KEAI: An Adaptive Dynamic Programming and Iterative Optimization Model for Event-Triggered Control in Complex Networks

Xiaoru Xing, Yueqiang Hu*, Jianjing Zhang, Dongnuan Zhao, Jing Li
Department of Basic Courses, Hebei Institute of Mechanical and Electrical Technology
Xingtai, 054000, China
E-mail: xingxiaoru586@163.com, 13663199870@163.com, wangyichenzhang31@163.com, zhaodongnuan1122@126.com, hbjd20250226@163.com
*Corresponding author

Keywords: complex network, adaptive dynamic programming, event-triggered control, iterative algorithm, K-means

Received: April 9, 2025

Achieving optimal control of Complex Networks is crucial for optimizing network structures and system solutions. However, current methods mainly suffer from high computational costs and poor performance. Therefore, this study proposes an optimal control model based on Adaptive Dynamic Programming and Iterative Algorithm to address these issues. The model uses Global Dual Heuristic Programming as the foundation, value iterationas the core optimization technique, and integrates event-triggered mechanism and K-means clustering algorithm. The results show that, compared to modelsbased on theHarris Eagle, Arithmetic, and Northern Goshawll optimization algorithms, the proposedmethod reduces physical indicators (e.g., floating-point operations) of the target network by 15%, ensuring accuracy, while achievinga lighter networkand faster convergence. In practical tests, the model is used to optimize the YOLOv8 network. The verification using the CIFAR-10 dataset showed that the YOLOv8's accuracy improved by 10.2%, with response time reducedby 15ms and energy consumption for single-image recognition cut by 31mJ. These results showthat the proposed model effectively achieves optimal control of complex networks, addresses issue like slow speed and high consumption, offers new approaches for optimization, and contribute to more efficientcomplex network development.

Povzetek: Prispevek predlaga model KEAI, ki z združitvijo dinamičnega programiranja, iterativne optimizacije in K-means razvrščanja omogoča učinkovitejše in energetsko manj potratno upravljanje kompleksnih omrežij.

1 Introduction

With the development of deep learning, its application has been steadily growing. Deep learning transforms complex problems across various fields into solvable mathematical problems, using deep neural networks to approximate functions and find optimal solutions [1-2]. Complex networks (CNs) use dynamical methods, graph theory, and other techniques to model complex system relationships as network structures, with nodes, edges, and mathematical relationships through simulation [3]. The significance of CNs lies in finding optimal solutions through mathematical methods and achieving optimal control, which helps solve real-world problems. Thus, achieving optimal control of CNs is crucial [4]. However, current CNs control methods suffer from low efficiency, poor adaptability, and limited scalability, creating an urgent need for more efficient and flexible solutions [5]. Optimal control improves network accuracy and computational speed by adjusting structures and parameters. Adaptive Dynamic Programming (ADP) is widely used for optimal control of systems, as it can optimize network parameters through function approximation using neural networks. ADP often incorporates Iterative Algorithms (IA) to avoid "dimensionality explosion" and improve its applicability. Therefore, the research proposes a CN optimal control framework that integrates ADP and IA, combining the strengths of thesealgorithms to develop a model capable of meeting both accuracy and efficiencyrequirements. The goal is to achieve optimal states in the optimization of CNs, enabling fast and effective implementation of optimization processes across various CNs, and addressing real-world societal challenges.

2 Related work

ADP can integrate reinforcement learning and iterative methods using neural networks to optimize and control various systems. Both domestic and international scholars have applied ADP to system optimization in various complex fields. For instance, Yang et al., in order to address the optimal control problem of systems under the Hamiltonian-driven framework, proposed an ADP method based on experience replay. This method expresses the traditional Hamilton-Jacobi-Bellman equation in a filtered form, allowing system parameter optimization even in weak excitation environments [6]. To solve the voltage control problem in the real-time current sharing process of renewable energy such as wind and solar power, Wang's team proposed an ADP-based voltage regulation method for renewable energy systems. This method treats the voltage regulation problem as an optimal control problem, ensuring the maximum utilization of renewable energy [7]. Selvaraj and Murugasamy, to improve the accuracy of community

Table 1: Past studies and their comparison of advantages and disadvantages.

Field	Method	Advantages	Disadvantages	Authors
Enhancing the efficiency of experience replay in ADP	Integration of Hamiltonian principle and experience replay	High experience replay efficiency	Depends on specific system structures; and sensitive to space distribution	Y. Yang et al. [6]
Sharing and voltage regulation in hybrid wind/solar systems	ADPapproach	High-precision power distribution	Intermittent new energy fluctuations may cause control delays	R. Wang et al. [7]
Node optimization of complex community networks	Integration of InfoMap and Sigmoid Fish Swarm Optimization	High accuracy of results	Fish swarm algorithm require manual tuning and high computational cost	D. Selvaraj et al. [8]
Vibration control for helicopter trailing-edge flaps	Data-driven ADP	High method validity	Relies on specific operational data and high computational cost	Y. Chen et al. [9]
Adaptive fault-tolerant control for non-minimum phase hypersonic vehicles	ADP-based fault-tolerant control	High model stability	High model uncertainty and potential control failure	L. Wang et al. [10]
Optimal control of discrete-time systems in networks	Distributed optimization algorithm	High applicability and repeatability	Relies on real-time node communication; high network loss	S. Battilotti et al. [11]
Safe tracking control with unknown dynamics and constraints	Game theory-integrated optimal control method	Ensures safe tracking under constraints	Validity was not verified in a dynamic environment	X. Cui et al. [12]
Optimal control for continuous-time unknown nonlinear affine systems	Q-learning algorithm	No need to rely on the system, the method is simple	Slow convergence in continuous state spaces; prone to local optima	S. Yu et al. [13]
High-dimensional optimal control for pathfinding	Neural network-driven high-dimensional optimal control method	High path finding accuracy	High computational cost and not suitable for dynamic environments	D. Onken et al. [14]
Influence maximization in complex networks	Evolutionary deep reinforcement learning algorithm	Improves influence propagation efficiency	High computational cost and limited adaptability	L. Ma et al. [15]

monitoring, treatedthe entire community as a CN and each member asa node. Based on the sigmoid fish swarm optimization algorithm, they proposed an improved CN optimal control method for influencing nodes in CN. The test results show that the dropout ratesfor this method on the Facebook, Twitter, and YouTube dataset are 95%, 96% and 94% respectively, effectively achieving optimal control of community monitoring CN [8]. Yu's team applied ADP to control the vibration of a helicopter's rear flap. They proposed a combined framework of ADP and reinforcement learning for rear flap vibration control, and introduced a non-policy reinforcement learning algorithm to determine the optimal control strategy. In practical tests, the method achieved Nash equilibrium and effectively solved the problem of vibration interference in helicopters [9]. Le and Ruiyun, based on ADP’s compensation control, proposed action-dependent heuristic dynamic programming and applied it to the issue of non-minimum phase hypersonic aircrafts affected by actuator faults and parameter uncertainties. In simulation experiments combining a basic fault-tolerant stable controller, the method demonstrated a certain degree of effectiveness [10].

As CNs have developed, research has focused on how to construct networks and optimally control them. Real-world societal problems can only be solved by achieving the optimal state of the abstracted network system. Therefore, domestic and international scholars have conducted extensive and in-depth research on optimal control of CNs. For example, Battilotti, to solve the problem of low computational accuracy caused by CNs nodes only obtaining local network information, proposed an optimal control method based on regulating the consensus step number of network nodes. This method achieves optimal control of the target network by adjusting the length of consensus steps [11]. Cui et al., in order to understand the player relationships in multi-player complex networks, proposed an optimal secure tracking control method. This method combines control barrier functions with non-policy integral reinforcement learning to establish an adaptively updating critic-actor neural network, which can effectively perform optimal control on player relationship CNs [12]. Yu’s team, to achieve optimal control of certain continuous nonlinear CNs without dynamic information, proposed a Q-learning method based on the Hamiltonian function and optimal cost function. This method achieves optimal control of CNs without requiring any dynamic information and improves control efficiency while reducing control costs [13]. The Onken team, to address the dimensionality explosion problem in the optimal control process of CNs, proposed a neural network-based optimal control method. This method integrates the maximum principle with the neural network parameterized value function, where the neural network adaptively provides the optimal strategy function for CNs optimization. Experimental results produced high-dimensional approximate solutions for

CNs optimal control, alleviating the dimensionality explosion problem in the process [14]. Ma et al., combining evolutionary algorithms and deep reinforcement learning, optimized control for CNs. This method optimizes the overall network by activating a small subset of the most influential seed nodes and provides the most optimized solution for network nodes through dynamic Markov

