Deep Reinforcement Learning For Adaptive Multi Axis CNC Tool Path Optimization Under Dynamic Constraints

*Xiaorong Zhou, Lidong Huang, and Xiaoping Liu

School of Mechanical Engineering, Hunan Mechanical & Electrical Polytechnic, Changsha, Hunan, 410151, China

*Corresponding author. E-mail: xiaorong$_z$hou34@outlook.com

Received June 5, 2026; accepted July 18, 2026

In modern production, multi-axis Computer Numerical Control (CNC) machines are utilized to cut complex pieces with extreme precision. However, during the cutting process, real-world problems such as tool wear, vibration, heat, and material changes make it difficult to adhere to predetermined tool paths. Traditional methods cannot react in real time, resulting in low efficiency and increased expenses. The goal of this research is to develop an intelligent and adaptive system for CNC tool path planning that uses deep reinforcement learning (DRL) to respond to dynamic machining environments and increase cutting performance. The dataset includes 3D CAD models, sensor data, tool positions, feed rates, and material parameters, which are utilized to train and validate the DRL agent. The proposed method utilizes fixed grid voxelization (FGV) to convert 3D CAD part representations into voxel grids, enabling the algorithm to analyze the machining region as discrete geometry blocks. Next, environment modeling is structured using a Markov Decision Process (MDP), considering tool conditions like location, spindle load, and vibration as states and tool movements as actions. Dijkstra's algorithm (DA) is used to find the shortest baseline path across the voxel grid, which improves the DRL agent's decision-making during toolpath generation. The learning agent uses Proximal Policy Optimizer Driven Double Deep Q-Network (PPO-DDQNet) for optimal tool pathways, ensuring precise value prediction and stabilizing the learning process. Geometry Planning under Varying Time Constraints based on angular loss, vibration loss 1450 -2000 seconds.

Keywords: CNC environments, tool path planning, Proximal Policy Optimizer Driven Double Deep Q-Network (PPO-DDQNet), cutting performance, Computer Numerical Control (CNC)

http://dx.doi.org/10.6180/jase.202611_34.015

1. Introduction

The CNC machines have become essential manufacturing resources to complete accurate material shaping in a diverse range of modern industries [1]. CNC machines with axes have the ability to create complex shapes and cut the number of setups involved and the market impact of surface appearances through a minimization of reorientation [2]. The determinism of traditional tool path planning algorithms leaves it unable to adapt to real-time machining, thus leading to poor machining performance, higher tool prices, and compromised product quality. The complications were exacerbated by the consideration of real-time dynamic constraints like tool wear, material inconsistencies, temperature [3] changes, and machine vibrations. An artificial intelligence (AI) model has led the production line to smart decision-making and independent optimization. It has attracted a lot of attention as a tremendously practical solution to the issues of sequential decision-making, and it represents better potential to drive adaptive toolpath planning through CNC dynamic environments [4]. The DRL model with the trial-error training enables the systems to acquire complex policies through learning a high-dimensional state, such as sensor input space, machine configuration, and material conditions [5].

DRL continuously optimizes CNC tool paths using real-time feedback on cutting speed, feed rate, and tool orientation to improve machining performance [6]. Edge computing, IIoT, and high-resolution sensors enable real-time monitoring and adaptive decision-making in CNC environments [7]. DRL supports predictive maintenance by reducing tool wear and enhancing machining efficiency [8]. It also improves energy efficiency, surface quality, and collision avoidance [9]. However, DRL requires extensive training data and computational resources, limiting real-time deployment across diverse machining environments. This study introduces an AI-driven framework for adaptive CNC tool path optimization to support smart manufacturing. Traditional CNC toolpath optimization relies on predefined offline paths and cannot adapt to dynamic machining conditions. Existing RL methods improve adaptability but suffer from unstable learning, inaccurate value estimation, and limited generalization. This study proposes a hybrid PPO-DDQNet framework integrating FGV and Dijkstra' s algorithm for adaptive multi-axis CNC toolpath optimization.

