Solid Freeform Fabrication 2024: Proceedings of the 35th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference
Reviewed Paper

REINFORCEMENT LEARNING FOR ENERGY-EFFICIENT TOOLPATH GENERATION IN ADDITIVE MANUFACTURING

George E. Duke, David K. Somade, Niechen Chen

Department of Industrial and Systems Engineering, Northern Illinois University, Dekalb, IL 60115

Abstract

Toolpath design plays a significant role in determining the efficiency of Additive Manufacturing (AM) processes. Traditional toolpath optimization methods frequently depend on empirical methods, which may not adequately account for the complex dynamics of the printing process. This study introduces a novel reinforcement learning (RL) approach, leveraging Proximal Policy Optimization (PPO), to optimize toolpath generation with a particular focus on reducing energy consumption. A custom-built environment was created, simulating the toolpath planning scenario as a discrete grid space, where an RL agent representing the printing nozzle learns to navigate and optimize its path. The RL agent, implemented using Proximal Policy Optimization (PPO), was trained on grids of increasing complexity (10x10 and 25x25) using two reward systems: a default system and an energy-optimized system based on a custom energy model. The energy model penalizes energy-intensive vertical and diagonal movements while rewarding horizontal movements. Results from training showed that the energy-optimized model achieved a significant reduction in energy consumption without compromising toolpath efficiency. On the 10x10 grid, energy consumption decreased from 92.7 $kWms$ to 83.5 $kWms$, while on the 25x25 grid, it dropped from 400.2 $kWms$ to 395.4 $kWms$. Statistical analysis using paired t-tests confirmed these reductions with p-values of 0.00, demonstrating the effectiveness of incorporating energy constraints in RL training for AM. This research highlights the potential of RL in improving the sustainability and efficiency of AM processes through intelligent toolpath design.

Keywords: additive manufacturing, reinforcement learning, toolpath optimization, sustainable manufacturing

1 Introduction

Additive Manufacturing (AM), commonly known as 3D printing, has transformed the manufacturing industry, showing significant growth and innovative potential. The global market size of the market was $20.37 billion in 2023 and is expected to grow at a Compound Annual Growth Rate (CAGR) of 23.3% from 2023 to 2030. This rapid expansion is driven by the widespread adoption of AM across various industries, including healthcare, aerospace, and automotive, where it is used for applications ranging from prototyping to the manufacturing of complex, lightweight parts [1].

Optimizing energy usage in AM is crucial for enhancing the efficiency and sustainability of the production process. Factors influencing energy consumption include the type of materials used, the toolpath generation, build orientation and other specific parameters of the AM process. Efficient toolpath generation is crucial for reducing energy consumption by minimizing non-

productive movements of the print head where no material is deposited, thus significantly lowering the energy needed to drive the motors. Furthermore, optimizing motor acceleration and deceleration is key to reducing energy consumption by ensuring smoother transitions and minimizing energy spikes linked to abrupt speed changes [2], [3].

The adoption of artificial intelligence (AI) algorithms to optimize printing parameters for better surface finish, strength, and energy efficiency is gaining traction among AM researchers. Malviya and Desai (2019) proposed a machine learning based computational framework for build orientation optimization with the aim of maximizing resistance to failure under prescribed loading conditions [4]. They employed a single hidden layer Artificial Neural Network (ANN) and Bayesian optimization algorithm in their research. Vahabili and Rahmati (2017) developed an AI methodology to improve the quality of AM products by estimating surface roughness distribution in advance [5]. They optimized an ANN using trial and error and evolutionary algorithms and integrated it with particle swarm optimization (PSO) and imperialist competitive algorithm (ICA) to create the PSOICA algorithm. This enhanced the ANN's speed and accuracy in estimating surface roughness for Fused Deposition Modelling (FDM) parts. Pazhamannil et al. (2021) used an ANN to predict the tensile strength of PLA models made with FDM [6]. They used the Taguchi L9 orthogonal array to design experiments and tested how nozzle temperature, layer thickness, and infill speed affected tensile strength. The ANN, trained with this data, achieved a high accuracy of 99.9% and was validated with confirmation experiments, showing predictions within a 5% error margin. They found that lower layer thickness and higher nozzle temperatures improved tensile strength, while infill speed had negligible impact. This study demonstrated the ANN's usefulness in optimizing FDM process parameters.