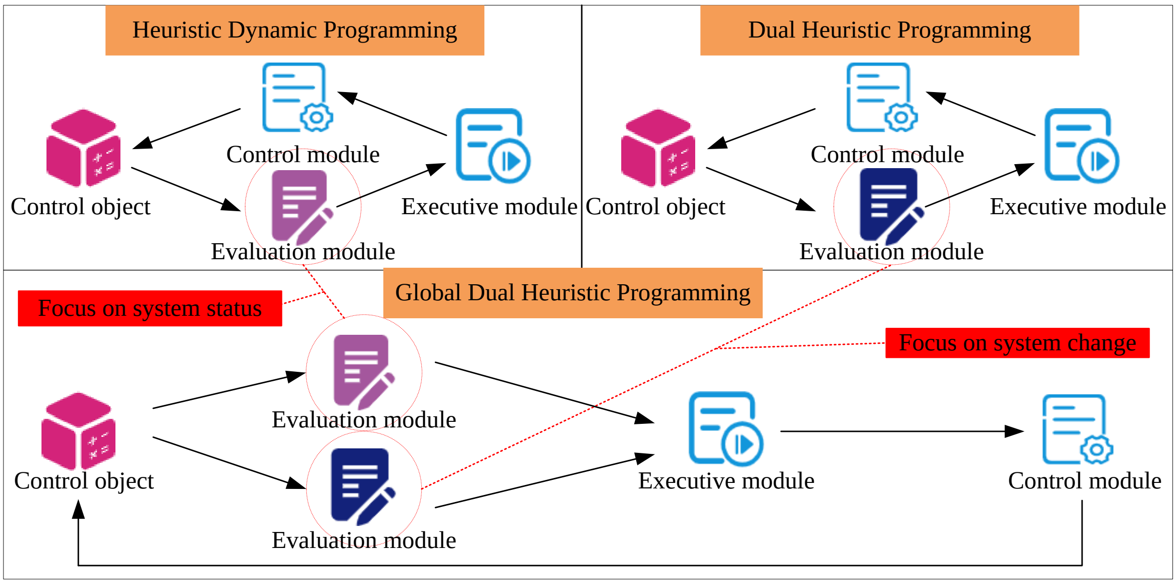


Figure 1: Schematic diagram of the structure of GDHP.

node returns. This method outperforms previous network node optimization methods [15].

In summary, although progress has been made in the research of CNs optimization control methods, existing methods still face problems such as poor control performance and weak adaptability. The ADP and IA-based CNs optimal control algorithm has strong adaptability, and its unique iterative termination mechanism can prevent infinite loops and dimensionality explosion in the control system. Therefore, the research proposes a CN optimal control framework combining ADP and IA, leveragingthe advantages of these algorithms. The optimal control model of CN can thus be obtained, meeting both accuracy and efficiency requirements. It is expected that in the optimization process of CN, it can achieve the optimal control of the target CN by reducing the physical quantities of the target network, optimizing the network nodes and parameters, so that the optimized CN can be more efficient and accurate when applied to practical problems.

3 Construction of optimal control algorithm model based on adp and IA

3.1 Design of optimal control method based on ADP and IA

ADP finds the optimal performance indicators and improvement strategies of the target system through the evaluation module and execution module in the algorithm. In the process, a neural network is used to apply the indicator function and control law, which helps the target system approach the optimal solution [16]. To accommodate different types of system control, researchers have developed various ADP structures based on the evaluation module and execution module. The study selects Global Dual Heuristic Programming (GDHP) as the core structure for complex network optimization. GDHP combines the advantages of heuristic dynamic programming and dual heuristic dynamic programming. Its structure is shown in Figure 1.

In Figure 1, the evaluation module in GDHP evaluates and provides feedback on the operational performance of the control objective through the constructed evaluation network. The control module generates different iteration strategies by adjusting the algorithm or network structure, and the execution module applies the strategies generated by the control module to the control objective. The output of the control module is shown in Equation (1).

$$S(k) = W\lambda(YZ(k)) \tag{1}$$

In Equation (1), W represents the weights between the control module output layer and the hidden layer, Y represents the weights between the hidden layer and the output layer, Z represents the input of the target network, and λ is the activation function. The output of the evaluation module is shown in Equation (2).

$$Q(k) = M\lambda(LN(k)) \tag{2}$$

In Equation (2), M represents the weights between the evaluation module output layer and the hidden layer, L represents the weights between the hidden layer and the output layer, and N represents the input of the target network. The output of the execution module is shown in Equation (3).

$$O(k) = U\lambda(JM(k)) \tag{3}$$

In Equation (3), U represents the weights between the execution module output layer and the hidden layer, J represents the weights between the hidden layer and the output layer, and M represents the input of the target network. The advantage of GDPH lies in combining heuristic dynamic programming with dual heuristic dynamic programming, using a parallel mechanism of dual evaluation modules in itsstructure. The additional evaluation module increases the accuracyof GDHP'sresults but may introduce more calculation

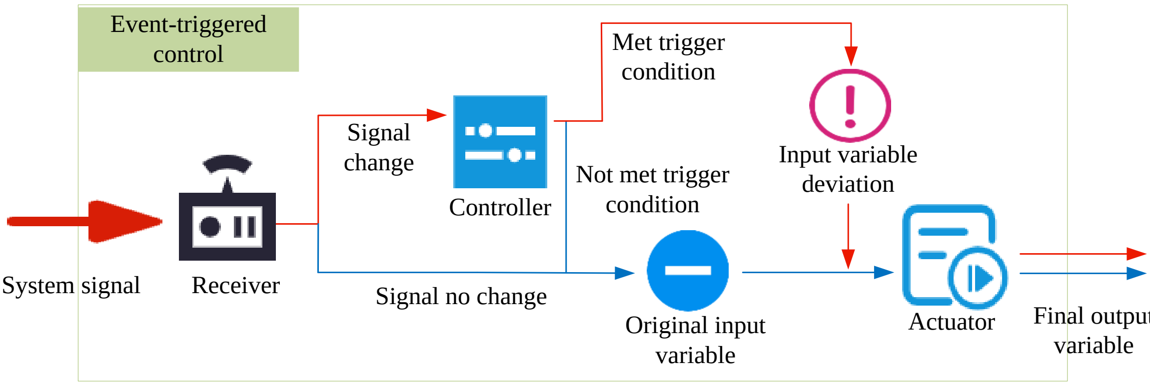


Figure 2: Schematic diagram of ETC working mechanism.

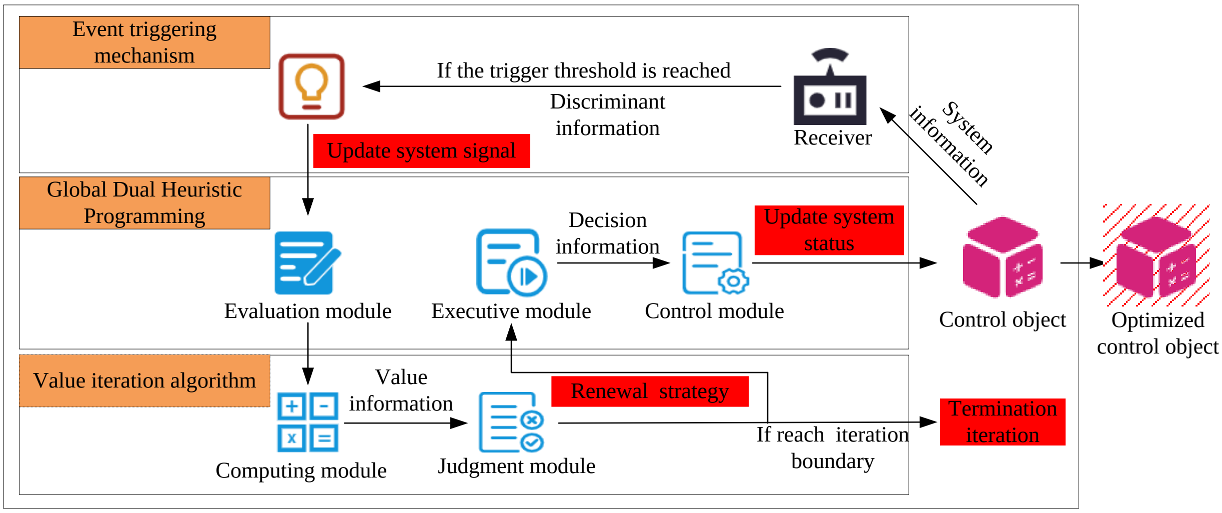


Figure 3: Schematic diagram of EAI workflow.

processes, slowing down the system's speed. The Event-Triggered Control (ETC) mechanism, as a non-periodic control mechanism, can update control signals based on specific conditions to reduce redundant information, optimizing the algorithm's computational speed and memory usage [17]. Therefore, the study integrates ETC before GDHP for data control to avoid unnecessary computational processes in GDHP. The operation mechanism of ETC is shown in Figure 2.

