# Generating Daily Load-Following Operation Scenarios by using Pre-trained Reinforcement Learning for Nuclear Power Plants

**Junhyeong Bang[a], Jonghyun Kim[a*]**
[a] Department of Nuclear and Quantum Engineering, Korea Advanced Institute of Science and Technology, 291, Daehak-ro, Yuseong-gu, Daejeon 34141, Republic of Korea
[*] Corresponding author: jonghyun.kim@kaist.ac.kr

**Abstract:** As renewable penetration increases, power supply variability is growing, requiring more flexible plant operation. This demand now extends to nuclear power plants (NPPs), which have traditionally operated at a constant power level. In this context, Daily Load-Following Operation (DLFO), in which a plant cycles between low and high power levels daily, has been discussed as an alternative. However, since repeated power transitions introduce additional thermal and neutronic stresses on the reactor core, DLFO must be conducted within safety-constrained operating bounds. To perform DLFO, a scenario matching the target power profile must be prepared in advance, and this process has so far relied on expert trial and error by using system codes or software. A DLFO scenario is time-series data of operating variables that achieve the target power from a given initial core state while satisfying safety constraints. Traditionally, experts manually designed such scenarios and verified them using detailed analysis codes, revising them until the criteria were met. However, this manual process is computationally expensive and makes it difficult to obtain scenarios covering diverse initial conditions and long operating periods. Reactor behavior is also nonlinear and history-dependent, so the required strategy changes with the current state and past operating history. This creates a need for AI-based automation that can learn the relation between operating conditions and control strategies. Supervised learning (SL) requires many input-output examples, but each scenario is still expensive to generate. Reinforcement learning (RL), in which an agent learns a policy through interaction and rewards, can reflect both target tracking and constraint satisfaction, but training from scratch can be inefficient and unstable. To address these limitations, this study proposes a framework that combines pre-training with RL. First, a sequence model combining LSTM and a Transformer-based architecture is designed to capture past operating history. The LSTM encodes the historical sequence, while the Transformer processes the current state and generates operational scenarios. Second, this model is pre-trained via SL with a small set of practically obtainable baseline DLFO scenarios, allowing it to internalize fundamental operating patterns and physical relationships. Third, the pre-trained model is employed as the initial policy of the RL agent, enabling physically informed exploration during early training. Finally, the study presents a framework for generating DLFO operational scenarios that satisfy safety constraints while reflecting the history-dependent physical behavior of the reactor core.

## 1. INTRODUCTION

As the share of renewable energy in the power grid increases, the variability of electricity supply and demand is growing. Wind and solar power in particular are expected to account for approximately 27% of generation by 2030 [1], and since their output is high during the day and low at night, the grid output fluctuates on the daily cycle [2]. To accommodate this variability, nuclear power plants (NPPs) which have traditionally operated at constant full power as baseload units are increasingly required to perform Daily Load-Following Operation (DLFO), in which the core power is adjusted on the daily cycle [3]. The European Utility Requirements (EUR) and the Electric Power Research Institute (EPRI) recommend a 100-50-100 maneuver, in which power is lowered from full power to 50% and returned, with each transition completed within two hours and a maximum ramp rate of 30% per hour [4,5]. DLFO should be performed while the Axial Shape Index (ASI), which characterizes the axial power distribution of the core, remains within its safety limit [6].

The operating safety condition requires |ASI| < 0.3 as a safety constraint [8]. At |ASI| = 0.3, the ratio of the lower- to upper- half integrated power reaches approximately 1.9 to 1 such a large axial power imbalance increases the axial power peaking, which reduces the thermal margin such as Departure from Nucleate Boiling Ratio (DNBR) and induces xenon-driven axial power oscillations, thereby compromising core safety [9].

Core thermal power is controlled by the soluble boron concentration and the control element assembly (CEA). Boron control maintains the ASI stably, however, responds slowly and remains as radioactive waste, so its use is preferably minimized. In i-SMRs (innovative small modular reactors), which adopt boron-free operation as a base operation, therefore CEA becomes the main control variables [10]. Moving the CEA, however, skews the power distribution and increases |ASI|. The problem addressed in this study is thus to find a CEA operation scenario that follows that 100-50-100 power schedule while keeping |ASI| < 0.3 throughout the entire operation.

Such a scenario must be secured before operation and is defined as the time series of operating variables that achieves the target power from a given initial core state. Traditionally, experts have designed and verified these scenarios by trial and error using core analysis codes [11]. However, this manual approach is computationally expensive and cannot cover diverse initial conditions and long operating periods. More fundamentally, core behavior is nonlinear and history-dependent since the xenon concentration, which is difficult to measure, accumulates according to past power changes, the required control strategy varies with the operating history even for an identical current state [12].