Integrating AI algorithms in toolpath optimization offers a promising solution for enhancing AM efficiency. Instead of using a static pre-generated toolpath, using AI algorithms can create and adjust the next moves in real-time during the manufacturing process to minimize non-productive movements considering the dynamical changes during a process, thus resulting a true and realistic energy optimization strategy. RL is a revolutionary tool in AI used to optimize toolpath generation in AM. The study conducted by Mozaffar et al. (2020) highlights the effectiveness of RL in toolpath design, especially in environments with dense reward structures [7]. RL algorithms continuously enhance toolpaths by learning from each print job, refining predictions, and adjustments for future tasks. RL stands out due to its ability to adapt the toolpath in real-time, resulting in reduced energy consumption and enhanced efficiency of the AM process. Moreover, RL algorithms improve with each print job, allowing for a progressively better performance over time. This continuous learning process enables RL to refine its predictions and adjustments, leading to more efficient and sustainable manufacturing practices [8].

2 Methodology

2.1 Reinforcement Learning Algorithm

RL is a machine learning methodology in which an AI agent learns to make decisions by interacting with an environment to achieve a predefined objective. The agent receives feedback in the form of rewards and penalties, which it uses to enhance its decision-making abilities over time. This reinforcement learning approach is most appropriate for cases where an optimal solution has not been predetermined. RL algorithms exhibit superior adaptability compared to other machine learning algorithms, enabling them to promptly and effectively respond to real-time changes in the

environment. This attribute is advantageous in dynamic and unpredictable conditions. RL agents continuously learn and improve during each interaction with the environment, leading to a progressively enhanced performance over time.

Some prominent RL algorithms are Q-learning, SARSA (State-Action-Reward-State-Action), Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO). Q-Learning is an algorithm that aims to determine the optimal action to take in each state by learning the value of each action in each state. It is considered an off-policy algorithm. SARSA, on the other hand, is an on-policy algorithm that updates the Q-values based on the action taken by the agent. DQN combines Q-Learning with deep neural networks to effectively handle state spaces with high dimensions. PPO, on the other hand, is an advanced policy gradient method that effectively manages the trade-off between exploration and exploitation, resulting in consistent and reliable performance improvements.

The PPO algorithm was selected for this study due to its capacity to manage complex observation spaces and its reliable performance across various environments. This research employs the multi-layer perceptron (MLP) model from StableBaselines3 for the neural network architecture. The MLP model is specifically designed for the PPO algorithm. This architecture employs multiple layers to effectively process complex input data from the environment. Each layer of the MLP is meticulously designed to transform the input data through non-linear activations, allowing the network to learn and generalize from complex sensory input.

Modifications were implemented to the original PPO algorithm to accommodate noisy observations and to facilitate dynamic transitions between various environments during training. The modifications were executed to establish a more authentic and resilient model capable of adjusting to variations in the printing environment. The revised loss function includes the original PPO loss components—clipped surrogate objective, value function loss, and entropy bonus—while incorporating noise and dynamic environment switching mechanisms. The adjusted loss function is expressed as:

$$L^{CLIP+VF+S}(\theta, \eta_t, e_t) = \mathbb{E}_t\left[L_t^{CLIP}(\theta, \eta_t, e_t) - c_1 L_t^{VF}(\theta, \eta_t, e_t) + c_2 S[\pi_\theta]\left(s_t^{e_t}\right)\right] \quad (1)$$

Where η_t represents the noise added to the observation at each timestep t, and e_t indicates the environment that the agent is currently interacting with. The modified clipped surrogate objective function, $L^{CLIP}(\theta)$, is given as:

$$L^{CLIP}(\theta, \eta_t, e_t) = \mathbb{E}_t[\min\left(r_t(\theta)\hat{A}_t^{\eta,e}, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{\eta,e},\right)] \quad (2)$$