In Figure 2, when the system remains unchanged, ETC outputs a constant control vector based on the previous system state. When the system state changes, a new set of control variables is output. The system state representation method of ETC in this process is shown in Equation (4).

$$x_{m+1} = f(x_m) + g(x_m)b_m + h(x_m)d_m \quad (4)$$

In Equation (4), m represents the time step, x_{m+1} and x_m represent the system states at times $m+1$ and m, respectively. $f(x_m)$, $g(x_m)$, and b_m are all dynamic differentiable functions, feedback functions, and input variables chosenaccording to specific control requirements. $h(x_m)$ represents the output variables, and d_m represents the external disturbance index. When the system state changes, the input variables are updated, as shown in Equation (5).

$$b_m = b(x_{mi}), m \in [m_i, m_{i+1}) \quad (5)$$

In Equation (5), $b(x_{m1})$ represents the input variables at the moment of system change. At this point, a sudden deviation in input variables occurs, as shown in Equation (6).

$$l_m = x_{mi} - x_m, m[m_i, m_{i+1}) \quad (6)$$

In Equation (6), l_m represents the input deviation between time steps m_i and m_{i+1}. Finally, ETC updates the status based on l_m and x_m to obtain the system status at time x_{m+1}, as shown in Equation (7).

$$x_{m+1} = f(x_m) + g(x_m)b(l_m + x_m) + h(x_m)d_m, |l_m| \le l_g \quad (7)$$

In Equation (7), l_g represents the event-triggered threshold. Using GDPH alone for system optimization may lead to issue like dimensional explosion and infiniteloops, while IA can terminate the algorithm by settingamaximum number of iterations or iteration boundaries. The Value iteration algorithm (VIA) is an optimized IA that is suitable for continuous control in large-scale data Spaces. Therefore, the study integrates VIA with GDPH. And construct the ADP-VIA optimal control algorithm integrated into ETC, named EAI, and the process is shown in Figure 3.

In Figure 3, if the system is in a constant state, the ETC mechanism will output a constant control vector.

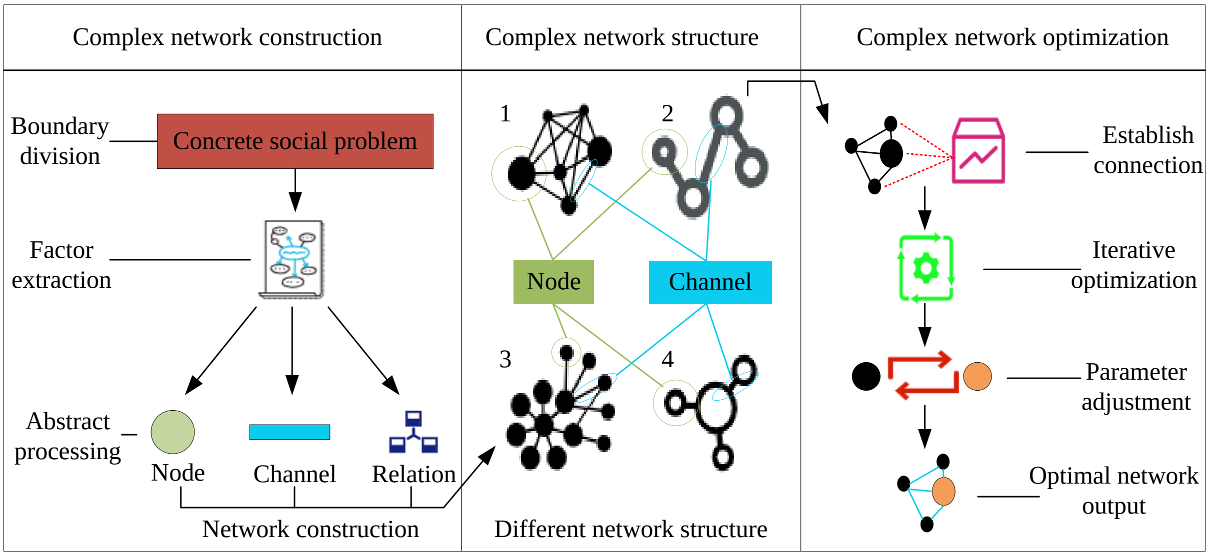


Figure 4: Schematic diagram of the process of CNs construction and optimization control.

When the system state changes, ETC generates a new control vector based on the output deviation caused by the state change of the target system, and the system stateisupdated accordingly. Later, GDHP completes optimal control of the target system through information exchange among the three modules, with policy updates primarily relying on the value iteration algorithm in the evaluation and control module. The theoretical basis of the value iteration algorithm is Bellman's optimal equation. The state value function calculation process is shown in Equation (8).

$$V^{\pi}(s)=\sum_{a\in A}\pi(a|s)\left[r(s,a)+\ell\sum_{s'\in S}p(s'|s,a)V^{\pi}(s')\right] \quad (8)$$

In Equation (8), s represents the current environmental state, a represents the action taken, A represents all actions, $\pi(a|s)$ represents the probability of performing action a among all actions, $r(s,a)$ represents the feedback value after executing action a, ℓ represents the degree of influence of the action on future states, s' is the environmental state after executing the action, and $p(s'|s,a)$ represents the probability of state change after executing the action. After IVA, it is placed in the GDPH structure. Iterative operations are performed on the data output by GDPH, and the optimal solution strategy of the target system is obtained by determining the maximum value of the equation. The calculation method is shown in Equation (9).

$$V_{K+1}(s)=\underset{a\in A}{MAX}\left[r(s,a)+\ell\sum_{s'\in S}p(s'|s,a)V^{\pi}(s')\right] \quad (9)$$

In Equation (9), $V_{K+1}(s)$ represents the optimal solution for all environmental states, In the specific process of CN optimal control, the iteration is usually terminated by setting the maximum number of iterations or boundary conditions to obtain the final optimal solution. The optimization strategy set through this solution can achieve the optimal control of the target CN.

3.2 Construction of CNs optimal control model based on EAI

CNs technology defines the target system as a "small world," isolating it from the overall society. By analyzing the components, information flow, etc., within the "small world," an abstract network is constructed. By optimizing the control of the "small world," optimization of a certain type of problem in society can be achieved [18-19]. Therefore, a specific real-world problem can be abstracted as an independent CNs to solve it. The specific construction and optimization process of CNs is shown in Figure 4.

In Figure 4, when converting a specific societal problem into an abstract CNs, the first step is to determine the specific boundary of the problem and divide it into independent subsystems. Next, a feature analysis of the system is performed, extracting components, relationships between components, etc. Finally, the abstract CNs are constructed based on various elements. Different social subsystems or societal problems often have different boundaries and components. However, a complete CN must include nodes, channels, and interconnections between nodes. These elements, along with physical quantities like the number and size of elements, collectively determine the properties and state of the entire CNs. The degree size represents the density of the network, and the calculation method is shown in Equation (10).

$$M = \frac{1}{m}\sum_{i=1}^{m} M_i = \frac{2n}{m} \quad (10)$$

In Equation (10), n and m represent the total number of nodes and channels in the network, and M_i represents the number of channels around node i. The network size can be represented by the average channel length, and the calculation method is shown in Equation (11).