This study develops PPO-DDQNet for adaptive CNC tool path optimization, improving value prediction, learning stability, cutting performance, tool life, and surface quality

- 3D CAD models, real-time sensor data, tool positions, feed rate, and material properties were used to train and validate the adaptive machining system.
- FGV converted CAD models into voxel grids, while MDP modeled the machining environment and tool dynamics.
- PPO-DDQNet optimized tool movements using real-time parameters, with DA improving DRL decision-making for efficient machining.
- The proposed method reduced tool wear, machining time, and improved surface quality.

The remaining research consists of a literature review on adaptive CNC machining, methodology (FGV, MDP, and PPO-DDQNet model development), and experimental results highlighting superior metrics. The conclusion emphasizes the model's adaptability and the real-time applicability of the PPO-DDQNet-based intelligent tool path planning system.

2. Materials and methods

Recent studies explore DRL-based adaptive CNC tool path optimization, integrating multi-source machining data and MDP-based dynamic modeling to improve tool path precision, reduce energy consumption and tool wear, and enhance machining efficiency.

MSAC-based MDP optimization reduced CNC energy consumption by 69.96% [10], while NAR-LSTM and TS-DQN improved feed-axis error prediction and compensation accuracy [11]. Related CNC tool path optimization studies are summarized in Table 1.

2.1. Problem Statement

Current DRL methods improve CNC machining but still face challenges in real-time adaptability, material inconsistencies, and generalization. NAR-LSTM requires large high-quality time-series datasets and is difficult to integrate into CNC systems [11]. PPO-DDQNet addresses limited generalization through adaptive tool path optimization using real-time feedback [13].

2.2. System Architecture

The integrated FGV-MDP-DA-PPO-DDQNet framework optimizes multi-axis CNC tool paths, enhancing adaptive decision-making, stability, and machining precision in dynamic environments as illustrated in Figure 1.

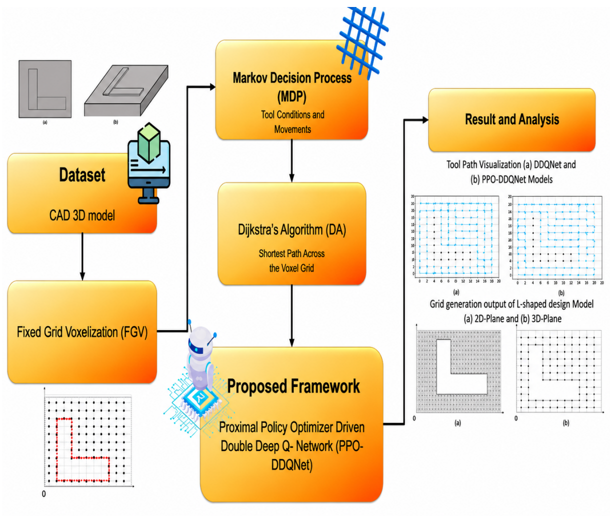

Fig. 1. Overall Flow of Multi Axis CNC Tool Path Optimization

The proposed framework follows a sequential workflow. FGV converts the 3D CAD model into a voxel representation, which is modeled as an MDP for reinforcement learning. Dijkstra' s algorithm generates an initial shortest path,

Table 1. Summary of Literature Review on CNC Toolpath Optimization and Planning

References	Technology Used	Objective	Result	Challenges
[12]	Intelligent agents, deep learning (DL) model	To recognize CNC toolpath geometric features and optimize machining trajectory	Improved smoothness and reduced machining defects in CNC paths	Handling path discontinuity and curvature defects
[13]	Reinforcement Learning (RL), Grid conversion,	To optimize toolpath generation for complex grinding	Demonstrated faster and higher-quality paths than evolutionary techniques	Lack of general path optimization methods for complex features
[14]	DRL model	Generalizable tool path prediction for arbitrary manufacturing	Demonstrated better generalization than pure RL methods	Prior RL methods lacked flexibility for unseen geometries
[15]	Deep Convolutional Neural Network (DCNN), Domain knowledge Integration	Intelligent CAM for optimizing effective toolpaths	Increased machining efficiency by up to 40.7%	Traditional tool paths lacked optimization and adaptability

and the PPO-DDQNet agent continuously refines this path using machining feedback to achieve adaptive toolpath optimization under dynamic conditions.