The probability ratio denoted as r_t is defined as the ratio of the policy $\pi_\theta\left(a_t^\eta \middle| s_t^\eta\right)$ to the old policy $\pi_{\theta_{old}}\left(a_t^\eta \middle| s_t^\eta\right)$. Where $\pi_\theta\left(a_t^\eta \middle| s_t^\eta\right)$ is the probability of selecting action, a, given state, s, under the policy π_θ based on the noisy observation, η. The advantage estimate, $\hat{A}_t^{\eta,e}$, is calculated as the difference between the expected return following the action and the value of the state.

The formula for the advantage estimation in PPO is given by:

$$\hat{A}_t^{\eta,e} = \delta_t^{\eta,e} + (\gamma\lambda)\delta_{t+1}^{\eta,e} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}^{\eta,e} \quad (3)$$

Where:

$$\delta_t^{\eta,e} = R_t + \gamma V(s_{t+1}) - V(s_t)$$

γ is the discount factor, which adjusts the importance of future rewards

λ is the trace decay parameter, which influences the weighting of future rewards

$V(s)$ is the value function estimating the expected return from state s

R_t is the reward received after taking an action at time t.

The surrogate objective is restricted by a clip function to stay within the range of $[1-\epsilon, 1+\epsilon]$, where ϵ is a hyperparameter called the clip range.

The value function is given as:

$$L^{VF}(\theta) = \left(V_\theta(s_t) - V_t^{target}\right)^2 \quad (4)$$

This loss function is made up of $V_\theta(s_t)$, which is the predicted value of state at time t under the current policy, and V_t^{target}, which is the target value for the state at time t.

An entropy bonus is also calculated to encourage action variation and prevent premature policy convergence. This methodical approach to calculating loss forms the foundation of the adaptive learning process, ensuring that every instance contributes to improving the agent's performance in navigating intricate toolpath environments.

$S[\pi_\theta](s_t)$ is the entropy of the policy

The data generated after each episode is crucial for calculating the loss function. This includes the observed grid, the agent's position, the signed distance function (SDF), gradients, sensor readings, and information about the nearest goal. This data allows for the assessment of the agent's actions through the advantage estimate, which determines if the actions taken were superior to what was statistically expected. This serves as the foundation of the policy loss component, which is crucial for updating the agent's decision-making policy. In addition, the mean squared error between the predicted and actual rewards helps improve the accuracy of our model's value predictions. This is achieved by integrating it through the value function loss.

2.2 Training Procedure

To enhance toolpath generation using RL, a custom environment was developed with OpenAI's Gym framework to simulate the print nozzle and print bed grid. The print nozzle operates within an 8-direction discrete action space, and the observation space includes the nozzle's position, the SDF of the grid, the position, and the distance to the nearest print cell. This setup offers a thorough state representation for the RL algorithm. The training environment was created using a tailor-made polygon generator which generated a series of connected polygons, which were then converted into grid arrays. In these arrays, cells representing print areas were marked as '1' while empty spaces were marked with '0'. To introduce a broad range of variations, 1,000 random grids were generated for each grid size.

The training process followed a curriculum learning approach which involved gradually increasing the complexity of the grids on which the RL agent was trained. The initial phase of training was carried out on 10x10 grids that divide the 2D manufacturing space into 10 rows by 10 column square grids, as shown in Figure 1(b). This provides a discretized environment to allow the agent to efficiently grasp fundamental navigation and decision-making strategies. This smaller grid size was ideal for laying the foundation for the agent's behavior, enabling it to develop a base

policy without being overwhelmed by the complexity of higher dimensioned grids. Upon achieving proficiency with the 10x10 grids, the agent was transitioned to the 25x25 grids, as shown in Figure 1(a). These larger grids introduced a higher level of complexity and demand more sophisticated decision-making strategies. The 25x25 grids challenged the agent's ability to adapt its learned policy to more intricate scenarios, further enhancing its navigation abilities. Each grid was subjected to 60 episodes of simulation, and the training was performed using a batch size of 4096.