Data-driven automation that learns the relationship between operating conditions and control strategies is therefore needed. Supervised learning (SL) is accurate but requires a large number of examples, and obtaining sufficient data satisfying |ASI| < 0.3 is difficult. Reinforcement learning (RL) can reflect both target-power tracking and constraint satisfaction in its reward but learning from random exploration is unstable [13]. A combined strategy that pre-trains the policy on a small set of scenarios and then fine-tunes it with RL is thus required.

This study proposes a framework that combines supervised pre-training and RL to generate 100-50-100 DLFO scenarios satisfying the |ASI| safety constraint. A sequence model combining a Long Short-Term Memory (LSTM) network and a Transformer is used as the actor network to reflect the past operating history. The actor is first pre-trained on a small set of baseline scenarios and then transferred as the initial policy of a Proximal Policy Optimization (PPO) agent. The remainder of this paper is organized as follows. Section 2 describes the methodology, Section 3 the pre-training stage, and Section 4 the RL stage.

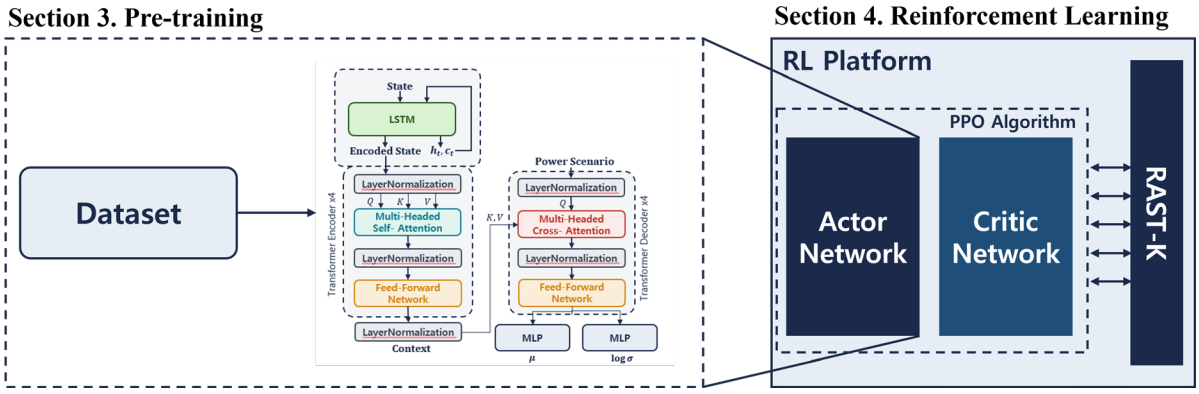
## 2. Method

The proposed methodology integrates the two stages as complementary roles. Pre-training places the actor in a region of the policy space where the |ASI| safety constraint is largely satisfied, providing RL with a stable starting point rather than the random initialization that would otherwise cause frequent early termination. RL then refines the actor against an ASI-based reward, improving the ASI margin and exploring diverse operating strategies for the same core state.

### 2.1. Proposed Framework

The pipeline of the proposed framework proceeds in two sequential stages as shown in Fig. 1. In the pre-training stage, baseline DLFO scenarios produced by the proposed algorithm are used to train the actor network by SL with the Mean-Squared-Error (MSE) loss. The resulting weights initialize the actor of the PPO agent. In the RL stage, the PPO agent interacts with the environment coupled with RAST-K, generating CEA-speed actions, receiving ASI-based rewards, and updating both the actor and the critic. The detailed procedures of the two stages are described in Section 3 and Section 4, respectively.

**Fig 1. Overview of this study.**




## 2.2. Agent Network Architecture

The target of pre-training is the actor network that generates actions. PPO uses an actor and critic. The critic, which only computes a value that evaluates whether an action is good or bad, is simpler than the actor, which directly determines the actions. The structure of the actor network is shown in Fig. 2a.

