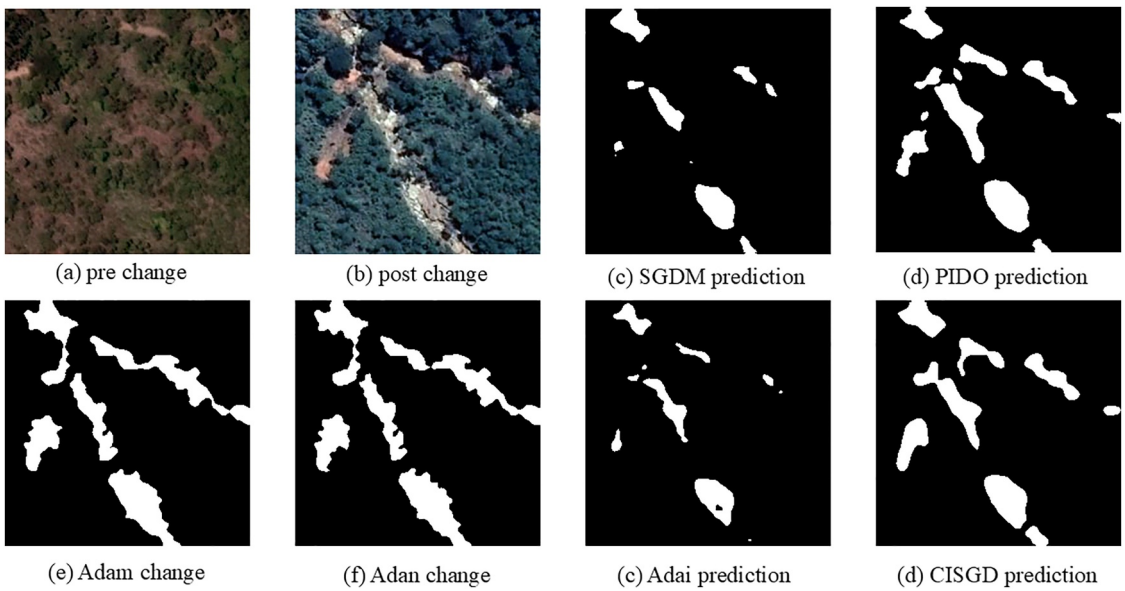


FIGURE 3 | Visual comparisons of the CISGD and the other optimisers using BIT model on the GVLM dataset.

