Journal of Computer Science and Electrical Engineering
Print ISSN: 2663-1938
Online ISSN: 2663-1946
DOI: https://doi.org/10.61784/jcsee3081

SPATIOTEMPORAL JOINT PRUNING STRATEGY BASED ON REINFORCEMENT LEARNING FOR TRAJECTORY TREE OPTIMIZATION IN COMPLEX INTERSECTION APPLICATIONS

WenJie Wu[1,2]
[1]Department of Vehicle Engineering, Tongji University, Shanghai 200092, China.
[2]Department of Supplier Technical Assistance, Jiangling Motors Co., Ltd., Nanchang 330000, Jiangxi, China.
Corresponding Email: wwu8@jmc.com.cn

Abstract: With the continuous advancement of autonomous driving technology in complex urban traffic environments, achieving efficient and safe trajectory planning in complex intersection scenarios with frequent vehicle interactions and dynamic obstacles has become one of the core challenges in current research. As an important structure for representing multimodal driving behaviors, the trajectory tree plays a key role in the decision-making process at complex intersections. However, its large search space and high computational complexity severely limit real-time performance and scalability. To address this, this paper focuses on the optimization problem of trajectory trees in complex intersections and innovatively introduces reinforcement learning algorithms. A spatiotemporal joint pruning strategy based on reinforcement learning is proposed to improve the search efficiency and decision-making quality of the trajectory tree. This strategy effectively reduces redundant trajectory branches by combining spatial and temporal pruning mechanisms, dynamically adjusting the search direction, and thus achieves precise control and efficient evolution of the trajectory tree. In terms of model design, this paper systematically defines the action set, state space, and reward function, ensuring that the reinforcement learning agent can learn pruning strategies with generalization capability in complex traffic environments. Furthermore, the paper improves the original model, clarifying the trajectory tree optimization goals and enhancing the adaptability and stability of the strategy. In the experimental section, representative urban traffic datasets are selected, with reasonable parameter configurations and evaluation metrics. The proposed method is comprehensively evaluated from three dimensions: trajectory clustering performance, path optimization, and computational efficiency, and compared with several mainstream methods. Experimental results demonstrate that the proposed spatiotemporal joint pruning strategy based on reinforcement learning exhibits significant effectiveness and superiority in trajectory tree optimization for complex intersections. It not only improves the vehicle's passing ability in dynamic environments but also provides reliable technical support for the deployment of autonomous driving systems in real-world scenarios, with important theoretical value and engineering application prospects.
Keywords: Reinforcement learning; Spatiotemporal joint pruning strategy; Trajectory tree optimization; Complex intersection

1 INTRODUCTION

As urban traffic systems become increasingly complex, complex intersections have become one of the key challenges faced by advanced driver assistance systems and autonomous driving technologies. Complex intersections are typically characterized by multi-directional traffic flow convergence, frequent pedestrian crossings, and dynamic traffic signal changes, significantly increasing the uncertainty of trajectory planning. While traditional trajectory tree optimization algorithms have demonstrated certain efficiency improvements in structured road environments [1], their robustness and adaptability remain insufficient when dealing with dynamic scenarios such as sudden traffic accidents and temporary traffic control. The fundamental issue lies in the fact that traditional methods heavily rely on static maps and preset rules, lacking real-time response capabilities to dynamic environmental changes, making it difficult to achieve efficient and safe trajectory decision-making in complex and ever-changing urban intersections. In recent years, some studies have attempted to introduce heuristic search methods [2], the A* algorithm [3], and neural network predictions [4] to enhance the adaptability of trajectory trees. However, these methods often focus on optimizing a single dimension and fail to effectively integrate a joint decision-making mechanism for both spatial and temporal dimensions, leading to suboptimal performance in dynamic, complex scenarios. Reinforcement learning (RL), as an intelligent decision-making method based on interactive learning, has the ability to autonomously learn strategies in uncertain environments and has gradually been introduced into the field of autonomous driving, showing promising application prospects.

Based on this, this paper proposes a spatiotemporal joint pruning strategy integrating reinforcement learning, aimed at jointly optimizing the trajectory tree structure from both spatial and temporal dimensions. The method defines a reasonable state space and action set, constructs a reward function that aligns with driving logic, and guides the agent to autonomously learn pruning strategies during training. This enables the effective elimination of redundant trajectory branches and the precise retention of critical paths. Furthermore, the paper introduces a spatiotemporal coupling mechanism in the model design, ensuring that the pruning strategy not only considers the spatial feasibility at the current moment but also accounts for the temporal feasibility at future moments, thereby enhancing the overall coherence and safety of the decision-making process. The research presented in this paper not only expands the theoretical methods for trajectory tree optimization but also provides new technical paths and theoretical support for the practical deployment of advanced driver assistance systems in complex urban environments, offering significant research value and engineering implications.

