Enhancing Flow Control through Transformer-based Reinforcement Learning

Daehyung Lee [a], Joongoo Jeon[b], Hyun Sun Park [a*]
[a]*Department of Nuclear Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, South Korea*
[b]*Jeonbuk National University, Graduate School of Integrated Energy-AI, 54896 Jeonju-si, South Korea*
[*]*Corresponding author: hejsunny@snu.ac.kr*

Keywords : transformer, flow control, reinforcement learning, machine learning, Rayleigh-Bènard convection, Nusselt number

1. Introduction

Rayleigh-Bénard Convection (RBC) is a convective phenomenon that occurs when a temperature difference is applied between the lower and upper boundaries, with the lower boundary being hotter than the upper boundary. This convection plays a crucial role in both natural and industrial processes, significantly affecting energy efficiency and system stability through heat transfer. For example, controlling RBC is essential in atmospheric circulation on Earth, ocean currents, heat transfer in chemical processes, and cooling systems for electronic devices. In the nuclear power plant, heat control is critical to ensure the safety and efficiency of operations during normal conditions, transients, and accident scenarios.

The goal of RBC control is not only to suppress or regulate the unstable convection occurring within the system to achieve the desired heat transfer characteristics but also to explore the nature of RBC. In this study, the primary objective is to suppress heat transfer by minimizing the Nusselt number, a dimensionless number that measures the amount of heat transferred by convection that indicates effective suppression of convection. However, the inherent unstable and chaotic nature of RBC makes control extremely challenging. Even small control or observation delays can make the system uncontrollable (Beintema et al., 2020 [1]). Due to these uncertainties and chaotic characteristics, traditional linear control methods have limitations, necessitating new approaches for effective control.

Beintema et al. (2020) introduced reinforcement learning for RBC control, demonstrating superior performance compared to traditional linear approaches. Vignon et al. (2023) [2] showed that using Multi-Agent Reinforcement Learning (MARL) can enhance control performance, although this method is complex and time-consuming. Recent attempts have also been made to improve performance by incorporating techniques such as Group Invariant Networks and Positional Encoding. However, this study integrates the transformer network, which has recently gained prominence in various fields [3], into the Actor-Critic structure, achieving more efficient and faster control using a Single-Agent Reinforcement Learning (SARL) with simple data augmentation. Transformers, originally successful in natural language processing, excel in handling complex correlations.

In the study, to suppress convection in the RBC system by minimizing the Nusselt number, a reinforcement learning technique is used. Specifically, a method is developed to reduce the amount of heat transferred to the upper boundary by controlling the temperature distribution of the lower boundary. The Proximal Policy Optimization (PPO) algorithm [4] is employed, and the attention mechanism [5] is incorporated into the Actor-Critic network structure, resulting in faster and more efficient learning compared to traditional neural network-based methods. The findings of this study provide important insights into the development of autonomous operation and digital twin technology in the nuclear industry through thermal hydraulic system control.

2. Methods and Results

2.1 Problem Definition

2.1.1 Concept of Rayleigh-Bénard Convection (RBC)

RBC is a thermal convection phenomenon that occurs when a temperature difference between the lower and upper boundaries exists. When the lower boundary is hotter, the resulting buoyancy effect causes the fluid to rise, cool near the upper boundary, and then descend, resulting in a circulation pattern. This circulation forms unstable convection patterns, which significantly affect heat transfer efficiency. The Non-dimensionalized governing equations of RBC are as follows:

- Continuity

$$\nabla \cdot u = 0 \tag{1}$$

- Momentum

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\nabla p + \sqrt{\frac{Pr}{Ra}}\nabla^2 u + Tj \tag{2}$$

- Energy

$$\frac{\partial T}{\partial t} + u \cdot \nabla T = \frac{1}{\sqrt{RaPr}}\nabla^2 T \tag{3}$$

Fig. 1 shows the temporal evolution of temperature and velocity fields between two boundaries from 10 seconds to 300 seconds.