For each grid dimension, two distinct models were trained: one with energy penalties during training and the other without. The use of a dual-model method allows for a direct comparison of how energy constraints affect the optimization of toolpaths. This approach provides valuable information on the advantages and disadvantages of incorporating energy considerations in the toolpath generating process.

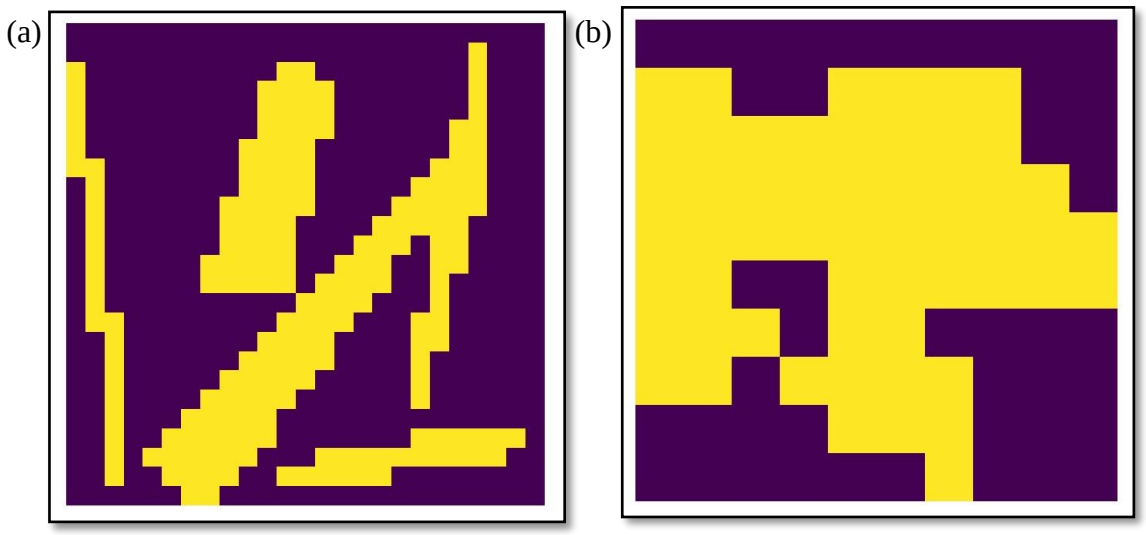

Figure 1: Grid resolutions. (a) 25x25 grid resolution; (b) 10x10 grid resolution. Yellow color marks the grids where AM deposition is required.

2.3 Reward mechanism

The reward mechanism implemented in the RL model was designed to optimize both path efficiency and energy consumption during the toolpath generation process. Two distinct reward mechanisms were utilized: the default reward system and an energy-based reward system.

2.3.1 Default reward system

In the default reward system, the RL agent is primarily rewarded for visiting print cells on the grid. A substantial reward is given to the agent if it completes the entire grid without exceeding the maximum number of allowed steps. In contrast, penalties are imposed when the agent revisits previously visited cells or moves into empty cells. This reward structure encourages the agent to minimize redundant movements, avoid inefficient detours, and complete the print task within the defined constraints.

2.3.2 Energy-based reward system

This reward system employs the same strategy of the default reward system while employing an additional strategy which focuses on energy consumption and is based on an energy model specific to the FDM printing developed by Somade [3]. This model distinguishes between movements along the X-axis and Y-axis, highlighting that Y-axis and diagonal motions require more energy than X-axis movements due to the greater energy required by the Y-axis motor and the increased energy demands from sudden acceleration. Consequently, movements along the X-axis receive rewards, with successive movements being progressively rewarded, thereby incentivizing the agent to formulate energy-efficient toolpaths that emphasize X-axis movements. On the other hand, penalties were awarded to diagonal and vertical movements to reflect the higher energy costs associated with these motions.