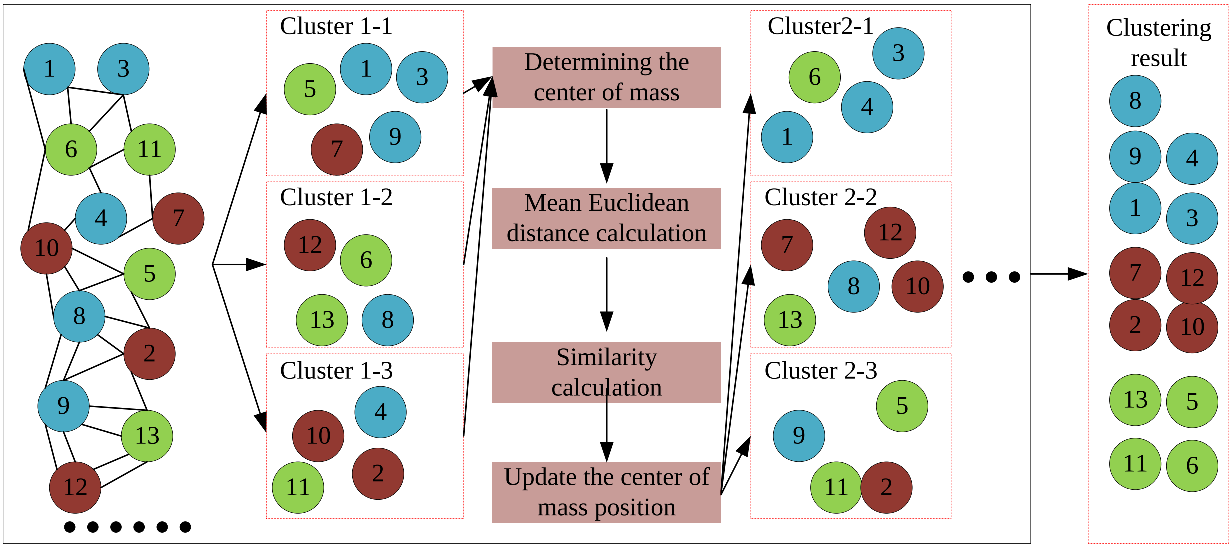


Figure 5: K-means clustering algorithm flow chart.

$$L = \frac{1}{\frac{1}{2}m(m-1)}\sum_{i \neq j} d_{ij} \quad (11)$$

In Equation (11), d_{ij} represents the distance between nodes i and j. The clustering coefficient represents the degree of association between nodes, and the calculation method is shown in Equation (12).

$$C_i = \frac{2E_i}{M_i(M_i - 1)} \quad (12)$$

In Equation (12), E_i represents the number of channels between node i and the current node. The assortativity coefficient represents the similarity between nodes, and the calculation formula is shown in Equation (13).

$$R = \frac{\sum_i (j_i k_i) - \frac{1}{T}(\sum_i \frac{j_i + k_i}{2})^2}{\sum_i (\frac{j_i^2 k_i^2}{2}) - \frac{1}{T}(\sum_i \frac{j_i + k_i}{2})^2} \quad (13)$$

In Equation (13), j and k represent the degrees of two nodes on the same channel, T represents the total number of channels in the entire network. In the CNs optimization control process, the CN must first be connected to the control algorithm. The establishment of data channels is essential for enabling information exchange. Later, the control algorithm adjustsindicators like the number of nodes, density, and assortativity coefficient in the CN through deep learning or function iteration to improve metrics like the network's accuracy and computational speed. However, in practical applications, CNs are often characterized by many nodes and complex components, leading to a heavy computational load during optimization. Clustering algorithms can help classify similar or identical data based on potential data patterns, speeding up the algorithm's understanding of the data distribution of the target system [20]. Among them, the K-means clustering algorithmis simpler, more efficient, and stronger inclassification ability than other clustering algorithms. Therefore, the study uses K-means to optimize the control algorithm. The process of K-means is shown in Figure 5.

In Figure 5, the clustering algorithm's idea is to classify data based on a particular feature. To apply it to the cluster network structure in CNs, the Euclidean distance is chosen as the criterion for data classification, and the calculation process is shown in Equation (14).

$$D(y_i, y_j) = \sqrt{(y_i - y_j)^T (y_i - y_j)}, i, j = 1, 2, 3 \ldots n \quad (14)$$

In Equation (14), n represents the total number of data points in the data, and y_i and y_j represent the i-th and j-th points. After partitioning the data, a criterion function is used to describe the similarity between data clusters. The study uses the sum of distances from all points to the cluster center as the accuracy function. The method of calculation is shown in Equation (15).

$$SSE = \sum_{j=1}^{n}\sum_{i=1}^{n}\left\| y_i - c_j \right\|^2 \quad (15)$$

In Equation (15), c_j represents the center of the current data cluster, and the smaller SSE is, the higher the similarity of the data in that cluster. By classifying data using K-means, the computational load and time for subsequent processes are reduced. Finally, K-means and EAI are applied to the CN optimal control process, resulting in the CNs optimal control model integrating K-means and EAI (KEAI). The structure of the KEAI model is shown in Figure 6.

In Figure 6, KEAI places the K-means algorithm module in front of the original EAI structure. The role of the K-means algorithm is to establish a connection between KEAI and the target CN and classify the similar structures in CN to shorten the connection time. After the connection is established, the ETC mechanism pays attention to the state changes of CN. During this process, the same information is classified for the second time through the K-means mechanism to reduce the frequency of event triggering. In KEAI, the GDHP structure simulates similar functions and iteratively optimizes strategies based on control signals from ETC, and the execution module changes the number of nodes, density, and other parameters in the target CNs to optimize the CNs until the predetermined control goal is achieved or the maximum iteration number is reached, completing the optimal control of the CNs.

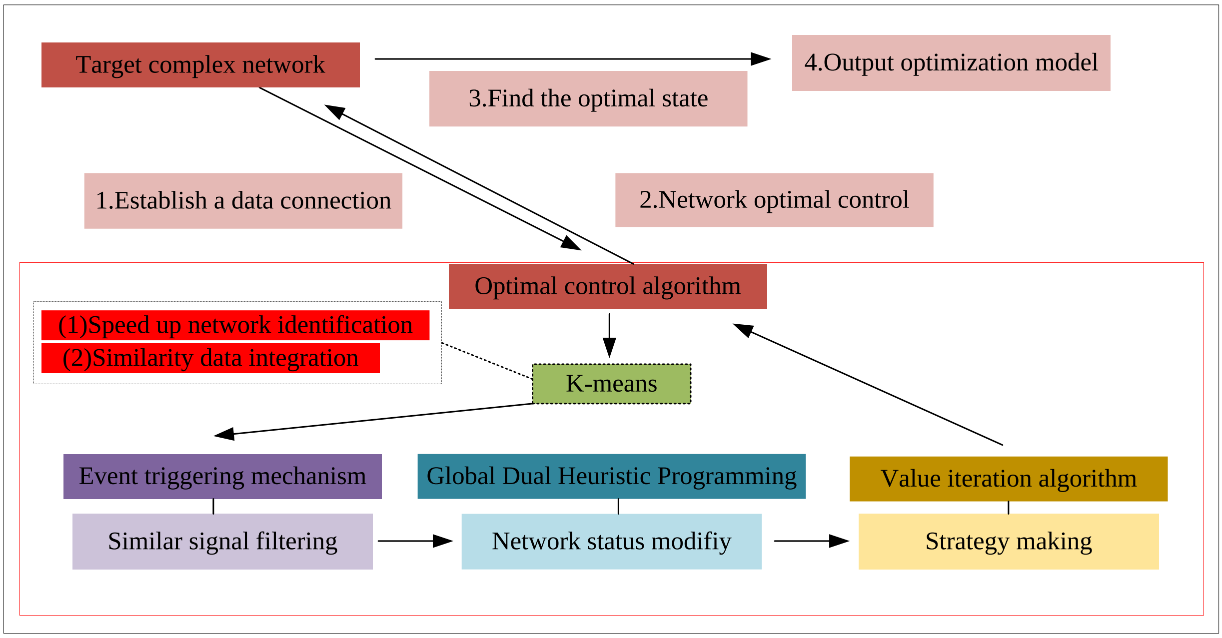


Figure 6: KEAI model structure diagram.

Table 2: Experimental environment and configuration.

Software		Hardware	
Operating system	Windows 10	CPU	Ryzen 7 9800X3D
Programming language	Python 3.8	GPU	GeForce RTX 4060
Deep learning framework	PyTorch 2.1	Memory	32 GB RAM
Data processing	Seaborn	Storage	128 GB SSD

Table 3: The preset parameters of each module in the model.