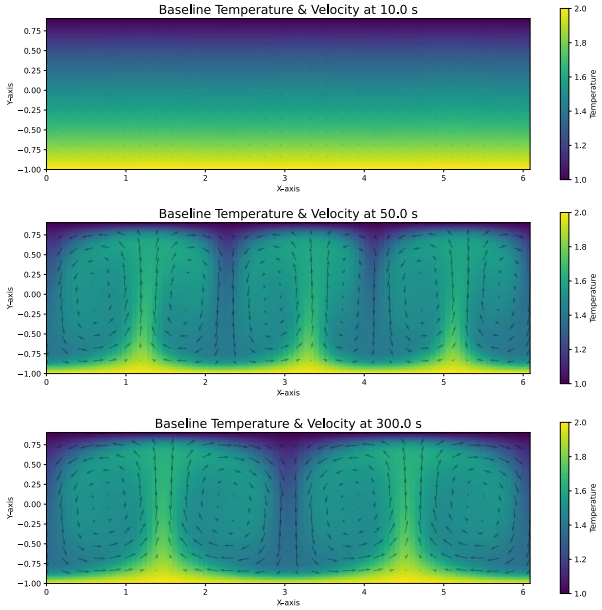

Fig. 1. Temporal evolution of temperature and velocity fields in the baseline RBC.

2.1.2 Definition of the Control Problem

In this study, to control the temperature of the lower boundary, the boundary is divided into 10 segments, and the temperature of each segment is chosen by the actions of the reinforcement learning agent. However, the temperature of the upper boundary is fixed, and the average temperature of the lower boundary is maintained constant. The agent learns to adjust the temperature of each segment to minimize heat transfer across the entire system.

2.1.3 Nusselt Number and Control Objectives

The Nusselt number is a dimensionless indicator of heat transfer within the system, used to evaluate heat transfer efficiency. A higher value indicates that more heat is being transferred by convection, and the control objective is to minimize this value to suppress convection. Specifically, the instantaneous Nusselt number is defined as follows, which is a function of heat flux $q(t)$:

$$Nu_{\text{inst}} = \frac{q(t)}{\kappa(T_H - T_C)/H} \tag{4}$$

2.2 Simulation Setup and Environment

2.2.1 Simulation Environment Setup

The simulations were performed using the Python-based open-source numerical analysis package shenfun [6], which models the 2D RBC system and computes the convection between two boundaries. The temperature of the lower boundary is divided into 10 segments and adjusted based on the agent actions, while the effects of these temperature changes on the Nusselt number across the entire system are evaluated. The example of temperature distribution is shown in Fig. 2.

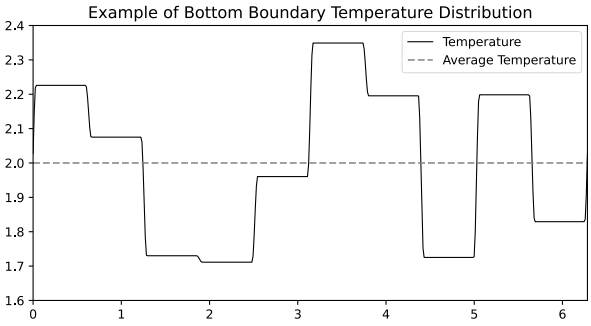

Fig. 2. Example of temperature distribution on the bottom boundary.

2.2.2 Definition of Simulation Parameters

Key parameters used in the simulation include the Rayleigh number, Prandtl number, grid size, and time step. The Rayleigh number indicates the strength of convection, while the Prandtl number represents the ratio of fluid viscosity to thermal diffusivity. The grid size determines the resolution of the simulation, and the time step affects the accuracy and computation speed of the simulation. The key variables in the simulation are summarized in the following table:

Table I: Parameters of the Simulation

Parameter	Value
Domain size $L \times H$	$2\pi \times 2$
Galerkin modes	96×64
Time step (sec)	0.02
Pr	0.7
Ra	10^4
Thermal expansion coefficient, β	0.0015
Action scaling factor, C	0.75
Number of observation probes	8×32
Number of CFD episodes	350
Number of action steps per episode	200
Number of control segments, N	10
Baseline duration (sec)	300
Action duration (sec)	1.5
Episode duration (sec)	300

2.3 Reinforcement Learning Framework

2.3.1 Reinforcement Learning Concept and PPO Algorithm

Reinforcement learning is a machine learning method where an agent learns to take optimal actions in a given environment to maximize rewards (Fig. 3). In this study, the PPO algorithm is used to develop a reinforcement learning-based temperature control strategy. PPO is a policy-based reinforcement learning algorithm that optimizes the agent's action policy to maximize rewards. The algorithm uses a clipping technique to limit policy changes, ensuring stable learning and high performance.