2.3. Fixed Grid Voxelization (FGV)

The complex 3D geometry workpiece was discretized by FGV for regular voxel grids, which simplifies working with space representation and collision detection in the course of machining. FGV was selected over alternative discretization strategies (e.g., octrees, adaptive grids, surface meshes) due to its uniform state-space representation, computational simplicity, and efficient collision detection. Unlike adaptive methods, FGV maintains consistent voxel resolution, simplifying DRL policy learning, while also capturing the full machining volume essential for subtractive manufacturing. FGV converts 3D CAD models into structured voxel grids, enabling DRL agents to learn efficient tool paths through spatial interaction. The grid is discretized into voxels based on dimensions and coordinate positions, where each point is mapped to its corresponding voxel location.

2.4. Markov Decision Process (MDP)

The MDP models CNC machining as a sequential decision process using states, actions, rewards, transition probabilities, and discount factor γ. The MDP framework is defined by the tuple (S, A, R, P, γ), as summarized in the table 2.

The MDP models agent-environment interactions using state, action, reward, and transition probability. The discount parameter γ balances immediate and future rewards, as expressed in Eq. (1).

$$Q_s = \sum_{l=s}^{\infty} \gamma^{l-s} q_l = q_s + \gamma Q_{s+1} \tag{1}$$

In Eq. (1), r_t denotes the immediate reward obtained after executing action a_t in state s_t, γ represents the discount factor, and R_t denotes the cumulative discounted reward. Throughout this manuscript, the notation s_s, a, r and γ is used consistently to represent state, action, reward, and discount factor, respectively.

2.5. Transition Probability

The discount factor γ balances short- and long-term CNC performance by prioritizing future rewards that improve tool life, reduce vibration, and maintain stability. An appropriate value (e.g., 0.95) achieves a trade-off between immediate cutting efficiency and long-term machining optimization.

2.6. Dijkstra' s algorithm (DA)

DA generates optimal CNC tool paths by modeling movements as graph nodes and edges, enabling efficient shortest-path computation with reduced complexity.

DA was selected as the baseline planner for its deterministic shortest-path guarantees, computational simplicity, and independence from heuristic tuning, unlike A*. Its complementary role lies in providing the PPO-DDQNet agent with an initial feasible path, reducing exploration space and accelerating policy convergence. This baseline serves as a stable reference, enabling the DRL agent to focus on adaptive refinements under dynamic machining conditions, improving both efficiency and stability.

2.7. Proximal Policy Optimizer Driven Double Deep Q-Network (PPO-DDQNet)

The hybrid PPO-DDQNet framework integrates PPO for stable policy optimization and DDQNet for accurate value

Table 2. MDP Formulation for CNC Tool Path Optimization

Element	Symbol	Description	CNC Context
State	S	Observable parameters at time t	Position, load, vibration, wear, temperature
Action	A	Control decisions	Movement, feed rate, spindle speed, orientation
Reward	R(s, a)	Feedback for action	+ for smooth cutting; - for vibration/wear/collision
Transition	P (s′ ∣ s, a)	Next state probability	Deterministic (ideal) / Stochastic (real)
Discount	γ	Balance rewards	0.95 prioritizes long-term quality

estimation, enabling adaptive CNC tool path planning. The policy-value fusion mechanism (Eq. (2)) improves machining efficiency, automation, and product quality under dynamic conditions.