3 Results

The performance of the RL models was evaluated by analyzing toolpath efficiency and energy consumption. Toolpath efficiency was utilized as a crucial metric to assess the performance of the RL models. This is the ratio of the theoretical minimum number of printing steps (i.e. directly taken as the total grids to fill in.) to the actual steps taken by the RL agent. Higher toolpath efficiency represents that the RL model can complete the task in fewer steps. Energy consumption is based on our previously developed energy model for a FDM printing process that differentiates the X-axis motion from the Y-axis motion. In our energy model, Y-axis motion consumes more energy [3]. To assess the impact of energy penalties on the training of the reinforcement learning model, a paired t-test was conducted. The null hypothesis assumed that there is no significant difference in the mean values of toolpath efficiency and energy consumption between the energy model and the default model.

In our preliminary experiment, 4 RL-models were trained in total: 10x10 grid trained without energy penalties, 10x10 grid trained with energy penalties, 25x25 grid trained without energy penalties, and 25x25 grid trained with energy penalties. The performance of these four trained models is compared.

3.1 Toolpath Efficiency

In the 10x10 grid resolution, the agent achieved an average toolpath efficiency of 83% for the energy model and 82.5% for the default model, as illustrated in Figure 2.

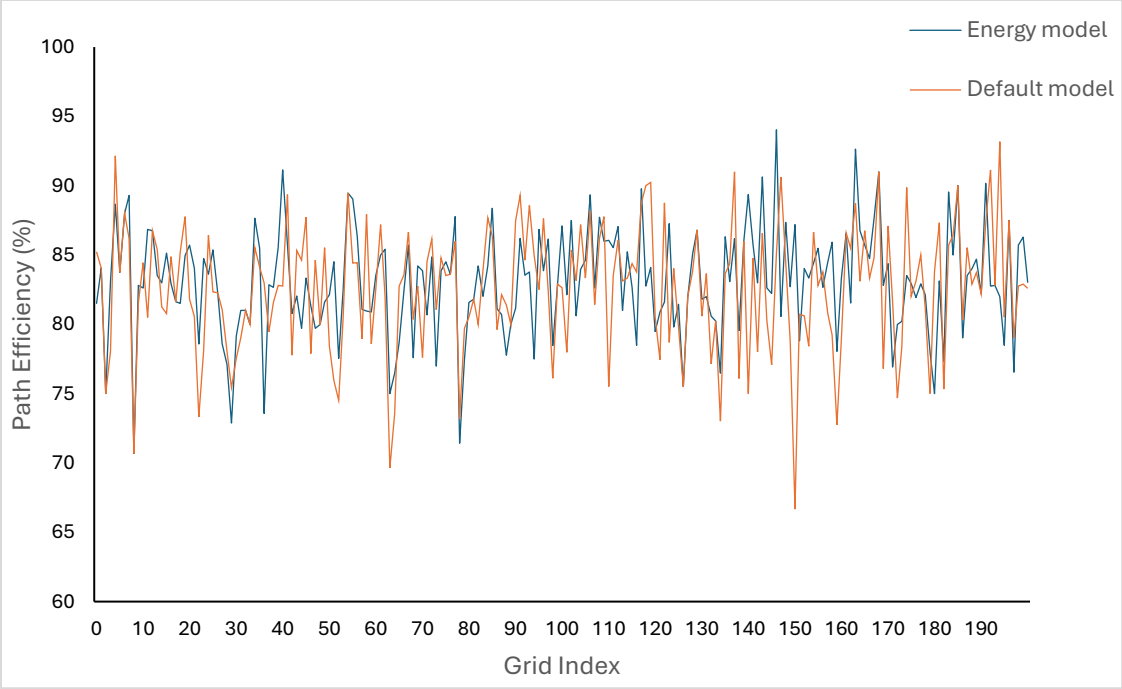

Figure 2: Path efficiency of RL models on 10x10 grids

For the default model and the energy model, the toolpath efficiency significantly reduced to 79.37% and 79.26% respectively on the 25x25 grid resolution, as shown in Figure 3.