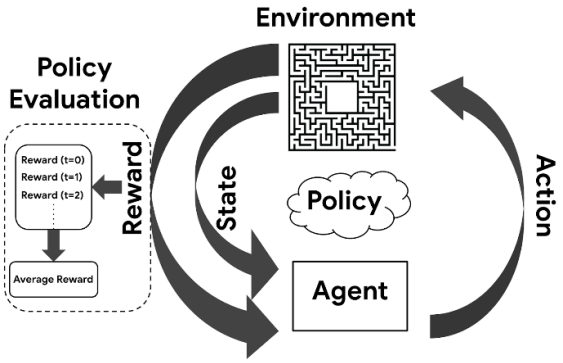

Fig. 3. Concept of Reinforcement Learning. [7]

Table II: Pseudo code of PPO Algorithm [4]

Algorithm 1 PPO, Actor-Critic Style

for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

2.3.2 Actor-Critic Structure and Transformer Network

In reinforcement learning, the Actor-Critic structure separates the policy (Actor) and value function (Critic) for learning. In the PPO algorithm, the Actor determines which action to take given a state, while the Critic evaluates the value of that state.

The Actor-Critic network used in this study is based on a transformer structure with multiple encoder blocks, designed to more effectively learn the (state-action) and (state-value) relationships in reinforcement learning. Unlike traditional sequential neural networks, the transformer model utilizes the self-attention mechanism to effectively learn the complex heat transfer patterns in the RBC system. This contributes to the agent learning more sophisticated control strategies. Transformers use multi-head self-attention to integrate various perspectives for each state, then learn to select the optimal action.

The network architecture used in this study is shown in Fig. 4. It consists of embeddings of size 128, with 4 heads and 4 transformer encoder blocks. The input state, flattened into one dimension, is added to positional embeddings and then passes through the multi-head attention layers and feed-forward layers of the transformer blocks four times. This network has approximately 500,000 parameters, making it more parameter-efficient than the conventional network with two hidden layers of size 512, which has about 660,000 parameters.

Additionally, data augmentation was performed by applying translations and y-axis symmetry to both the state and actions during policy training, taking advantage of the periodicity and symmetry of the simulation domain.

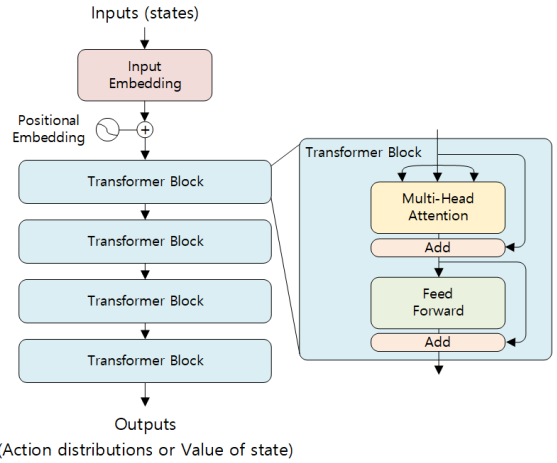

Fig. 4. Structure of transformer networks for actor-critic.

2.4 Results

In Fig. 5, the green line represents the performance of the reinforcement learning algorithm developed in this study (PPO using a transformer architecture as an actor-critic network), while the blue and orange lines show the results of the PPO with SARL and MARL algorithms, respectively. The dashed lines represent the final Nusselt number of each method per episode, and the bold lines represent the 25-episode moving average of the final Nusselt number per episode.