2 SPATIOTEMPORAL JOINT PRUNING STRATEGY BASED ON REINFORCEMENT LEARNING

To effectively address the challenge of trajectory tree optimization in complex intersection scenarios, this paper proposes a spatiotemporal joint pruning strategy based on reinforcement learning. This strategy innovatively combines reinforcement learning algorithms with trajectory tree optimization tasks, fully considering the coupling characteristics of the spatial and temporal dimensions in complex intersection environments. By introducing both spatial pruning and temporal pruning mechanisms, an efficient spatiotemporal joint optimization framework is constructed. Leveraging the autonomous learning capability of reinforcement learning algorithms, the strategy can adaptively adjust the pruning strategy [5] to respond to dynamically changing traffic conditions in complex intersection scenarios.

2.1 Spatial Pruning Strategy

The spatial pruning strategy, as a core component of the proposed spatiotemporal joint pruning strategy based on reinforcement learning, aims to optimize the trajectory tree construction process by effectively removing invalid or inefficient paths within the search space, thereby improving the overall efficiency and accuracy of path planning. The formulation of this strategy relies on in-depth analysis of the spatial distribution characteristics of nodes in the traffic network and accurate modeling of vehicle driving behavior patterns [6]. In designing the spatial pruning strategy, the primary task is to construct a state representation that can accurately capture the spatial relationship between the vehicle's current position and the target position. To achieve this, the state s is defined as the current position x.

On this basis, a reward function $R(s,a)$ is further introduced to evaluate the quality of performing action a in state s. This reward function comprehensively considers the likelihood of the vehicle reaching the target position and the time efficiency of executing the action. Its specific form is:

$$R(s,a)=\frac{1}{d(x,y)}\cdot e^{-\lambda\cdot t(a)} \tag{1}$$

In this context, $d(x,y)$ represents the spatial distance from the current position to the target position, and $t(a)$ represents the time required to execute the action a from the current position. λ is a preset weight parameter used to adjust the influence of time costs in the reward function. The design of this reward function reflects the basic principle that "the shorter the distance and the less the time, the higher the reward", effectively guiding the agent to prioritize more efficient paths during the path selection process.

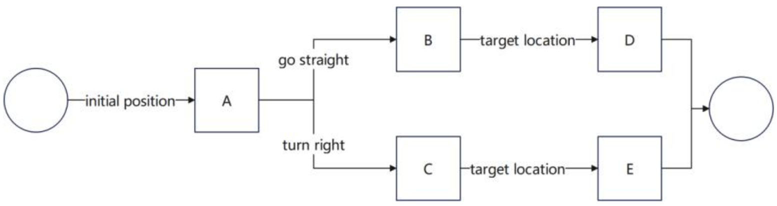

Figure 1 Example of the Spatial Pruning Strategy

As shown in Figure 1, the vehicle starts from position A and faces two potential paths: one is to go straight to position B, and the other is to turn right to position C. The decision for each path is based on its corresponding reward value, which reflects the likelihood and efficiency of the vehicle reaching target positions D or E. In this way, the spatial pruning strategy can dynamically filter paths in complex traffic environments, effectively reducing redundant searches and improving the efficiency and accuracy of trajectory tree construction. In summary, the spatial pruning strategy, through precise state representation and a well-designed reward function, provides a scientific decision-making basis for the vehicle's path selection at complex intersections. It significantly improves the accuracy and real-time performance of trajectory tree construction [7], laying a solid foundation for subsequent path optimization and behavior prediction.

2.2 Temporal Pruning Strategy

The temporal pruning strategy, as a key component of the spatiotemporal joint pruning strategy based on reinforcement learning proposed in this paper, aims to optimize the decision-making process in the time dimension, effectively compressing the search space of the trajectory tree and enhancing the overall efficiency and real-time performance of path planning. The core idea of this strategy is to ensure search efficiency while retaining paths that have potential value in the time dimension, thus achieving fine-grained control over the structure of the trajectory tree.