Parameter	Parameter description	Parameter value
Lr	Learning rate	1e-4
Dropout	Discard rate	0.5
Batch_size	Lot size	24
Hidden_dims_f	The finance module hides the layer dimension	256
Hidden_dims_t	Text modules hide layer dimensions	384
Kernel_size	Convolution kernel size	(2,3,4)
Concat_size	Fusion dimension	192
Max_length	Maximum length of model input	1000*3
Embed_dim	Text embedding dimension	960

4 Verification of CNs optimal control model based on ADP and IA

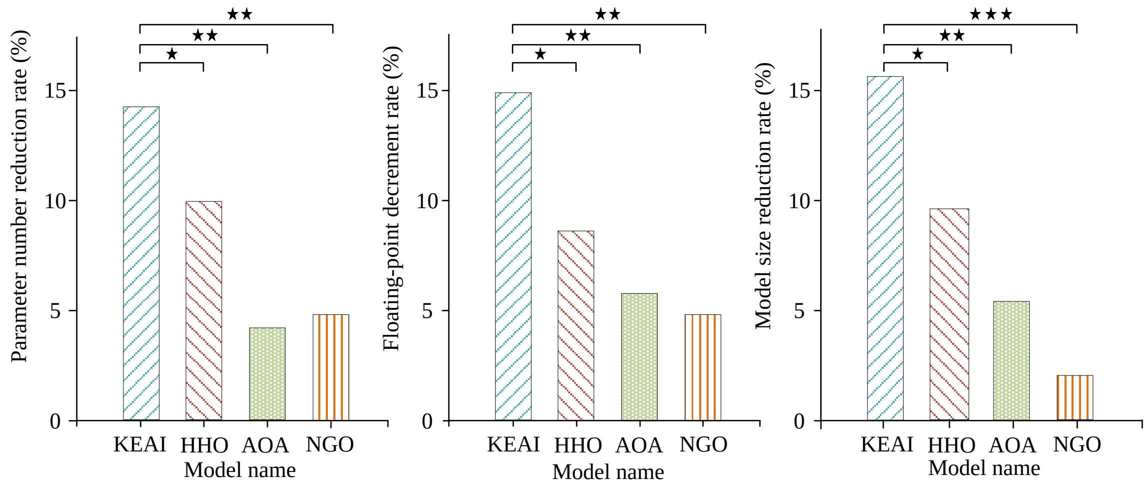
4.1 Verification of optimal control performance of KEAI Model

To evaluate the optimal control performance of the KEAI model for CNs, the study compared it with CN optimal control models based on the Harris Hawks Optimization (HHO), Arithmetic Optimization Algorithm (AOA), and Northern Goshawk Optimization (NGO). During the experiment, four models areused to optimize the complex systems constructed by the same Convolutional Neural Network (CNN) in the same experimental environment.

The parameter Settings of the proposed model and each comparison model during the experiment are shown in Table 3.

In the optimal control of CN, the number of network parameters, floating-point numbers and network size are physical indicators that affect the operation speed and accuracy of the network. Therefore, the number of parameters, floating-point numbers and algorithm size of the optimized network are statistically analyzed, and the results are shown in Figure 7.

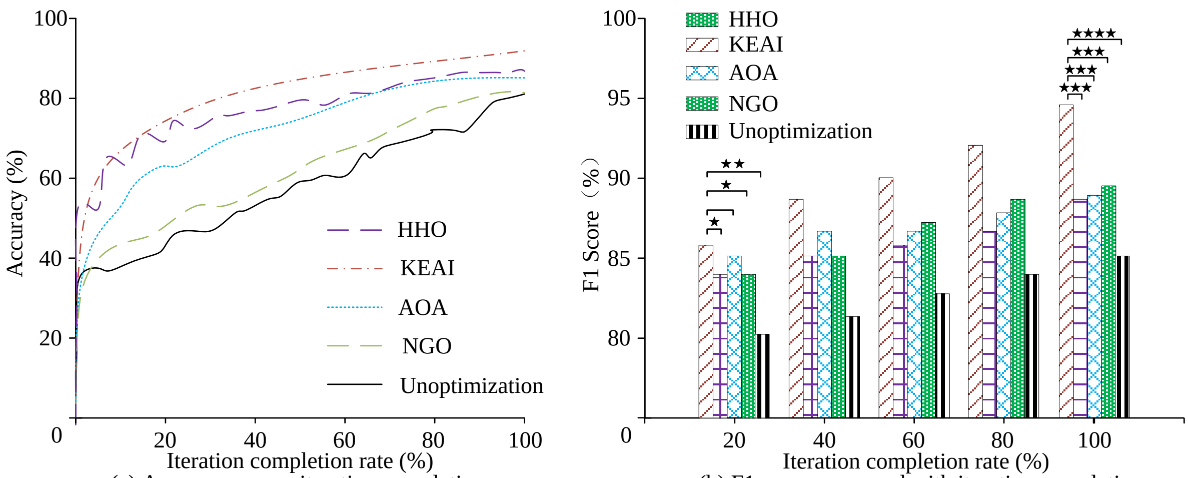
As shown in Figure 7(a), the KEAI model reduced the number of parameters in the target network by 14.2% after achieving optimal control for the CNs, which was the highest value among all the comparison models.



(a) Parameter number variation (b) Floating point variation (c) Model size variation

Figure 7: Statistics of parameter, floating-point number and algorithm variations changes.

Note: ★indicates a slight difference, ★★indicates a significant difference, ★★★ indicates a significant difference.



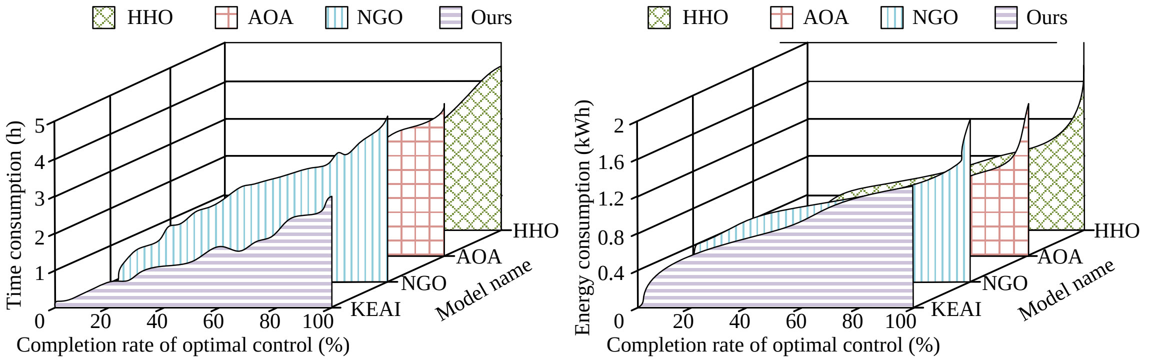
(a) Accuracy versus iteration completion (b) F1 score compared with iteration completion

Figure 8: Comparison of optimal control effects of various optimization methods.

Note: ★indicates a slight difference, ★★indicates a significant difference, ★★★ indicates a significant difference★★★★indicates a significant difference.

From Figure 7(b), it can be seen that the KEAI model reduced the floating-point numbers by 14.9% after achieving optimal control for the target CNs, which was the highest value among all comparison models. Figure 7(c) shows that the KEAI model significantly outperformed the other comparison models after achieving optimal control for the target CN. In summary, compared with the HHO model, AOA model and NGO model involved in the experiment, the KEAI modeldemonstrated stronger optimal control ability, particularly in terms of lightweight aspects such as redundant data cleaning and channel reduction. To further verify the optimal control effect of the KEAI model on the target CNs, the accuracy and F1 score changes of the CNs after the optimal control of each model were compared, as shown in Figure 8.

As shown in Figure 8(a), the accuracy of the unoptimized CN at the end of the iteration was 70.1%, while the accuracy of the CNs after optimal control by the KEAI model was 83.5%. The accuracy of the target CNs after optimal control by the HHO, AOA, and NGO models were 81.2%, 74.3%, and 70.5%, respectively. From Figure 8(b), it can be seen that the F1 score of the CNs optimized by the KEAI model at the end of the iteration was 94.7%, while the F1 scores of the CNs optimized by the HHO, AOA, and NGO models were 88.2%, 88.4%, and 88.9%, respectively. The unoptimized network had an F1 score of 84.5%. In summary, the KEAI model improved the accuracy of the target CNs



(a) Complete the comparison of optimal control consumption time (b) Complete the comparison of optimal control energy consumption

Figure 9: Comparison of time consumption and energy consumption of each model.