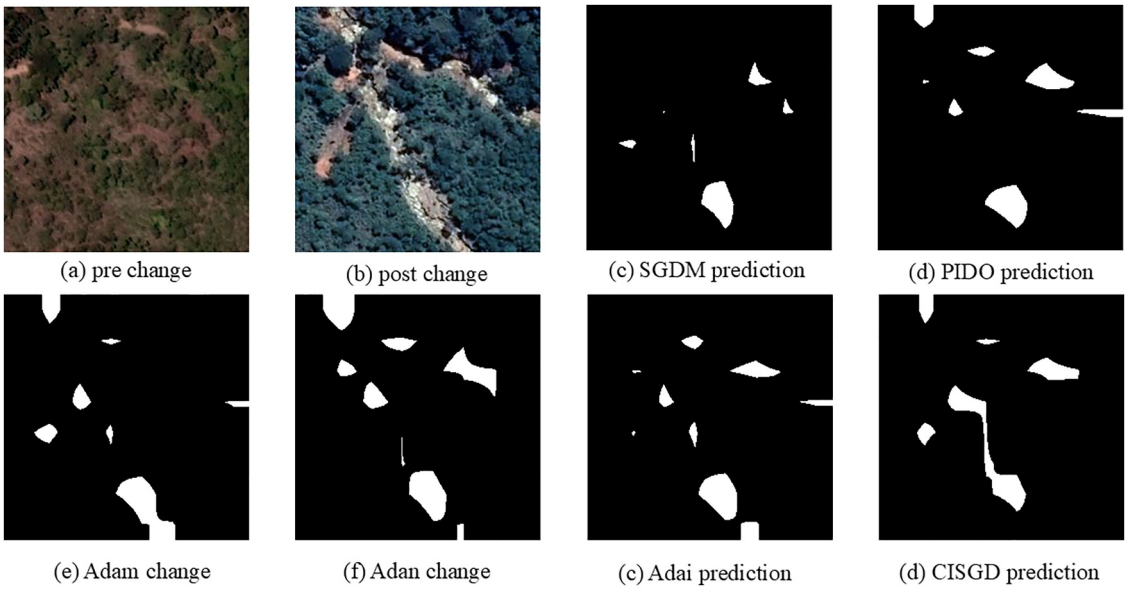


FIGURE 4 | Visual comparisons of the CISGD and the other optimisers using A2Net model on the GVLM dataset.

provided a convergence analysis establishing global gradient convergence under mild conditions. Experiments on image classification and remote sensing change detection have shown that CISGD consistently surpasses SGDM and performs competitively with adaptive optimisers. These results highlight the benefit of leveraging historical gradient information within the optimiser design. Although our empirical validation is currently focused on vision benchmarks and remote sensing tasks, the conceptual contribution of integrating multi-order inertia with automatically learnt coefficients offers a promising direction for developing more robust optimisation algorithms. Future work will include deeper robustness analysis and extensions to broader application domains.

Funding

This work was supported by the National Natural Science Foundation of China under Grants 62476115 and 62506148, in part by the Fundamental Research Funds for the Central Universities under Grant lzujbky-2025-pd05, in part by the Postdoctoral Fellowship Programme (Grade C) of China Postdoctoral Science Foundation under Grant GZC20251039, and in part by the Supercomputing Center of Lanzhou University.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The complete data and code will be released after the anonymous review is completed.

References

1. I. Corley, C. Robinson, and A. Ortiz, "A Change Detection Reality Check," preprint, arXiv:2402.06994 (2024).

2. Y. Zhao, T. Celik, N. Liu, F. Gao, and H. C. Li, "SSLChange: A Self-Supervised Change Detection Framework Based on Domain Adaptation," *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024): 1–14, https://doi.org/10.1109/tgrs.2024.3489615.

3. D. Yu and S. Ji, "A New Spatial-Oriented Object Detection Framework for Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022): 1–16, https://doi.org/10.1109/tgrs.2021.3127232.

4. X. Ma, Q. Wu, X. Zhao, X. Zhang, M. O. Pun, and B. Huang, "SAM-Assisted Remote Sensing Imagery Semantic Segmentation With Object and Boundary Constraints," *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024): 1–16, https://doi.org/10.1109/tgrs.2024.3443420.

5. Y. Zhao, S. Li, C. H. Liu, Y. Han, H. Shi, and W. Li, "Domain Adaptive Remote Sensing Scene Recognition via Semantic Relationship Knowledge Transfer," *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023): 1–13, https://doi.org/10.1109/tgrs.2023.3267149.

6. Z. Lv, P. Zhang, W. Sun, J. A. Benediktsson, J. Li, and W. Wang, "Novel Adaptive Region Spectral–Spatial Features for Land Cover Classification With High Spatial Resolution Remotely Sensed Imagery," *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023): 1–12, https://doi.org/10.1109/tgrs.2023.3275753.

7. J. Yang, X. Li, I. Fatkhullin, and N. He, "Two Sides of One Coin: The Limits of Untuned SGD and the Power of Adaptive Methods," in *Advances in Neural Information Processing Systems* (2023), 74257–74288.

8. Z. Lin, S. Zhang, Y. Zhou, H. Wang, and S. Wang, "Learning Rate Burst for Superior SGDM and AdamW Integration," *Journal of Intelligent and Fuzzy Systems* 49, no. 2 (2024): 1–11, https://doi.org/10.3233/jifs-239157.