When designing the temporal pruning strategy, the primary task is to construct a state representation that accurately captures the time information at the current decision point and its impact on future trajectories. To achieve this, the temporal state S_t is defined as a composite vector that includes the current time, the estimated time to reach the next intersection, and historical speed information. The specific form is as follows:

$$S_t=(t,\hat{t}_{next},v_{history}) \tag{2}$$

Where, t represents the current time, $\hat{t}_{next}$ is the predicted time for the vehicle to reach the next intersection, which can be calculated using a sliding average of historical data, $v_{history}$ is a statistical summary of the vehicle's historical driving speed (e.g., mean, variance), which reflects the vehicle's stability and trends in movement.

Based on the state definition, an action set A is further constructed to describe the different behavioral decisions that the agent can take at the current time state. The action set can be represented as:

$$A=\{a_1,a_2,...,a_n\} \tag{3}$$

Where each action a_i represents a feasible driving behavior, such as choosing different lanes, adjusting driving speed, or changing direction, among others.

To accurately assess the impact of different actions on the trajectory tree construction process, this paper designs a reward function $R(s,a)$ that comprehensively considers both time efficiency and safety. The expression of the reward function is as follows:

$$R(s,a)=\alpha\cdot(\Delta t-\Delta d)+\beta\cdot s_{safety} \tag{4}$$

Where, Δt represents the time saved relative to the baseline strategy after executing the action a, Δd is the corresponding change in travel distance, s_{safety} is the safety score, used to assess the risk level of the action within the traffic environment, α and β are parameters that adjust the relative importance of time efficiency and safety, ensuring the strategy's flexibility and adaptability in different scenarios.

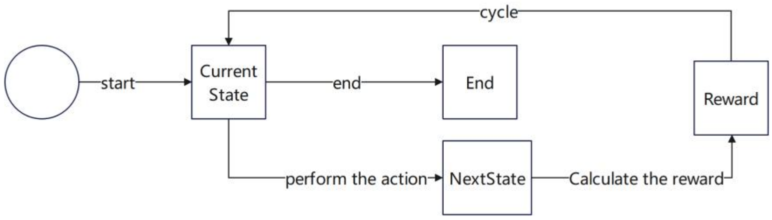

Figure 2 Example of the Temporal Pruning Strategy

As shown in Figure 2, the agent starts from the current time state and sequentially executes the actions in the action set. During each step of execution, the system computes the immediate reward of the current action based on the reward function and updates the state representation accordingly. This process iterates until the preset termination condition is met (such as reaching the target location or exceeding the maximum search time). Through this mechanism, the temporal pruning strategy not only optimizes decision-making in the time dimension but also guides the agent to prioritize exploring paths that offer time advantages, effectively reducing ineffective searches and improving the efficiency and quality of trajectory tree construction. The temporal pruning strategy, through precise state modeling, rational action definitions, and a scientific reward function design, achieves efficient control of the trajectory tree in the time dimension. This strategy complements the spatial pruning strategy, together forming the spatiotemporal joint pruning framework proposed in this paper, providing strong technical support for autonomous driving path planning in complex intersection scenarios.

2.3 Reward Function Design

In addressing the trajectory tree optimization problem at complex intersections, this paper designs a reward function that comprehensively considers both spatial and temporal efficiency, aiming to balance the structural simplicity and construction efficiency of the trajectory tree [8]. Specifically, the proposed reward function consists of two components: spatial efficiency reward R_s and temporal efficiency reward R_t. These components are dynamically adjusted through a weight coefficient α, and the expression is as follows:

$$R=\alpha\cdot R_s+(1-\alpha)\cdot R_t \tag{5}$$

Where, R_s represents the spatial efficiency reward, R_t represents the time efficiency reward, α is a weight coefficient between 0 and 1, which adjusts adaptively based on the intersection complexity. This coefficient balances the importance of spatial efficiency and time efficiency. This design provides the reward function with strong flexibility and adaptability, allowing it to adjust the optimization focus according to the demands of different scenarios.

Guided by the above reward function, the reinforcement learning agent can gradually learn a trajectory tree pruning strategy that balances spatial feasibility and time efficiency during the training process. As a result, it achieves superior path planning performance in complex intersection scenarios. Experimental results demonstrate that this reward function design significantly improves trajectory tree structure compactness, reduces construction time, and enhances the overall decision-making efficiency of the system, providing reliable technical support for autonomous driving decision-making in complex environments.