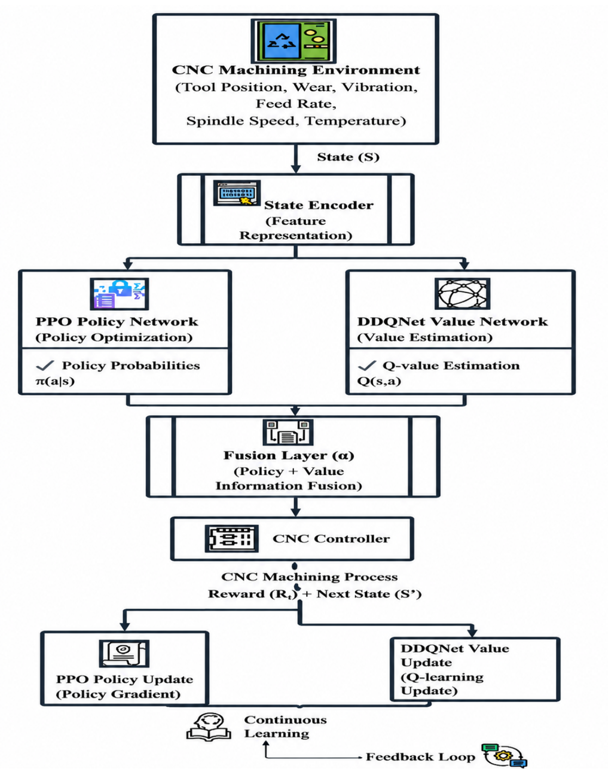

Fig. 2. . Hybrid PPO-DDQNet Architecture for Multi-Axis CNC Tool Path Optimization

Figure 2 illustrates the proposed hybrid PPO-DDQNet framework, where machining states are encoded and processed by the PPO policy network and DDQNet value network to generate optimized tool path decisions.

$$b_s^{\text{hybrid}} = \arg\max_b \Big[\alpha \cdot \pi_\theta(b_s \mid t_s) + (1-\alpha) \cdot \frac{Q_{\text{DDQNet}}(t_s, b)}{\sum_b Q_{\text{DDQNet}}(t_s, b)}\Big] \tag{2}$$

where b_s^{hybrid} is the final selected action, arg max denotes the action selection operator, α is the fusion coefficient, and $\pi_\theta(b_s \mid t_s)$ represents the probability of selecting action b_s in state t_s, as formulated in Eq. (2).

The fusion coefficient $\alpha \backslash alpha\alpha$ controls the relative contribution of the PPO policy probabilities and DDQNet Q-value estimates. A moderate value of $\alpha \backslash alpha\alpha$ enables stable policy learning while preserving accurate value estimation during dynamic CNC machining.

Exploration–exploitation is balanced through ε-greedy decay (ε: 1.0 $\rightarrow$ 0.01) in DDQNet and PPO entropy regularization, ensuring extensive exploration initially and stable convergence through gradual exploitation. The fusion coefficient α was empirically tuned to balance PPO policy optimization and DDQNet value estimation, improving decision stability, learning efficiency, and adaptive tool path optimization.

2.8. Double Deep Q-Network (DDQNet)

- **Deep Q-Network**

DQN learns optimal tool path decisions through trial-and-error interactions, updating Q-values to minimize machining errors, improve precision, and enable real-time path adaptation.

A target network update mechanism is employed in DDQNet to stabilize learning and prevent oscillations during training. The delayed update strategy reduces overestimation bias, enhances convergence reliability, and improves stability in long-horizon tool path optimization tasks. This approach significantly mitigates the curse of dimensionality, particularly in high-dimensional state spaces, by using a neural network function.

- **Double Deep Q-Network (DDQNet)**

DDQNet reduces Q-value overestimation through dual-network action selection and value evaluation, enabling stable learning and adaptive tool path optimization. The target value formulation of DDQNet is expressed in Eq. (3).

$$z(s) = H(s) + \mu\varrho\left(t' \underset{b*}{argmax} \varrho(\, t', b^* \; \theta(s)); \theta'(s)\right) \tag{3}$$

Here, $H(s)$ as the immediate reward, $\mu\varrho$ as evaluation function, t' as the next state, and b^* as online Q-network. The greedy policy constitutes the possibility of an agent that would respond randomly to encourage exploration rather than using the maximum Q-value for the subsequent state. This study employs the PPO-DDQNet framework to enable AI-driven predictive maintenance for adaptive tool path optimization in smart manufacturing.

2.9. Proximal Policy Optimizer (PPO)

PPO optimizes tool path policies using clipped objectives and policy gradients, enabling stable learning and trajectory refinement under dynamic conditions (Equation 4).

$$K^{CLIP}(\theta) = \widehat{F}_S \left[\min \left(q_s(\theta)\, \widehat{B}_s, clip\left(\epsilon, q_s(\theta)\right)\, \widehat{B}_s \right) \right] \tag{4}$$

Where, K^{CLIP} denotes the clipped surrogate function, and $q_s(\theta)$ represents the probability ratio in PPO.