9. M. Liu, L. Chen, X. Du, L. Jin, and M. Shang, "Activated Gradients for Deep Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems* 34, no. 4 (April 2023): 2156–2168, https://doi.org/10.1109/tnnls.2021.3106044.

10. Z. Chen, J. Zhang, Y. Kou, X. Chen, C. J. Hsieh, and Q. Gu, "Why Does Sharpness–Aware Minimization Generalize Better than SGD?," in *Advances in Neural Information Processing Systems* (2023), 72325–72376.

11. J. Fan, L. Jin, P. Li, J. Liu, Z. G. Wu, and W. Chen, "Coevolutionary Neural Dynamics Considering Multiple Strategies for Nonconvex Optimization," *Tsinghua Science and Technology* (2025), https://doi.org/10.26599/TST.2025.9010120.

12. H. Li, A. Rakhlin, and A. Jadbabaie, "Convergence of Adam Under Relaxed Assumptions," in *Advances in Neural Information Processing Systems* (2023), 52166–52196.

13. D. Friedman, A. Wettig, and D. Chen, "Learning Transformer Programs," in *Advances in Neural Information Processing Systems* (2023), 49044–49067.

14. B. Hoover, Y. Liang, B. Pham, et al., "Energy Transformer," in *Advances in Neural Information Processing Systems* (2023), 27532–27559.

15. J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced Transformer With Rotary Position Embedding," *Neurocomputing* 568 (2024): 12706, https://doi.org/10.1016/j.neucom.2023.127063.

16. Y. Ye, M. Wang, L. Zhou, G. Lei, J. Fan, and Y. Qin, "Adjacent-Level Feature Cross-Fusion With 3-D CNN for Remote Sensing Image Change Detection," *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023): 1–14, https://doi.org/10.1109/tgrs.2023.3305499.

17. C. Han, C. Wu, H. Guo, M. Hu, J. Li, and H. Chen, "Change Guiding Network: Incorporating Change Prior to Guide Change Detection in Remote Sensing Imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 16 (2023): 8395–8407, https://doi.org/10.1109/jstars.2023.3310208.

18. W. Yu, X. Zhang, R. Gloaguen, X. X. Zhu, and P. Ghamisi, "MineNetCD: A Benchmark for Global Mining Change Detection on Remote Sensing Imagery," *IEEE Transactions on Geoscience and Remote Sensing* 62 (2024): 1–16, https://doi.org/10.1109/tgrs.2024.3491715.

19. P. Zhou, J. Feng, C. Ma, C. Xiong, and S. C. H. Hoi, "Towards Theoretically Understanding Why SGD Generalizes Better Than Adam in Deep Learning," in *Advances in Neural Information Processing Systems* (2020), 21285–21296.

20. Z. Xie, X. Wang, H. Zhang, I. Sato, and M. Sugiyama, "Adaptive Inertia: Disentangling the Effects of Adaptive Learning Rate and Momentum," in *International Conference on Machine Learning*, Vol. 162 (July 2022), 24430–24459.

21. X. Xie, P. Zhou, H. Li, Z. Lin, and S. Yan, "Adan: Adaptive Nesterov Momentum Algorithm for Faster Optimizing Deep Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, no. 12 (December 2024): 9508–9520, https://doi.org/10.1109/tpami.2024.3423382.

22. L. Jin, Y. Zhang, and S. Li, "Integration-Enhanced Zhang Neural Network for Real-Time-Varying Matrix Inversion in the Presence of Various Kinds of Noises," *IEEE Transactions on Neural Networks and Learning Systems* 27, no. 12 (December 2016): 2615–2627, https://doi.org/10.1109/tnnls.2015.2497715.

23. L. Jin, J. Zhao, L. Chen, and S. Li, "Collective Neural Dynamics for Sparse Motion Planning of Redundant Manipulators Without Hessian Matrix Inversion," *IEEE Transactions on Neural Networks and Learning Systems* 36, no. 3 (2025): 4326–4335, https://doi.org/10.1109/tnnls.2024.3363241.