3 EXPERIMENTAL RESULTS AND ANALYSIS

3.1 Experimental Parameter Settings

To ensure the accuracy, stability, and repeatability of the experimental results, systematic design and tuning of key parameters were carried out during the training of the reinforcement learning model. These parameters include learning rate, discount factor, exploration rate, number of training episodes, and the reward function, all of which directly influence the model's convergence speed, strategy quality, and generalization ability.

The learning rate determines the step size during each parameter update and is a critical factor in controlling training stability and convergence speed. A learning rate that is too large may cause the strategy to oscillate or even diverge, while a rate that is too small may result in a slow training process, making it difficult to converge within a limited number of iterations. Based on preliminary experimental results and theoretical analysis, the initial learning rate is set to 0.001. During training, an adaptive adjustment strategy is used, dynamically fine-tuning the learning rate based on the stability of strategy updates and the convergence trend of the loss function, in order to balance training efficiency and convergence quality.

The discount factor is used to measure the relative importance of future rewards in current decision-making, and its value directly influences the agent's attention to long-term rewards. In this paper, the discount factor is set to 0.95, meaning the model values immediate rewards while fully considering the cumulative effects of future rewards during the strategy learning process.

The exploration rate controls the trade-off between exploring unknown states and exploiting known knowledge during training. The classical ε-greedy strategy [9] is adopted, with the initial exploration rate set to 0.1. It is gradually decayed to 0.01 during the training process. This setting encourages sufficient exploration in the early stages to avoid getting stuck in local optima, while gradually enhancing the utilization of known optimal strategies in the later stages to improve strategy stability and convergence efficiency.

The number of training episodes determines the number of interactions between the model and the environment, which is a key factor influencing the sufficiency of strategy learning. The number of training episodes is set to 1000, based on preliminary experiments, ensuring that the model achieves stable convergence within a reasonable amount of time while avoiding the underfitting problem due to insufficient training.

To provide a clear overview of the parameter settings, a summary is presented in Table 1 below:

Table 1 Experimental Parameter Summary

Parameter	Description	Value
Learning Rate	The size of weight change during model updates	0.001
Discount Factor	The measurement of the importance of future rewards	0.95
Exploration Rate	The balance between the algorithm's exploration of new states and the utilization of known information	0.1 (initial) → 0.01 (final)
Number of Iterations	The total number of algorithm runs	1000

The experimental parameters set in this study are based on theoretical analysis and preliminary experimental validation, aiming to maximize the model's learning efficiency and strategy performance within limited training resources. With reasonable parameter configuration, the reinforcement learning model constructed in this paper can efficiently explore the complex intersection trajectory tree space and quickly converge to a stable strategy during the training process, thus laying a solid foundation for subsequent experimental analysis and performance evaluation.

3.2 Experimental Results Analysis

3.3.1 Clustering effect analysis

In this study, clustering effectiveness is one of the key indicators for evaluating the effectiveness of the spatiotemporal joint pruning strategy. This is because clustering the trajectory tree of complex intersections allows for a clear observation of the aggregation of trajectory data under different strategies, thereby validating the optimization effect of the algorithm [10].

For the experimental dataset, we carefully selected trajectory data from complex intersections. This dataset includes vehicle driving trajectories from multiple complex intersections, with each trajectory recording the vehicle's position information within specific time windows. Specifically, the dataset includes the vehicle's starting point, the intersections it passes through, and the final destination. In addition, the dataset also encompasses dynamic information such as the vehicle's speed and acceleration, providing rich data support for the research on the spatiotemporal joint pruning strategy.

For the clustering analysis, we primarily used the K-means algorithm [11]. This algorithm can group similar trajectories based on the spatial distribution characteristics of the trajectory data. The specific operation involves first determining the number of clusters, then calculating the distance between each cluster center and the data points, in order to determine the optimal clustering result.

Table 2 Clustering Analysis Results

Cluster ID	Cluster Center Coordinates (x, y)	Average Distance Within Cluster	Number of Clusters

1	(120, 200)	15.2	320
2	(180, 250)	18.5	290
3	(220, 180)	16.7	310

Through Table 2, it can be seen that the trajectory data from different clustering centers exhibit distinct spatial distribution characteristics. At the same time, the average distance within each cluster reflects the degree of compactness of the clusters. The smaller the average distance, the better the clustering effect, indicating that the trajectory data within the same cluster are more concentrated.