The initial Nusselt number for all algorithms starts at approximately 2.6. During the initial episodes, each algorithm adapts to the environment and learns to reduce the Nusselt number. All three cases successfully reduce the Nusselt number in the early stages, but the transformer network proposed in this study consistently and rapidly achieves further reductions. While the highest performance was observed in the MARL case, the transformer case achieved high performance more quickly and provided more stable control with lower variance compared to other cases. This indicates that the transformer network was able to effectively learn the

complex correlations of convection patterns, leading to a more stable control strategy.

In detail, as shown in Fig. 6, the TR case shows that in most episodes, the Nusselt number decreases gradually at first, followed by a sharp decline after 150 seconds, ultimately stabilizing around 2.1. This suggests that the transformer network effectively controls the convection patterns. On the other hand, in the MARL case, a significant decrease in the Nusselt number is observed only in the 210th and 240th episodes, with little change in other episodes.

Fig. 7 shows the temporal evolution of the temperature and velocity fields in the transformer case from 350 seconds to 600 seconds. In the final state, as shown in the figure, the fluid stabilizes into a converged form, where the convection pattern merges into a single large cell. This indicates that the convection has transitioned from an unstable, multi-vortex structure to a single stable state.

Compared with the previous MARL studies, the results show that although the final performance was slightly lower, the learning speed and efficiency provided significant advantages. This suggests that the method could be highly useful in fields such as the nuclear industry, where real-time control is critical.

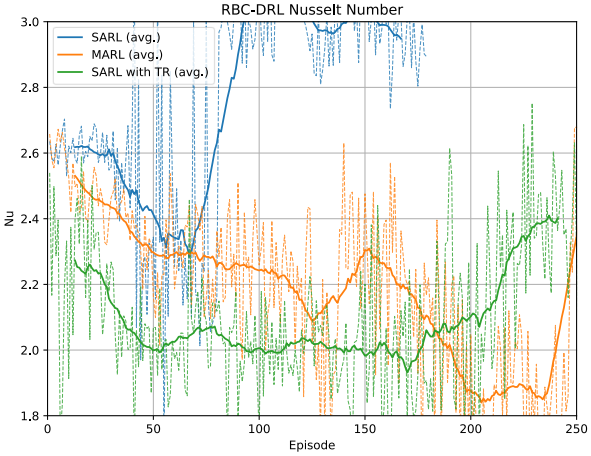

Fig. 5. Results of RBC-DRL cases.

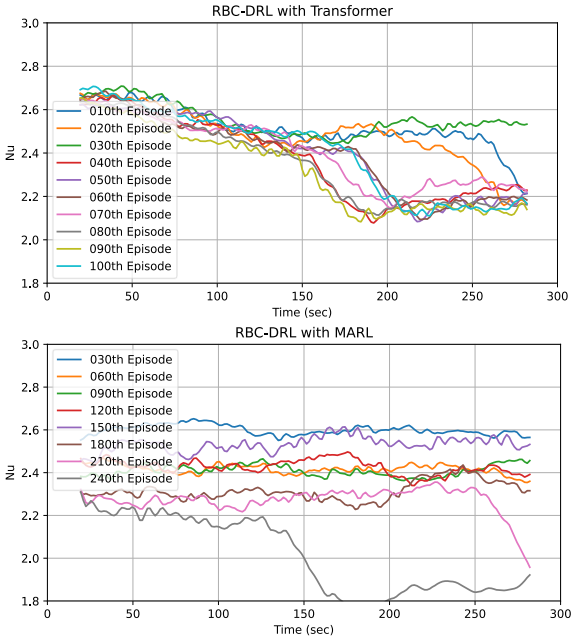

Fig. 6. Nu changes during episodes for each case (TR, MARL).