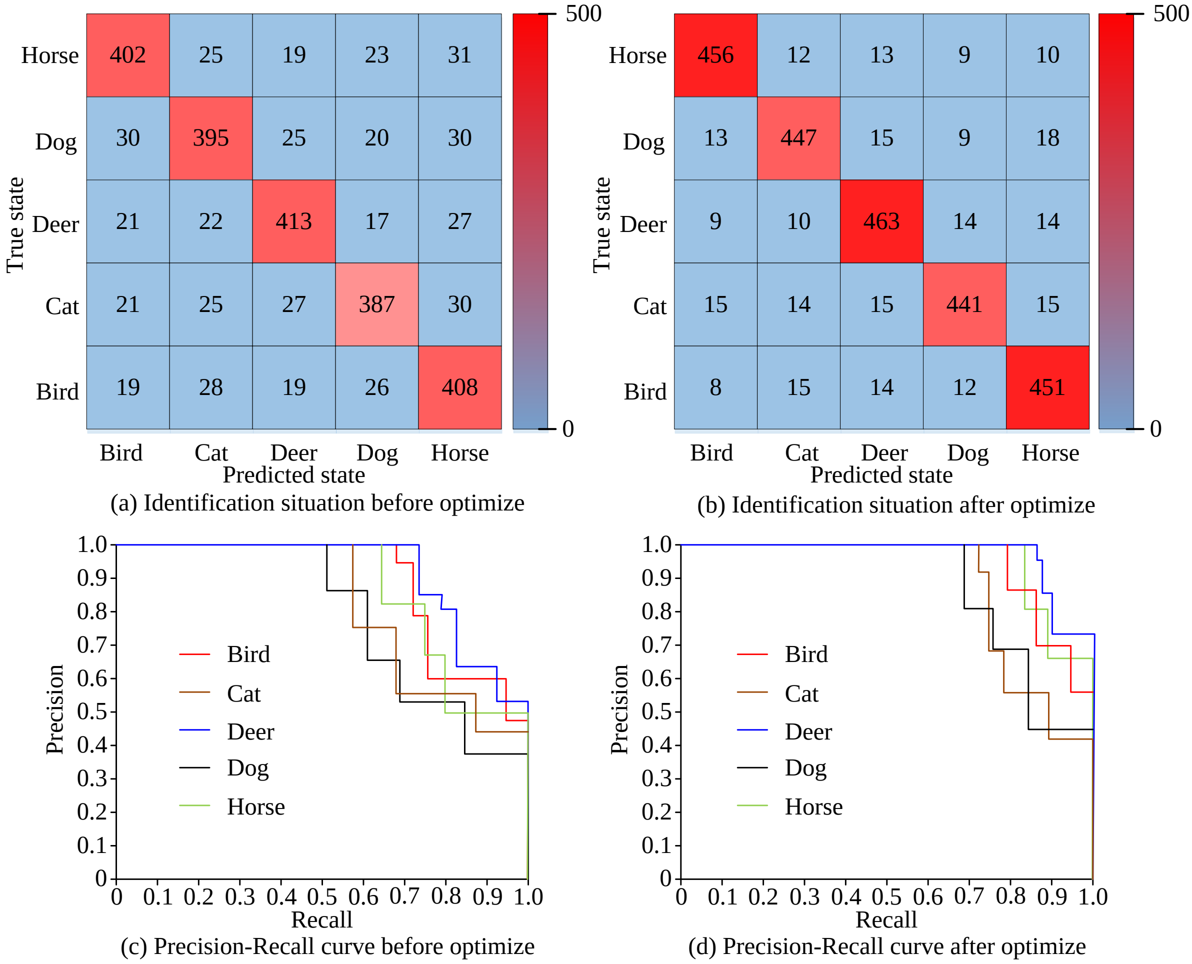


Figure 10: Optimization results of YOLOv8 network's recognition ability.

during the optimal control process. Next, to verify the efficiency of the KEAI model, the time and energy consumption required for optimal control by the four models were analyzed, as shown in Figure 9.

As shown in Figure 9(a), the KEAI model consumed 3.0 hours for optimal control of the target CNs, while the SA model required 4.5 hours, the AOA model required 4.1 hours, and the HHO model required 4.3 hours. This indicates that the proposed KEAI model had a higher optimal control efficiency. From Figure 9(b), it can be seen that the energy consumed by the KEAI model for optimal control of the target CNs was 1.25 kWh, while the AOA model consumed 1.63 kWh, the NGO model consumed 1.60 kWh, and the HHO model consumed 1.59 kWh. Additionally, the energy consumption of the KEAI model remained relatively stable over time, while the energy consumption rate of the other models increased as the control proceeded. This indicated that the KEAI model had lower energy consumption.

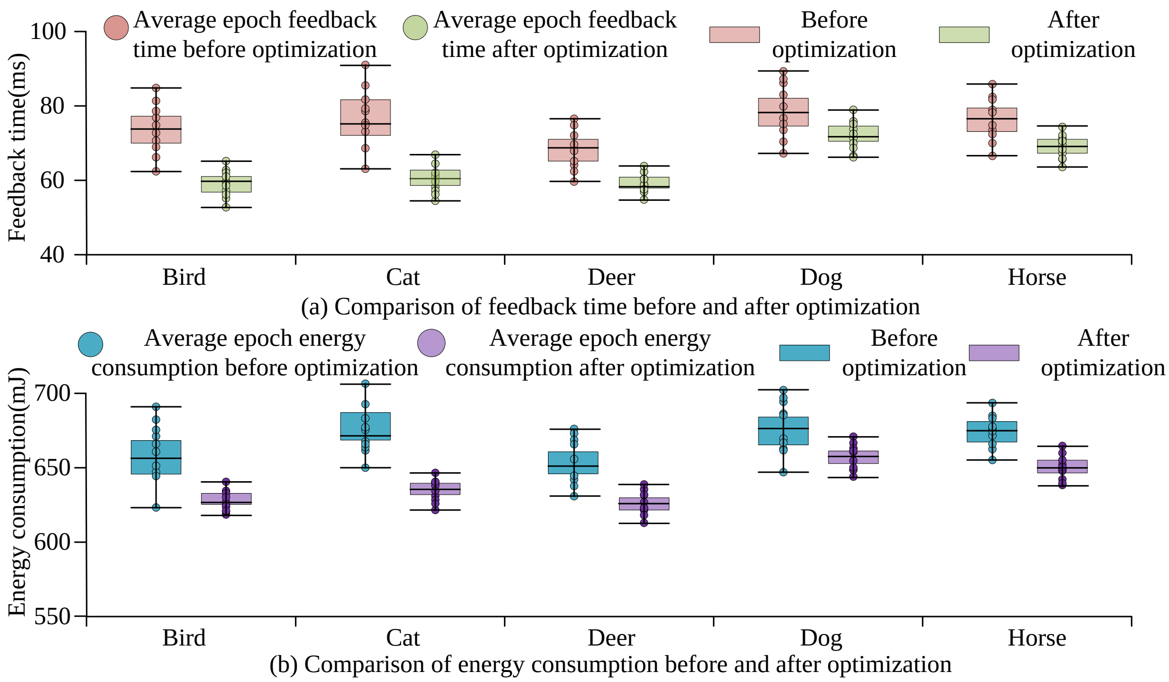


Figure 11: Comparison of average feedback time and average energy consumption.

4.2 Verification of practical effectiveness of KEAI Model in CNs optimal control

After verifying the performance of the KEAI model, the study further evaluated its effect in practical applications by selecting the YOLOv8 complex network, which was constructed for image recognition, as the experimental object. The performance of the YOLOv8 network before and after optimal control by the KEAI model was compared to evaluate its practical application in CNs. The experiment chose the CIFAR-10 dataset to validate the performance of model, CIFAR-10 dataset is commonly used in the field of machine learning, and can be downloaded from http://www.cs.toronto.edu/~kriz/cifar.html. This dataset was collated and released by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton in 2009. It contains a total of 60,000 images of 32*32 pixels in 10 categories, and all the images are RGB three-channel images. When verifying the model performance, 500 pictures of birds, cats, deer, dogs and horses were selected respectively. Every 50 pictures were taken as an epoch to compare and verify the recognition ability of the YOLOv8 network before and after optimization. The results are shown in Figure 10.

As shown in Figure 10(a), the YOLOv8 network was able to identify 2005 images with an accuracy of 80.2% before optimal control. As indicated in Figure 10(b), after optimal control, the YOLOv8 network identified 2258 images with an accuracy of 90.4%, reflectinga 10.2% increase in accuracy. Figures 10(c) and 10(d) show that the AUC values of the YOLOv8 network for bird recognition before optimization were 0.841, for cats 0.732, for deer 0.884, for dogs 0.724, and for horses 0.854. The AUC values of the optimized YOLOv8 network for bird recognition were 0.901, for cats 0.822, for deer 0.942, for dogs 0.814, and for horses 0.925. In summary, the KEAI model effectively improved the recognition ability of the target CNs in practical applications. Next, to verify the optimization ability of the KEAI model in terms of computational time and resource consumption, the average feedback time and average energy consumption per epoch were analyzed, as shown in Figure 11.