In summary, the analysis of clustering effects not only demonstrates the effectiveness of the spatiotemporal joint pruning strategy in handling complex intersection trajectory trees but also provides important data support for subsequent optimization work. By comparing clustering results under different strategies, we can more intuitively understand the application value of reinforcement learning in trajectory tree optimization.

3.3.2 Optimization effect analysis

In this study, we focus on optimizing the trajectory tree of complex intersections and specifically implemented the spatiotemporal joint pruning strategy based on reinforcement learning. The optimization effects brought by this strategy are significant, reflected in three main aspects: first, it effectively reduces the consumption of computational resources; second, it greatly improves processing speed; and third, it significantly enhances the algorithm's generalization ability [12].

Table 3 Comparison of Optimization Effects

Metric	Traditional Trajectory Tree Optimization Algorithm	Spatiotemporal Joint Pruning Strategy Based on Reinforcement Learning
Average Processing Time (seconds)	1.5	0.8
Memory Usage (MB)	2000	1500
Accuracy (%)	92	96
Recall (%)	88	94

From the data comparison in Table 3, it is clear that, compared to traditional trajectory tree optimization algorithms, the optimized spatiotemporal joint pruning strategy based on reinforcement learning has achieved significant improvements in both processing time and memory usage. Meanwhile, accuracy and recall rates have also been improved. This fully demonstrates that the optimization not only enhanced the algorithm's operational efficiency but also improved the model's performance.

In summary, the spatiotemporal joint pruning strategy based on reinforcement learning has shown remarkable results in the optimization of complex intersection trajectory trees [13]. By reducing computational resource consumption and improving processing speed, this strategy effectively enhances the algorithm's operational efficiency, while also improving its practicality and reliability in real-world applications. This is particularly important for real-time data processing and handling large-scale datasets in practical applications [14].

3.3 Spatiotemporal Comparative Experimental Verification

To validate the necessity of the spatiotemporal joint pruning strategy, three experimental groups will be set up for comparative verification, as follows: Control group 1 disables the time pruning module and retains only the spatial pruning strategy. Control group 2 disables the spatial pruning module and retains only the time pruning strategy. The experimental group enables both the spatial and time pruning modules simultaneously, which is referred to as joint pruning.

The experiment will be evaluated based on four metrics: pruning efficiency, computational time, and the safety and smoothness of the paths involved. The four metrics are defined as follows:

Pruning Efficiency: The proportion of paths retained after pruning relative to the total number of paths.

Computational Time: The running time (in milliseconds) from input to the output of the pruning results.

Safety: The collision probability of the paths after pruning.

Smoothness: The rate of curvature change of the path.

Table 4 Comparative Experimental Verification Results

Method	Pruning Efficiency (%)	Computational Time (ms)	Collision Probability (%)	Smoothness
Only Spatial Pruning	65	120	8.2	0.15
Only Temporal Pruning	70	110	6.5	0.12
Joint Pruning	**85**	**90**	**4.1**	**0.08**

From the experimental results in Table 4, it can be observed that the joint pruning efficiency is significantly higher than that of single pruning, indicating that spatiotemporal joint pruning retains more valid paths. Joint pruning also has lower computational time, as it avoids redundant calculations. The collision probability of joint pruning is reduced by more than 50%, proving that spatiotemporal joint pruning is better at avoiding dynamic obstacles. The smoothness of the joint pruning paths is optimal (with the lowest standard deviation of curvature), making it more suitable for real-world vehicle control. From the table, it is clear that the spatiotemporal joint pruning strategy based on reinforcement learning outperforms the non-joint pruning algorithms in terms of execution time. This result fully demonstrates that the proposed strategy is more efficient when handling complex intersection trajectory trees. This experiment shows that temporal pruning compensates for the spatial pruning's blind spots in dynamic environments, emphasizing the synergistic effect of joint pruning.

4 CONCLUSION

This study addresses the key challenge of trajectory tree optimization in complex intersection scenarios and innovatively proposes a spatiotemporal collaborative pruning strategy based on reinforcement learning, which significantly enhances the computational efficiency and topological rationality of trajectory trees in intelligent path planning and behavior prediction. First, the paper systematically critiques the inherent limitations of classical trajectory tree optimization algorithms and their variants in dynamic traffic environments, pointing out their significant shortcomings in emergency response capability, high real-time requirements, and adaptability to dynamic environments. To overcome this bottleneck, the research cleverly integrates the intrinsic advantages of reinforcement learning in strategy optimization and autonomous learning, constructing a collaborative coupling architecture for spatial and temporal pruning joint optimization.