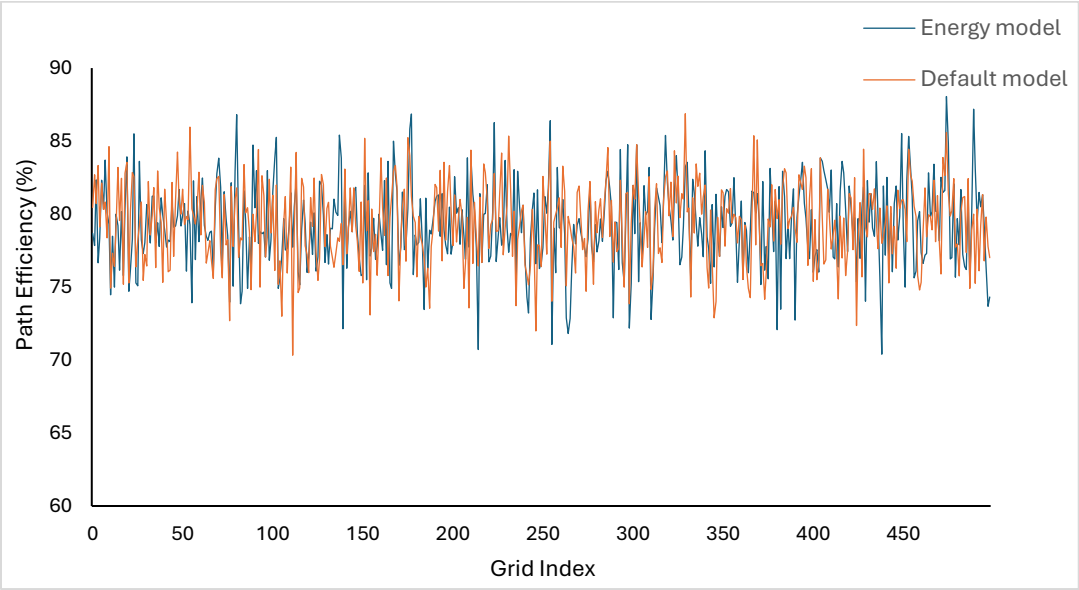

Figure 3: Path efficiency of RL models on 25x25 grids

A paired t-test for the mean scores of the toolpath efficiency of both models on the 10x10 grids gave a mean difference of 0.45, with a p-value of 0.12 (refer to Table 1). This shows there is no statistically significant difference in the toolpath efficiency of both models. In the comparison of the toolpath efficiency of the models on the 25x25 grids, the t-test also showed no significant

difference as the p-value obtained was 0.19 with a mean difference of -0.23. The statistical results indicate that the RL models are effective in generating toolpaths at a decent level of toolpath efficiency. The inclusion of energy penalties during training has no substantial impact on toolpath efficiency. The t-test results demonstrate no notable difference in efficiencies between the default and energy models for both grid sizes.

Table 1: Paired T-test results for path efficiency

Grid resolution	Estimation for Paired Difference		Test	
	Mean	StDev	T-value	P-value
10x10	0.45	1.09	1.56	0.12
25x25	-0.23	3.92	-1.30	0.19

3.2 Energy Consumption

This is a critical metric in evaluating the effectiveness of the RL models. The energy consumption of the toolpaths was computed using the energy consumption estimation model [3]. The energy consumption was measured for models trained with and without energy penalties to determine the impact of incorporating the energy penalties during training. For the 10x10 grid resolution, the default model has an average energy consumption of 92.7 $kWms$. In contrast, the energy-optimized model exhibited a reduced average energy consumption of 83.5 $kWms$, as highlighted in Figure 4.