For consistency throughout this study, the notation s denotes the machining state, a represents the selected machining action, r denotes the immediate reward, $Q(s,a)$ represents the action-value function, $\pi(a \mid s)$ denotes the policy function, θ represents the policy network parameters, γ is the discount factor, and α is the fusion coefficient used in the hybrid PPO-DDQNet framework.

Algorithm 1. PPO-DDQN

```python
import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim
from collections import deque
import random

class CNCEnvironment:
    def reset(self): ...
    def step(self, action): ...
    def get_dynamic_state(self): ...

class DDQNet(nn.Module):
    def __init__(self, state_dim, action_dim): ...
    def forward(self, x): ...

class PPOActorCritic(nn.Module):
    def __init__(self, state_dim, action_dim): ...
    def forward(self, x): ...
    def act(self, state): ...
    def evaluate(self, state, action): ...

env = CNCEnvironment()
ddqn_online = DDQNet(state_dim, action_dim)
ddqn_target = DDQNet(state_dim, action_dim)
ddqn_target.load_state_dict(ddqn_online.state_dict())
ppo_model = PPOActorCritic(state_dim, action_dim)
ddqn_optimizer = optim.Adam(ddqn_online.parameters(), lr = DDQN_LR)
ppo_optimizer = optim.Adam(ppo_model.parameters(), lr = PPO_LR)
replay_buffer = deque(maxlen = BUFFER_SIZE)
ppo_memory = []

for episode in range(num_episodes):
    state = env.reset()
    for t in range(max_steps):
        if random.random() < epsilon:
            action = env.sample_action()
        else:
            q_action = torch.argmax(ddqn_online(torch.tensor(state).float())).item()
            policy_action, log_prob = ppo_model.act(torch.tensor(state).float())
            action = combine_actions(q_action, policy_action)

        next_state, reward, done, _ = env.step(action)
        replay_buffer.append((state, action, reward, next_state, done))
        ppo_memory.append((state, action, reward, log_prob, next_state, done))
        state = next_state

        if len(replay_buffer) > BATCH_SIZE:
            minibatch = random.sample(replay_buffer, BATCH_SIZE)
            update_ddqn(ddqn_online, ddqn_target, minibatch, ddqn_optimizer)

        if t % UPDATE_FREQ == 0:
            update_ppo(ppo_model, ppo_memory, ppo_optimizer)
            ppo_memory = []

        if done:
            break
```

3. Results and discussion

Experimental results demonstrate that PPO-DDQNet improves CNC tool path optimization by reducing machining time, tool wear, and surface quality issues under dynamic conditions with varying execution constraints.

3.1. Simulation Setup

The proposed PPO-DDQNet framework was implemented in Python 3.8 and evaluated using L-shaped geometry models with an EMC54120 4F end mill cutter (20 mm diameter, five teeth, 35° helical angle, and 75 mm length).

Table 3. Evaluation of Models in Tooth Planning under Varying Time Constraints

Methods	Methods	30 seconds	5 minutes	10 minutes
DDQNet	La	1320.00	1185.00	1112.00
DDQNet	Lv	845.00	722.00	655.00
DDQNet	La+ Lv	2165.00	1907.00	1767.00
PPO-DDQNet [Proposed]	La	1205.00	1062.00	988.00
PPO-DDQNet [Proposed]	Lv	790.00	635.00	574.00
PPO-DDQNet [Proposed]	La+ Lv	1995.00	1697.00	1562.00

The proposed PPO-DDQNet framework used a replay buffer of 100,000 transitions, batch size 128, PER (α = 0.6), PPO clipping ϵ = 0.2, learning rates of 3e-4 (PPO) and 1e-3 (DDQN), Adam optimizer with gradient clipping (0.5), target network updates every 500 steps, and convergence after stable average loss for 50 consecutive episodes.

3.2. Result Analysis

PPO-DDQNet consistently outperforms DDQNet under restricted operating times. Table 3 compares angular loss (L_a), vibration loss (L_v), and total loss ($L_a + L_v$), demonstrating lower losses and improved performance under dynamic machining conditions. The proposed method achieves consistently lower losses; for example, angular loss reduces from 1112 to 988 under the 10-minute constraint, further confirming its robustness in maintaining stable performance under dynamic machining conditions.