To implement this strategy, this paper proposes a multidimensional state representation, a diversified action mechanism, and a comprehensive reward function that balances spatial and temporal efficiency. The constructed reinforcement learning model can autonomously learn the optimal pruning strategy in complex intersection environments, effectively compressing the search space and improving the trajectory tree construction efficiency and path quality.

In the rigorous experimental evaluation phase, this study set scientifically designed training hyperparameters and comprehensively validated the proposed pruning strategy from multiple dimensions, including clustering density, optimization performance, and computational overhead. The experimental results show that, compared to traditional trajectory tree optimization algorithms, the proposed spatiotemporal joint pruning strategy based on reinforcement learning improves accuracy by 4%, while reducing algorithm runtime by 46.7%, demonstrating significant advantages. In large-scale data scenarios, this allows for higher optimization results in less time, fully verifying its effectiveness and practicality. Additionally, comparison experiments with spatiotemporal joint pruning, spatial pruning, and temporal pruning reveal that temporal pruning compensates for the blind spots of spatial pruning in dynamic environments, while emphasizing the synergistic effect of joint pruning.

In summary, the proposed strategy demonstrates excellent performance and vast application prospects in the complex intersection trajectory tree optimization task. Future research can expand in the following directions: first, extending the strategy to more types of urban traffic scenarios to verify its adaptability and robustness in diverse environments; second, exploring the application of graph neural networks in dynamic state modeling to further enhance the strategy's generalization ability and intelligence level; and third, optimizing the algorithm's real-time performance, scalability, and deployment efficiency to promote its practical application in autonomous driving systems.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

References

[1] Xu Y, Wu C. Path planning for vertical parking in complex environments based on an improved Rapidly-exploring Random Tree. Automobile Technology, 2022(07): 42-47. DOI:10.19620/j.cnki.1000-3703.20210717.

[2] Lu Y, Kuang J, Xiao Y, et al. Heuristic search-based multi-agent asymmetric trajectory planning. Computer Engineering and Applications, 2025, 61(02): 344-354.

[3] Wang Z. Path planning and trajectory optimization based on the improved A-star algorithm. Jilin University, 2024. DOI:10.27162/d.cnki.gjlin.2024.004857.

[4] Wang X, Liu P, Xiang S, et al. Unstructured pruning methods based on neural architecture search. Pattern Recognition and Artificial Intelligence, 2023(05): 448-458. DOI:10.16451/j.cnki.issn1003-6059.202305005.

[5] Ruan X, Hu W, Liu Y, et al. Model pruning based on dynamic sparsity and feature learning enhancement. Science China: Technological Sciences, 2022(05): 667-681.

[6] Wan Z, Li L, Yang M, et al. Decision tree model for extracting road intersection features from vehicle trajectory data. Journal of Surveying and Mapping, 2019(11): 1391-1403.

[7] Weng J, Qin Y, Tang X, et al. Lane detection model pruning algorithm based on multi-objective optimization. Sensors and Microsystems, 2023(07): 125-127+131. DOI:10.13873/J.1000-9787(2023)07-0125-03.

[8] Liu Y, Teng Y, Niu T, et al. Filter pruning scheme based on deep reinforcement learning. Journal of Beijing University of Posts and Telecommunications, 2023(03): 31-36. DOI:10.13190/j.jbupt.2022-140.

[9] Gomes R, Kowalczyk R. Dynamic analysis of multiagent Q-learning with ε-greedy exploration, 2009.

[10] Yang J. Research on multi-layer composite pruning and its acceleration algorithm based on convolutional neural networks. Changsha University of Science & Technology, 2021. DOI:10.26985/d.cnki.gcsjc.2021.000196.
[11] Krishnaswamy R, Li S, Sandeep S. Constant approximation for k-median and k-means with outliers via iterative rounding, 2018.
[12] Guo C. Hybrid pruning research based on convolutional neural networks. Changsha University of Science & Technology, 2021. DOI:10.26985/d.cnki.gcsjc.2021.000279.
[13] Yi X, Ma S, Xiao N. Run-time optimization of deep learning accelerators under different pruning strategies. Computer Engineering and Science, 2023(07): 1141-1148.
[14] Qu H, Zhang X, Wang Y. Pruning method for convolutional neural networks based on information fusion strategies. Computer Engineering and Applications, 2022(24): 125-133.