24. D. Su, P. S. Stanimirovic, L. B. Han, and L. Jin, "Neural Dynamics for Improving Optimiser in Deep Learning With Noise Considered," *CAAI Transactions on Intelligence Technology* 9, no. 3 (July 2024): 722–737, https://doi.org/10.1049/cit2.12263.

25. L. Dai, H. Xu, Y. Zhang, and B. Liao, "Norm-Based Zeroing Neural Dynamics for Time-Variant Non-Linear Equations," *CAAI Transactions on Intelligence Technology* 9, no. 6 (2024): 1561–1571, https://doi.org/10.1049/cit2.12360.

26. C. Qin, L. Chen, Z. Cai, M. Liu, and L. Jin, "Long Short-Term Memory With Activation on Gradient," *Neural Networks* 164, no. 4 (2023): 135–145, https://doi.org/10.1016/j.neunet.2023.04.026.

27. L. Chen, L. Jin, M. Shang, and F.-Y. Wang, "Enhancing Representation Power of Deep Neural Networks With Negligible Parameter Growth for Industrial Applications," *IEEE Transactions on Systems, Man, and Cybernetics: Office Systems* 54, no. 11 (2024): 6837–6848, https://doi.org/10.1109/tsmc.2024.3387408.

28. M. Liu, Y. Li, Y. Chen, Y. Qi, and L. Jin, "A Distributed Competitive and Collaborative Coordination for Multirobot Systems," *IEEE Transactions on Mobile Computing* 23, no. 12 (2024): 11436–11448, https://doi.org/10.1109/tmc.2024.3397242.

29. B. Liao, L. Han, X. Cao, S. Li, and J. Li, "Double Integral-Enhanced Zeroing Neural Network With Linear Noise Rejection for Time-Varying Matrix Inverse," *CAAI Transactions on Intelligence Technology* 9, no. 1 (2024): 197–210, https://doi.org/10.1049/cit2.12161.

30. L. Jin, L. Wei, and S. Li, "Gradient-Based Differential Neural-Solution to Time-Dependent Nonlinear Optimization," *IEEE Transactions on Automatic Control* 68, no. 1 (2023): 620–627, https://doi.org/10.1109/tac.2022.3144135.

31. E. Weinan, "A Proposal on Machine Learning via Dynamical Systems," *Communications in Mathematics and Statistics* 5, no. 1 (March 2017): 1–11, https://doi.org/10.1007/s40304-017-0103-z.

32. H. Huang, L. Jin, and Z. Zeng, "A Momentum Recurrent Neural Network for Sparse Motion Planning of Redundant Manipulators With Majorization-Minimization," *IEEE Transactions on Industrial Electronics* 72, no. 12 (2025): 1–10, https://doi.org/10.1109/tie.2025.3566731.

33. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2016), 770–778.

34. A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features From Tiny Images," *Technical Reports Series* 4 (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

35. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of International Conference on Learning Representations* (2015), https://doi.org/10.48550/arXiv.1409.1556.

36. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2017), 2261–2269.

37. X. Zhang, W. Yu, M. O. Pun, and W. Shi, "Cross-Domain Landslide Mapping From Large-Scale Remote Sensing Images Using Prototype-Guided Domain–Aware Progressive Representation Learning," *ISPRS Journal of Photogrammetry and Remote Sensing* 197 (2023): 1–17, https://doi.org/10.1016/j.isprsjprs.2023.01.018.

38. Z. Li, C. Tang, X. Liu, et al., "Lightweight Remote Sensing Change Detection With Progressive Feature Aggregation and Supervised Attention," *IEEE Transactions on Geoscience and Remote Sensing* 61 (2023): 1–12, https://doi.org/10.1109/tgrs.2023.3241436.

39. H. Chen, Z. Qi, and Z. Shi, "Remote Sensing Image Change Detection With Transformers," *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022): 1–14, https://doi.org/10.1109/tgrs.2021.3095166.