Table 4 illustrates the running time comparison, showing that PPO-DDQNet achieves higher time efficiency than DDQNet across different epochs. The 17% computational time reduction at 1000 epochs demonstrates improved convergence efficiency. Although training time per epoch increases with model complexity, PPO-DDQNet reaches stable policies faster, reducing total training time and supporting real-time machining applications.

Th e effectiveness of the proposed PPO-DDQNet framework, additional comparative experiments were conducted using widely adopted reinforcement learning algorithms, including Soft Actor-Critic (SAC), Twin Delayed Deep Deterministic Policy Gradient (TD3), Advantage Actor-Critic (A2C), and Double Deep Q-Network (DDQNet). All algorithms were evaluated under identical machining conditions using the same L-shaped geometry, hyperparameter settings, and evaluation metrics. The comparative results are presented in Table 5.

Table 4. Comparison of Running Time between DDQNet and PPO-DDQNet Model

Epochs	DDQNet	PPO-DDQNet
50	70.80	63.40
100	67.58	60.32
150	66.70	59.14
200	65.78	57.96
250	64.50	56.87
300	63.18	55.72
350	62.10	54.62
400	61.32	52.46
450	60.24	52.32
500	59.36	51.20
550	58.48	50.08
600	57.64	49.04
650	56.84	48.18
700	56.02	47.02
750	55.24	46.36
800	54.48	45.68
850	53.68	44.96
900	52.97	44.12
950	52.30	43.46
1000	51.62	42.80

3.3. Visual Analysis Trial

In the L-shaped grid domain, Figure 3 shows tool traversal directions between grid nodes. Figure 3(a) exhibits dense overlapping bidirectional arrows, indicating repeated visits and excessive non-productive blanking movements. In contrast, Figure 3(b) follows a predominantly unidirectional trajectory with minimal backtracking, suppressing redundant blanking routes and achieving a cleaner, more efficient machining path.

The L-shaped geometry was discretized into a structured grid framework to facilitate systematic tool path evaluation. The 2D grid plane (Figure 4 a) maps planar spatial adjacency between neighboring machining regions, enabling efficient path connectivity analysis across the workpiece surface. The 3D grid plane (Figure 4 b) incorporates depth information across different cutting levels, which is essential for multi-axis operations as it models spatial constraints and supports collision-free trajectory planning. This structured representation ensures that the DRL agent can effectively evaluate tool movements at both the sur-

Table 5. Comparative Performance of PPO-DDQNet with State-of-the-Art Reinforcement Learning Algorithms

Algorithm	Average Cutting Time (s)	Angular Loss (L_a)	Vibration Loss (L_v)	Total Loss ($L_a + L_v$)	Surface Roughness (Ra μm)
Soft Actor-Critic (SAC)	75.8	1246	808	2054	1.24
Twin Delayed Deep Deterministic Policy Gradient (TD3)	78.9	1272	826	2098	1.32
Advantage Actor-Critic (A2C)	82.6	1318	862	2180	1.46
Double Deep Q-Network (DDQNet)	72.4	1205	790	1995	1.18
Proximal Policy Optimizer Driven Double Deep Q-Network (PPO-DDQNet)	68.1	1148	742	1890	1.05

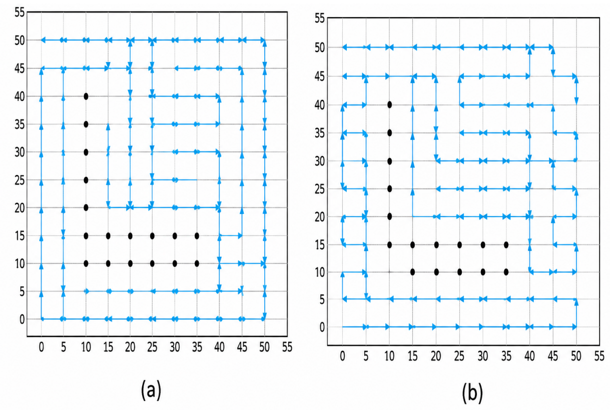

Fig. 3. Tool Path Visualization (a) DDQNet and (b) PPO-DDQNet Models

face and depth dimensions, contributing to the overall path quality improvements observed in the proposed method.