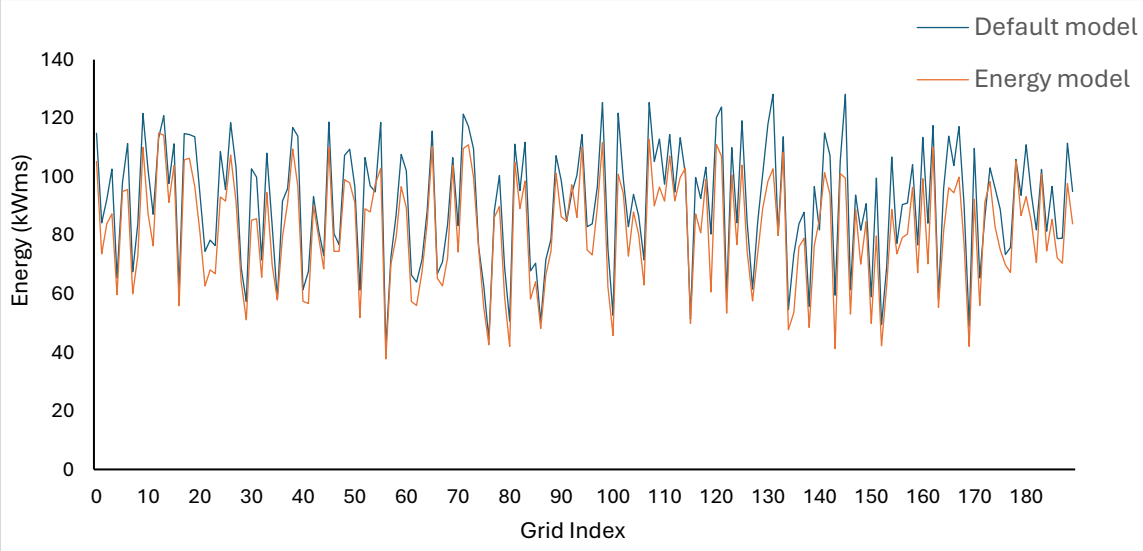

Figure 4: Energy consumption of RL models on 10x10 grids

In the 25x25 grid resolution, the default model attained an average energy consumption of 400.2 $kWms$, whereas the energy-optimized model highlighted enhanced efficiency with a reduced average consumption of 395.4 $kWms$ as depicted in Figure 5.