As shown in Figure 11(a), through comparison of the average feedback time before and after optimization, it can be seen that the average response time of the network for bird images decreased by 14s, for cat images by 15s, for deer images by 12s, for dog images by 10s, and for horse images by 11s. This indicated that when the KEAI model performs optimal control on complex networks, it effectively reduced the computational time of the target network. In Figure 11(b), the average energy consumption of the YOLOv8 network for bird images decreased by 27mJ, for cat images by 31mJ, for deer images by 25mJ, for dog images by 14mJ, and for horse images by 21mJ. This indicates that the KEAI model effectively reduced the energy consumption of the target network during optimal control. Finally, to verify the general applicability of the model in real-world scenarios, the KEAI model was used for the commonly used neural networks Evolutionary Transformer (Evoformer) in the field of biology and the commonly used Schrodinger neural networks Schrodinger Network (SchNet)) in the field of chemistry Physics-informed Neural Networks (PINNs) commonly used in the field of Physics and Dynamic Causal Neural networks (DCNN) commonly used in the field of economics, Mixture of Experts Neural Network (MoE-NN) commonly used in the field of artificial intelligence Optimization control experiments were conducted using and the Spatio-

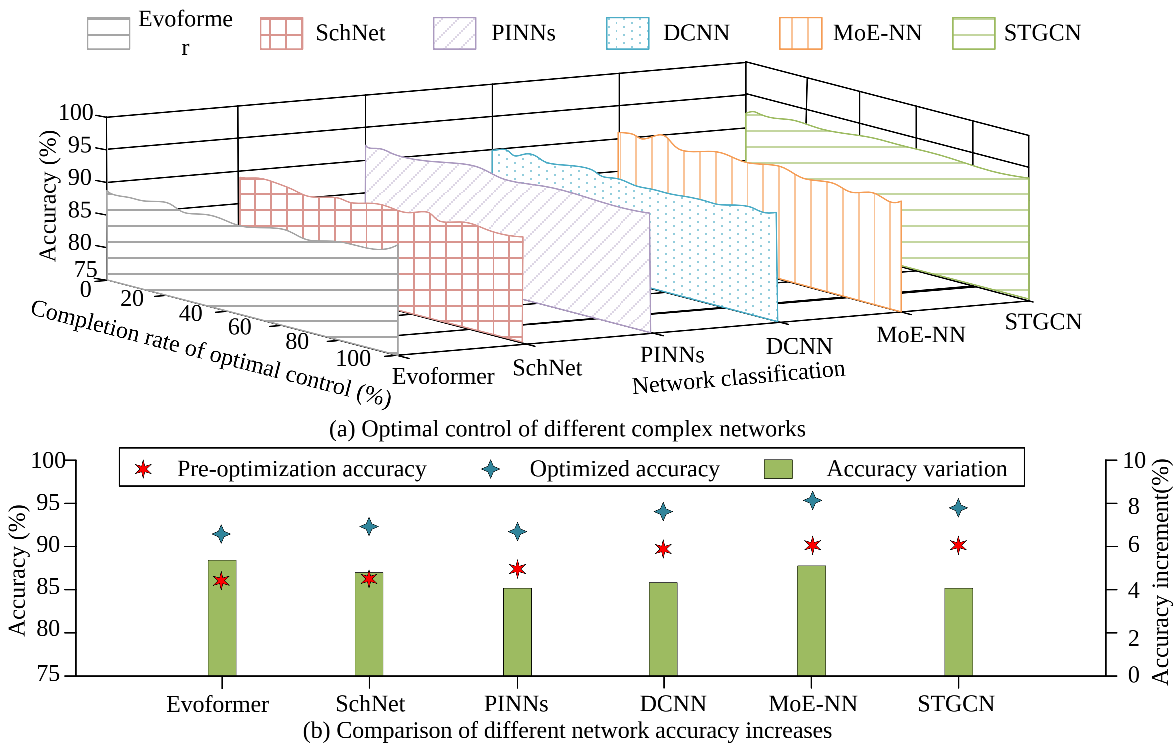


Figure 12: Comparison of average feedback time and average energy consumption.

Temporal Graph Convolutional Network (STGCN) commonly used in the field of transportation. The changes in the accuracy rates of each network are shown in Figure 12.

As shown in Figure 12(a), after optimal control by the KEAI model, the final accuracy of Evoformer was 92.7%, SchNet was 91.8%, PINNs was 89.1%, DCNN was 94.8%, MoE-NN was 94.4%, and STGCN was 93.5%. From Figure 12(b), it can be seen that the accuracy of each model increased after optimal control, with the improvement ranging from 4% to 5%. The average accuracy improvement across all networks was calculated to be 4.7%. This indicates that the KEAI model can be applied to CNs optimal control in different fields and effectively improve the accuracy of the target CNs. Overall, the KEAI model achieved a balance of accuracy, efficiency, and general applicability in the optimal control process of the target CNs, demonstrating strong potential for practical applications. Subsequently, to verify the model performance at different confidence levels, the confidence intervals of the floating-point numbers, parameter quantities, and model sizes of the model for each CN at different confidence levels were compared, as shown in Table 4.

As shown in the table, at different confidence levels, the KEAI model has a good lightweight effect on each target CN, and the floating-point number, parameter quantity and network size of the target CN are all greater than 10%. Moreover, as the confidence level increased, the lightweight effect of the KEAI model on the target network wasalso enhanced, and the confidence interval gradually expands. Among them, when the confidence level was 95%, The floating-point reduction rates of KEAI for Evoformer, SchNet, PINNs, DCNN, MoE-NN and STGCN were 15.1±2.5%, 15.0±1.7%, 13.5±1.9%, 12.5±1.7%, 18.9 ± 2.4% and 13.6 ± 2.2%, respectively. The parameter reduction for Evoformer, SchNet, PINNs, DCNN, MoE-NN and STGCN was 14.2±2.0%, 14.5±1.4%, 13.4±1.9%, 12.0±1.5%, 17.5±2.4% and 12.5±2.0%, respectively. The reduction rates of network size for Evoformer, SchNet, PINNs, DCNN, MoE-NN and STGCN were 13.6±2.4%, 14.1±1.8%, 13.1±1.7%, 12.1±1.7%, 17.5±2.4% and 13.2±1.8% respectively. The above results indicate that the KEAI model has relatively high credibility.

5 Discussion and conclusion

5.1 Discussion

The KEAI model proposed in the research reduced the floating-point operations, parameter count, and network size of the target network by 14.2%, 14.9% and 15.1% respectively in the test experiment. The lightweight effect was significantly better than that of the comparison models, which was related to the advantage of high accuracy of the core GDPH in the KEAI model. Meanwhile, this wasconsistent with the results in the study of Onken et al. using the optimal control to optimize the lightweight index of the target CN [21]. Furthermore, the accuracy rate of CNN optimized by the KEAI model has increased by 13.4%, further demonstrating that the KEAI model has fully exerted the advantages of GDPH and VIA.

This wassimilar to the result that Chen et al. proposed the optimal control

Table 4: Comparison of the lightweighting effects of the KEAI model under different confidence levels.

/	Confidence level(%)	Evoformer	SchNet	PINNs	DCNN	MoE-NN	STGCN
Floating-point reduction(%)	85	11.1±1.4	13.5±1.2	11.2±1.1	10.5±1.0	16.7±1.6	11.3±1.5
	90	13.2±1.7	14.2±1.4	12.2±1.4	11.2±1.2	17.8±1.8	12.5±1.8
	95	15.1±2.5	15.0±1.7	13.5±1.9	12.5±1.7	18.9±2.4	13.6±2.2
Parameter count reduction(%)	85	12.3±1.6	12.4±1.0	11.1±1.5	10.2±0.9	15.9±1.5	11.2±1.4
	90	13.4±1.9	13.1±1.2	12.4±1.8	11.2±1.2	16.5±2.0	11.4±1.8
	95	14.2±2.0	14.5±1.4	13.4±1.9	12.0±1.5	17.5±2.4	12.5±2.0
Network size reduction(%)	85	11.8±1.5	13.0±1.2	11.5±1.0	10.3±1.1	15.9±1.4	11.5±1.2
	90	12.5±1.8	13.5±1.6	12.4±1.4	11.1±1.4	16.4±1.9	12.0±1.6
	95	13.6±2.4	14.1±1.8	13.1±1.7	12.1±1.7	17.5±2.4	13.2±1.8

technology of human resource CN based on BP network and Logistic regression, which optimally increased the recruitment success rate and training effect by 15% [22]. In the performance verification of the model itself, the processing time and energy consumption of KEAI are much lower than those of the comparison model, indicating that the ETC mechanism and K-means algorithm in the model have played a practical role and reduced the total computational load of the model.

In the practical application test, the KEAI model increased the image recognition accuracy of the YOLOv8 network by 10.2%, indicating that the KEAI model can achieve effective optimal control of the target CN, which wasconsistent with the results obtained by Xie et al. in the study of improving the energy utilization rate of the automotive CN through optimal control [23]. In the applicability test, the KEAI model has improved the accuracy of Evoformer, SchNet, PINNs, DCNN, MoE-NN and STGCN from different fields. Thiswas due to the cluster analysis of the K-means algorithm, which enables the KEAI model to effectively achieve data connection and parameter control for different CNS. Meanwhile, the application of CN and its optimal control in route planning and ship navigation in combination with Wan, and the Santander team's view of the enterprise's business strategy as an example of the CN problem can further illustrate that CN was often used to solve various social practical problems [24-25]. The KEAI model proposed in the research has certain promotion space in reality.