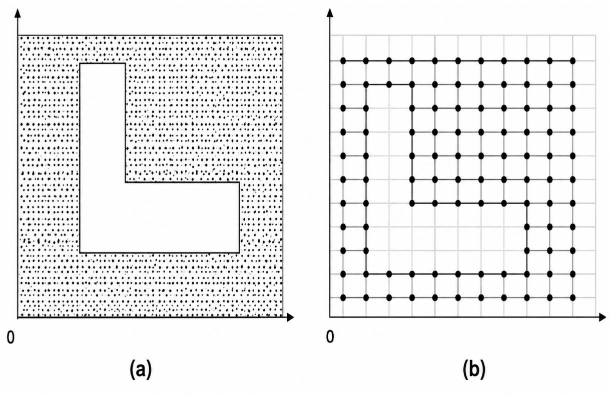

Fig. 4. Grid generation output of L-shaped Geometry Model (a) 2D-Plane and (b) 3D-Plane

3.4. Discussions

DRL enables adaptive CNC machining and fault identification through reward-based learning. However, it requires large datasets and computational resources with limited generalization across machining conditions [14]. CNN-based toolpath prediction suffers from poor generalization [11], while PPO-DDQNet improves adaptive learning, value estimation, cutting efficiency, and machining robustness. The computational complexity of the proposed PPO-DDQNet framework is primarily determined by the deep reinforcement learning training process and neural network inference stages. DDQNet introduces computational overhead, while PPO adds policy update complexity. GPU acceleration ensures stable convergence with lightweight inference for real-time CNC deployment.

4. Conclusion

The DRL model is used to optimize tool paths in the real-time process of feedback, including cutting speed, feed rate, and tool orientation, for dynamically optimizing tool life and productivity in manufacturing. Multi-axis CNC machines were used in modern production to precisely cut intricate parts. During the cutting process, real-world problems such as tool wear, vibration, heat, and material changes make it difficult to observe the predetermined tool paths. By converting 3D CAD component representations into voxel grids by using FGV, the system is able to examine the machining region as separate L-shaped geometries. An MDP was used to build environment modeling, taking tool motions as actions and factors such as spindle load, vibration, and position as states. The ideal tool routes were achieved by using the PPO-DDQNet technique, which guarantees accurate value prediction and stabilizes the learning process. The shortest path on the voxel grid was found by using DA, which enhances the DRL agent' s ability to make decisions during toolpath development. PPO-DDQNet achieves adaptive L-shaped planning with losses of 1205s (angular), 790s (vibration), and 1995s (total). Overall, the proposed framework demonstrates that AI-driven predictive maintenance significantly enhances machining efficiency, stability, and smart manufacturing performance.

4.1. Limitations and Future Scope

Dynamic machining environments were unexpected and presented challenges for real-time implementation, including rapid tool wear, workpiece deformation, and material irregularities that made it hard to precisely determine the model accuracy. Future studies should enhance DRL generalization, transfer learning, lightweight architectures, model compression, and real-time deployment across wider machining platforms. Future research should focus on enhancing real-time adaptability of DRL-based machining systems through transfer learning mechanisms, lightweight model compression, improved stability under uncertain conditions, and validation beyond L-shaped geometry to contour surface and cavity machining.

Declarations

Funding

The authors did not receive any funding for this study.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Data availability statement

The data generated and analyzed during the current study are available from the corresponding author upon reasonable request. The data are not yet publicly available due to ongoing research.

Code availability

Not applicable.

Authors' contributions

Xiaorong Zhou, Lidong Huang, and Xiaoping Liu contributed to the study conception and framework design, performance analysis, validation of results, and manuscript preparation. All authors read and approved the final manuscript.

References

[1] O. Tuyboyov, A. Baydullayev, A. Jeltuxin, and Z. Muxiddinov, (2024) *"Enhancing CNC machining tool path planning through reinforcement learning and optimization techniques"* **Applied Mechanics and Materials** ***923***: 49–58. DOI: 10.4028/p-cU4ff5.