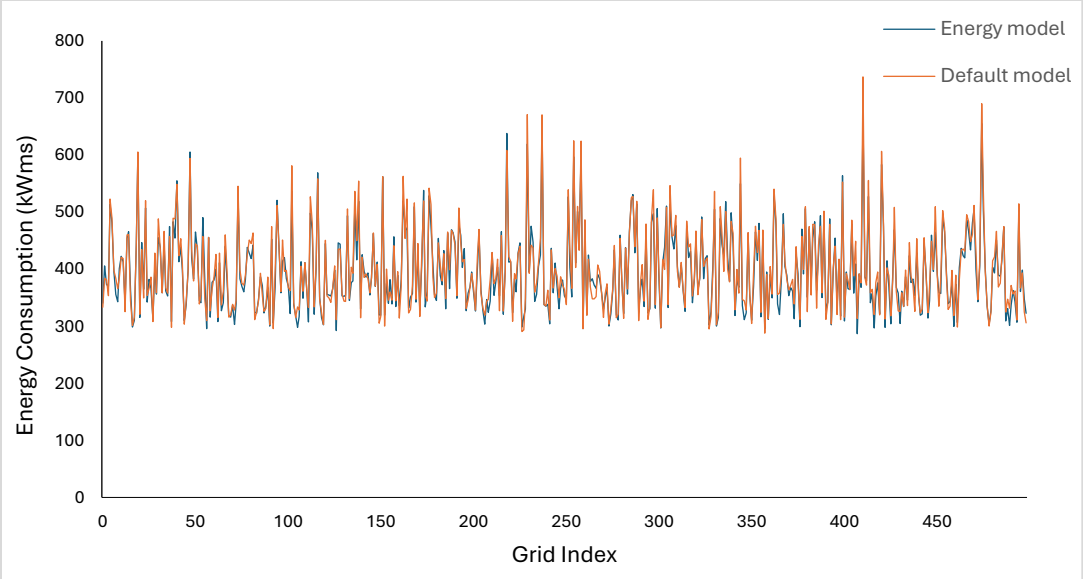

Figure 5: Energy consumption of RL models on 25x25 grids

In the statistical analysis of energy consumption across two grid resolutions for the RL models, notable disparities were found in the energy usage. In the 10x10 grid, the analysis showed a significant reduction in energy consumption by the energy-optimized model compared to the default model. The mean difference was $-9.2\ kWms$, which was statistically significant with a p-value of 0.00. Continuing in the 25x25 grid, the energy model consistently showed lower consumption compared to the default, with an average difference of $-4.8\ kWms$. This confirms substantial energy savings, as indicated by a p-value of 0.00, as shown in Table 2. These findings demonstrate the effectiveness of incorporating energy penalties into the model to decrease overall energy usage.

Table 2: Paired T-test results for energy consumption

Grid resolution	Estimation for Paired Difference		Test	
	Mean	StDev	T-value	P-value
10x10	-9.19	5.56	-23.37	0.00
25x25	-4.78	15.92	-6.70	0.00

4 Conclusion

This study highlights the potential of incorporating advanced machine learning techniques into manufacturing processes as the AM industry continues to grow. The evaluation of models trained with and without energy penalties showed significant improvement in energy efficiency, highlighting the effectiveness of the PPO algorithm. The results indicate that RL significantly optimizes toolpath efficiency. By incorporating energy penalties into the RL agent's training, the efficiency of the generated toolpaths remains largely unaffected. However, this approach does lead to a reduction in energy consumption for the toolpaths. This distinction emphasizes the efficiency of RL in optimizing energy usage during the toolpath generation process while still maintaining the overall efficiency of the toolpath.

5 References

[1] "Additive Manufacturing Market Size Report, 2030," Grand View Research. Accessed: Jun. 08, 2024. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/additive-manufacturing-market

[2] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim, "Demonstration-Guided Reinforcement Learning with Learned Skills." arXiv, Jul. 21, 2021. Accessed: May 21, 2024. [Online]. Available: http://arxiv.org/abs/2107.10253

[3] D. K. Somade, "Part Design Geometry-Driven Toolpath Optimization for Additive Manufacturing Energy Sustainability Improvement," Northern Illinois University, Dekalb, Illinois, 2023.

[4] M. Malviya and K. Desai, "Build Orientation Optimization for Strength Enhancement of FDM Parts Using Machine Learning based Algorithm," *Comput.-Aided Des. Appl.*, vol. 17, no. 4, pp. 783–796, Nov. 2019, doi: 10.14733/cadaps.2020.783-796.

[5] E. Vahabli and S. Rahmati, "Improvement of FDM parts' surface quality using optimized neural networks – medical case studies," *Rapid Prototyp. J.*, vol. 23, no. 4, pp. 825–842, Jan. 2017, doi: 10.1108/RPJ-06-2015-0075.

[6] R. V. Pazhamannil, P. Govindan, and P. Sooraj, "Prediction of the tensile strength of polylactic acid fused deposition models using artificial neural network technique," *Mater. Today Proc.*, vol. 46, pp. 9187–9193, Jan. 2021, doi: 10.1016/j.matpr.2020.01.199.

[7] M. Mozaffar, A. Ebrahimi, and J. Cao, "Toolpath design for additive manufacturing using deep reinforcement learning." arXiv, Sep. 29, 2020. Accessed: Jan. 24, 2024. [Online]. Available: http://arxiv.org/abs/2009.14365

[8] J. Tuo, X. Wang, X. Zhang, and P. Liu, "An Energy Utilization Prediction Method for FDM 3D Printing Processes," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, Auckland, New Zealand: IEEE, Aug. 2023, pp. 1–8. doi: 10.1109/CASE56687.2023.10260409.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms." arXiv, Aug. 28, 2017. doi: 10.48550/arXiv.1707.06347.