All the above results indicate that the KEAI model proposed in the study integrates the advantages of GDPH, the ETC mechanism, VIA, and the K-means algorithm very well, and obtains an optimal control model that is both fast and accurate.

5.2 Conclusion

To enhance the application efficacy of CNs and facilitate more effective solutions to practical social problems, this study proposes the KEAI model, which takes GDPH as the core structural framework and integrates the K-means algorithm, ETC mechanism, and VIA. Experimental results indicate that the KEAI model significantly outperforms comparative models in both lightweight processing of target CNs and accuracy improvement, while consuming less computational time. Additionally, the KEAI model demonstrates strong generalization capability, enabling its adaptation to target CNs across diverse domains. Overall, the KEAI model optimizes various aspects of CN performance by refining parameters, nodes, and other structural components, ultimately achieving optimal solutions for specific problems and showcasing substantial practical relevance. However, this research was only conducted under stable conditions in the laboratory. The optimal control effect of the target CN under the influence of network failures, unstable flows, dynamic environments, etc., was not considered, nor was the performance of the model under different parameter conditions. Therefore, in future research, experiments can be carried out to address this issue, and the performance of the model in all aspects can be continuously improved.

References

[1] Ammar Mohammed, and Rania Kora. A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University-Computer and Information Sciences, 35(2):757-774, 2023. https://doi.org/10.1016/j.jksuci.2023.01.014

[2] Matjaž Gams, and Tine Kolenik. Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 10(4):346, 2021. https://doi.org/10.3390/electronics10040514

[3] Raissa M. D' Souza, Mario di Bernardo, and Yang Yu Liu. Controlling complex networks with complex nodes. Nature Reviews Physics, 5(4):250-262, 2023. https://doi.org/10.1038/s42254-023-00566-3

[4] Guiyang Luo, Quan Yuan, Jinglin Li, Shangguang Wang, and Fangchun Yang. Artificial intelligence powered mobile networks: From cognition to

decision. IEEE Network, 36(3):136-144, 2022. https://doi.org/10.1109/MNET.013.2100087

[5] Jan Carius, René Ranftl, Farbod Farshidian, and Marco Hutter. Constrained stochastic optimal control with learned importance sampling: A path integral approach. The International Journal of Robotics Research, 41(2):189-209, 2022. https://doi.org/10.1177/02783649211047890

[6] Yongliang Yang, Yongping Pan, Cheng Zhong Xu, and Donald C. Wunsch. Hamiltonian-driven adaptive dynamic programming with efficient experience replay. IEEE Transactions on Neural Networks and Learning Systems, 35(3):3278-3290, 2022. https://doi.org/10.1109/TNNLS.2022.3213566

[7] Rui Wang, Dazhong Ma, Ming Jia Li, Qiuye Sun, Huaguang Zhang, and Peng Wang. Accurate current sharing and voltage regulation in hybrid wind/solar systems: An adaptive dynamic programming approach. IEEE Transactions on Consumer Electronics, 68(3):261-272, 2022. https://doi.org/10.1109/TCE.2022.3181105

[8] Devi Selvaraj, and Rajalakshmi Murugasamy. Evaluation of community detection by improving influence nodes in complex networks using InfoMap with sigmoid fish swarm optimization algorithm. Acta Informatica Pragensia, 11(3):380-395, 2022. https://doi.org/10.18267/j.aip.201

[9] Yu Chen, Qun Zong, Xiuyun Zhang, and Jinna Li. Data-driven active vibration control for helicopter with trailing-edge flaps using adaptive dynamic programming. Chinese Journal of Aeronautics, 37(6):151-166, 2024. https://doi.org/10.1016/j.cja.2024.02.019

[10] Le Wang, Ruiyun Qi, and Bin Jiang. Adaptive fault-tolerant control for non-minimum phase hypersonic vehicles based on adaptive dynamic programming. Chinese Journal of Aeronautics, 37(3):290-311, 2024. https://doi.org/10.1016/j.cja.2023.11.006

[11] Stefano Battilotti, Filippo Cacace, and Massimiliano d'Angelo. Distributed optimal control of discrete-time linear systems over networks. IEEE Transactions on Control of Network Systems, 11(2):671-682, 2023. https://doi.org/10.1109/TCNS.2023.3295346

[12] Xiaohong Cui, Wenjie Chen, Binrui Wang, and Kun Zhou. Safe tracking in games: Achieving optimal control with unknown dynamics and constraints. Asian Journal of Control, 26(6):3190-3209, 2024. https://doi.org/10.1002/asjc.3397

[13] Shuhang Yu, Huaguang Zhang, Zhongyang Ming, and Jiayue Sun. Optimal control for continuous-time unknown nonlinear affine systems: A Q-learning approach. IEEE Transactions on Automation Science and Engineering, 21(4):6519-6527, 2023. https://doi.org/10.1109/TASE.2023.3327264

[14] Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach for high-dimensional optimal control applied to multiagent path finding. IEEE Transactions on Control Systems Technology, 31(1):235-251, 2022. https://doi.org/10.1109/TCST.2022.3172872

[15] Lijia Ma, Zengyang Shao, Xiaocong Li, Qiuzhen Lin, Jianqiang Li, and Victor C. M. Leung. Influence maximization in complex networks by using evolutionary deep reinforcement learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(4):995-1009, 2022. https://doi.org/10.1109/TETCI.2021.3136643

[16] Anuradha M. Annaswamy. Adaptive control and intersections with reinforcement learning. Annual Review of Control, Robotics, and Autonomous Systems, 6(1):65-93, 2023. https://doi.org/10.1146/annurev-control-062922-090153

[17] Navjot Singh, Deepesh Data, Jemin George, and Suhas Diggavi. SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization. IEEE Transactions on Automatic Control, 68(2):721-736, 2022. https://doi.org/10.1109/TAC.2022.3145576

[18] Martin Becker, Huda Nassar, Camilo Espinosa, Ina A. Stelzer, Dorien Feyaerts, Eloise Berson, Neda H. Bidoki, Alan L. Chang, Geetha Saarunya, Anthony Culos, Davide De Francesco, Ramin Fallahzadeh, Qun Liu, Yeasul Kim, Ivana Marić, Samson J. Mataraso, Seyedeh Neelufar Payrovnaziri, Thanaphong Phongpreecha, Neal G. Ravindra, Natalie Stanley, Sayane Shome, Yuqi Tan, Melan Thuraiappah, Maria Xenochristou, Lei Xue, Gary Shaw, David Stevenson, Martin S. Angst, Brice Gaudilliere, and Nima Aghaeepour, Large-scale correlation network construction for unraveling the coordination of complex biological systems. Nature Computational Science, 3(4):346-359, 2023. https://doi.org/10.1038/s43588-023-00429-y

[19] Luca Serena, Stefano Ferretti, and Gabriele D'Angelo. Cryptocurrencies activity as a complex network: Analysis of transactions graphs. Peer-to-Peer Networking and Applications, 15(2):839-853, 2022. https://doi.org/10.1007/s12083-021-01220-4

[20] Abdesslem Layeb. ck-means and fck-means: Two deterministic initialization procedures for k-means algorithm using a modified crowding distance. Acta Informatica Pragensia, 12(2):379-399, 2023. https://doi.org/10.18267/j.aip.223

[21] Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach for high-dimensional optimal control applied to multiagent path finding. IEEE Transactions on Control Systems Technology, 31(1):235-251, 2022. https://doi.org/10.1109/TCST.2022.3172872

[22] Feng Chen. Development and evaluation of a machine learning-powered human resource management system utilizing BP neural networks and logistic regression for influencing factor analysis. Informatica, 48(23):133-143, 2024. https://doi.org/10.31449/inf.v48i23.6791

[23] Xuefei Xie, and Lin Chen. Dynamic programming-based energy optimization for pure electric

commercial vehicles with predictive cruise contro. Informatica, 48(22):129-145, 2024. https://doi.org/10.31449/inf.v48i22.6922
[24] Jianxia Wan, and Sukui Zhang. Hybrid deep learning approach for ship navigation in curved river sections using PPO and CNN. Informatica, 48(22):15-29, 2024. https://doi.org/10.31449/inf.v48i22.6909
[25] Omar Santander, Ioannis Giannikopoulos, and Michael Baldea. An integrated stochastic deep learning–short-term production scheduling–optimal control framework for general batch processes. Industrial & Engineering Chemistry Research, 62(5):2124-2137, 2024. https://doi.org/10.23919/ACC55779.2023.10155963