[2] L. Zhang, H. Yu, C. Wang, Y. Hu, W. He, and D. Yu, (2025) *"A digital solution for CPS-based machining path optimization for CNC systems"* **Journal of Intelligent Manufacturing** ***36***(2): 1261–1290. DOI: 10.1007/s10845-023-02289-9.

[3] Y. F. Feng, H. Y. Ma, L. Y. Shen, C. M. Yuan, and X. Jiang, (2023) *"Real-time tool-path planning using deep learning for subtractive manufacturing"* **IEEE Transactions on Industrial Informatics** ***20***(4): 5979–5988. DOI: 10.1109/TII.2023.3342474.

[4] X. Zhao, C. Li, Y. Tang, X. Li, and X. Chen, (2024) *"Reinforcement learning-based cutting parameter dynamic decision method considering tool wear for a turning machining process"* **International Journal of Precision Engineering and Manufacturing-Green Technology** ***11***(4): 1053–1070. DOI: 10.1007/s40684-023-00582-9.

[5] D. Kalandyk, B. Kwiatkowski, and D. Mazur, (2024) *"CNC machine control using deep reinforcement learning"* **Bulletin of the Polish Academy of Sciences Technical Sciences**: e148940. DOI: 10.24425/bpasts.2024.148940.

[6] A. Pajaziti, O. Tafilaj, A. Gjelaj, and B. Berisha, (2025) *"Optimization of toolpath planning and CNC machine performance in time-efficient machining"* **Machines** ***13***(1): 65. DOI: 10.3390/machines13010065.

[7] K. Wang, S. Zhang, Y. Wu, and F. Jiang, (2025) *"Cutting path planning using reinforcement learning with adaptive sequence adjustment and attention mechanisms"* **The International Journal of Advanced Manufacturing Technology** ***136***(11): 5599–5612. DOI: 10.1007/s00170-025-15200-y.

[8] J. Dornheim, L. Morand, S. Zeitvogel, T. Iraki, N. Link, and D. Helm, (2022) *"Deep reinforcement learning methods for structure-guided processing path optimization"* **Journal of Intelligent Manufacturing** ***33***(1): 333–352. DOI: 10.1007/s10845-021-01805-z.

[9] Y. Zhang, Y. Li, and K. Xu, (2022) *"Reinforcement learning–based tool orientation optimization for five-axis machining"* **The International Journal of Advanced Manufacturing Technology** ***119***(11): 7311–7326. DOI: 10.1007/s00170-022-08668-5.

[10] F. Lu, G. Zhou, C. Zhang, Y. Liu, F. Chang, Q. Lu, and Z. Xiao, (2025) *"Energy-efficient tool path generation and expansion optimisation for five-axis flank milling with meta-reinforcement learning"* **Journal of Intelligent Manufacturing**: 1–25. DOI: 10.1007/s10845-024-02412-4.

[11] Y. Jiang, J. Chen, H. Zhou, J. Yang, P. Hu, and J. Wang, (2022) *"Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning"* **The International Journal of Advanced Manufacturing Technology *118***(1): 551–570. DOI: 10.1007/s00170-021-07895-6.

[12] P. Li, M. Chen, C. Ji, Z. Zhou, X. Lin, and D. Yu, (2024) *"An agent-based method for feature recognition and path optimization of computer numerical control machining trajectories"* **Sensors *24***(17): 5720. DOI: 10.3390/s24175720.

[13] Y. Wan, W. Xu, and T. Y. Zuo, (2023) *"Tool path optimization for complex cavity milling based on reinforcement learning approach"* **IEEE Access *11***: 66793–66807. DOI: 10.1109/ACCESS.2023.3262169.

[14] S. Liu, Z. Shi, J. Lin, and H. Yu, (2025) *"A generalisable tool path planning strategy for free-form sheet metal stamping through deep reinforcement and supervised learning"* **Journal of Intelligent Manufacturing *36***(4): 2601–2627. DOI: 10.1007/s10845-024-02371-w.

[15] C. Guo, Z. Wang, K. Li, L. Ye, N. Yu, and F. Gong, (2025) *"Domain knowledge integrated CAM system based on multi-objective path optimal planning and a deep convolutional neural network"* **Expert Systems with Applications**: 127788. DOI: 10.1016/j.eswa.2025.127788.