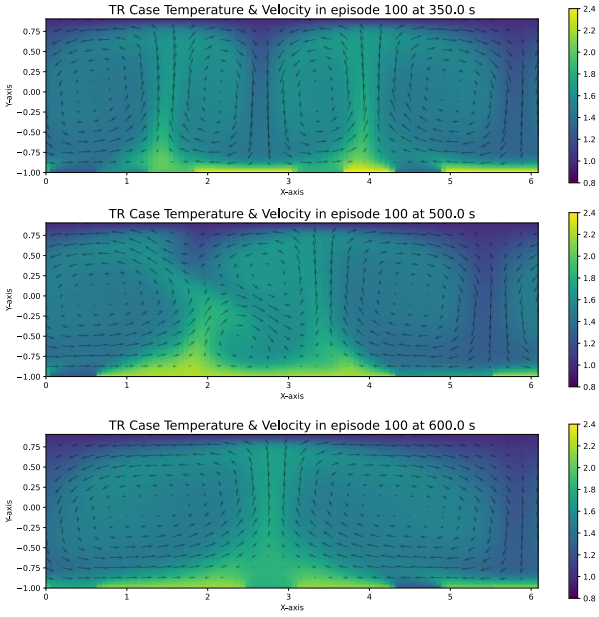

Fig. 7. Temporal evolution of temperature and velocity field in the TR case episode 100.

3. Conclusions

This study proposed a reinforcement learning method for controlling convection in the RBC system, aimed at minimizing the Nusselt number. Unlike previous studies that primarily focused on MARL for improving control performance, this study significantly enhanced learning speed and efficiency by incorporating a transformer network into the SARL framework. The key contribution of this research lies in the combination of a state-of-the-art network with the PPO algorithm, enabling faster and more efficient convection control compared to traditional

methods using simple neural networks.

The Actor-Critic reinforcement learning algorithm incorporating the transformer network demonstrated excellent performance in controlling complex convection patterns. The proposed temperature control method effectively reduced the Nusselt number compared to existing SARL methods, indicating improved flow control performance. Although the final performance was somewhat lower than MARL results, the proposed method achieved high performance much more quickly, with simpler implementation and faster learning, making it more applicable in industries where real-time control is essential.

The reinforcement learning-based convection control method developed in this study has potential applications in autonomous operation systems and accident management systems within the nuclear industry. For example, it could enhance plant stability by controlling heat transfer in the reactor cooling system or effectively managing convection around fuel rods. Additionally, this method shows potential for application in real-time monitoring and control of complex physical phenomena in actual systems through digital twin technology.

This research focused on confirming the feasibility of combining reinforcement learning with transformer networks for thermal hydraulic control. Future research could expand in the following directions:

- Comparison and enhancement of reinforcement learning algorithms: Testing different algorithms to further improve control performance or exploring hyperparameter tuning for better outcomes
- Application in complex fluid environments: Exploring the applicability in more complex environments such as turbulence control or three-dimensional flow systems
- Implementation of real-time control systems: Implementing systems capable of real-time control to evaluate practical applicability in industrial settings

This study identified new possibilities for reinforcement learning-based heat transfer control by incorporating the transformer architecture, which offers significant advantages due to its powerful performance across various domains.

ACKNOWLEDGEMENTS

This work was supported by the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (Grant No. 2106033-0222-CG100).

REFERENCES

[1] G. Beintema, A. Corbetta, L. Biferale, and F. Toschi, “Controlling Rayleigh–Bénard convection via reinforcement learning,” Journal of Turbulence, vol. 21, no. 9–10, pp. 585–605, Oct. 2020.
[2] C. Vignon, J. Rabault, J. Vasanth, F. Alcántara-Ávila, M. Mortensen, and R. Vinuesa, “Effective control of two-dimensional Rayleigh–Bénard convection: Invariant multi-agent reinforcement learning is all you need,” Physics of Fluids, vol. 35, no. 6, p. 065146, Jun. 2023.
[3] S. Islam et al., “A Comprehensive Survey on Applications of Transformers for Deep Learning Tasks,” Jun. 11, 2023.
[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Aug. 28, 2017.
[5] A. Vaswani et al., “Attention Is All You Need,” Aug. 01, 2023.
[6] M. Mortensen, "Shenfun: High performance spectral Galerkin computing platform," Journal of Open Source Software, vol. 3, no. 31, p. 1071, 2018.
[7] R. S. Alonso, I. Sittón-Candanedo, R. Casado-Vara, J. Prieto, and J. M. Corchado, “Deep Reinforcement Learning for the Management of Software-Defined Networks and Network Function Virtualization in an Edge-IoT Architecture,” Sustainability, vol. 12, no. 14, p. 5706, Jul. 2020.