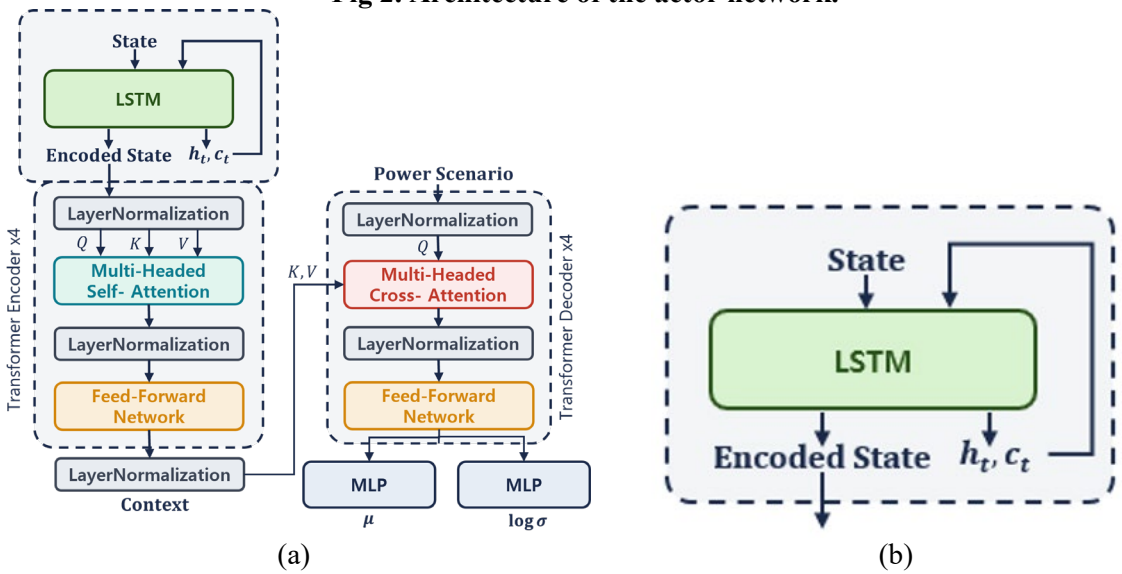
The actor network uses an LSTM as its input head to reflect the past core operating history. This history is key information for estimating the unmeasurable xenon distribution and is decisive for generating CEA speeds that satisfy the ASI. Since the length of the history varies, this study reuses the LSTM cell state and hidden state for the next input, carrying the context over a history of arbitrary length to produce an embedding state as shown in Fig. 2b. Unlike a fixed time window, this approach imposes no limit on the history length. The generated embedding state is used as the input to the Transformer, which extracts information through positional encoding is given in Eq. 1 and attention is given in Eq 2 [14].

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d}}\right), \quad \mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d}}\right) \tag{1}$$

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Here, $pos$ is the position in the sequence, $i$ is the embedding-dimension index, $d$ is the embedding dimension, $Q, K,$ and $V$ are the query, key, and value matrices, and $d_k$ is the dimension of the key vector.

**Fig 2. Architecture of the actor network.**



### 2.3. Method of Reinforcement Learning

Since the CEA speed is a continuous variable, value-based RL methods such as Q-learning are not directly applicable, and policy-based methods that produce the action directly are more natural [15]. Among policy-based methods, vanilla policy gradient does not limit the magnitude of each update and is therefore prone to divergence, while trust Region Policy Optimization (TRPO) guarantees stable updates but requires explicit handling of the trust-region constraint and is complex to implement and tune [16]. PPO limits the size of each update by a simple clipping of the policy probability ratio, achieving stability comparable to TRPO with a much simpler implementation, and is widely used as the standard method for continuous-action RL [17]. Moreover, in the present setting where RAST-K simulations are computationally expensive, the clipped surrogate objective enables stable reuse of the same samples across multiple updates and is therefore also advantageous in term of sample efficiency. For these reasons, this study adopts PPO. The explicit form its loss function and advantage estimator is presented in Section 4.

## 3. PRE-TRAINING

### 3.1. Purpose of Pre-training

Learning a CEA scenario with RL alone does not guarantee stable convergence. At the start of training, the RL agent expands its search space through random exploration. However, since the core power distribution changes substantially even with small CEA movements, there is no guarantee that such random exploration proceeds in the correct learning direction. AS a result, reward convergence is slow and, in the worst case, the policy may lose its learning direction and fail to converge.

Therefore, the action network is pre-trained before RL in order to restrict the search space in advance and provide a learning direction. This enables stable reward convergence, where convergence of the reward means that the generated scenario keeps |ASI| within the target range. In other words, the purpose of pre-training is to start not from a policy that constantly violates the ASI constraint with random CEA speeds, but from a policy whose |ASI| is already close to the constraint, so that it can be fine-tuned by RL.

### 3.2 Dataset Generation

Since pre-training is performed by SL, it requires a substantial amount of dataset. However, it is practically impossible to manually generate a dataset that covers all initial conditions and operating periods. This study therefore designs an algorithm that automatically generates scenarios satisfying the safety constraint in order to construct the dataset. Although this algorithm generates scenarios by random sampling, the narrow constraint yields only simple and similar path. Thus, RL is still required to explore diverse operating strategies for a given core state.

The dataset consists of 100-50-100 scenarios in which the power is lowered from 100% full power to 50% over two hours, held for eight hours, and returned to 100% over two hours. The data points are taken at one-hour intervals, and the target reactor is the APR1400.

The CEAs move only during power-transition periods, and their speeds are generated by random sampling. The generated speeds are converted to CEA positions, which are input to the core analysis code RAST-K to compute the ASI. The boron concentration corresponding to the target power is also obtained automatically. Thus, dataset generation consists of two steps: CEA pattern generation in section 3.2.1 and ASI screening in section 3.2.2.

#### 3.2.1 Algorithm-based Control Rod Pattern Generation

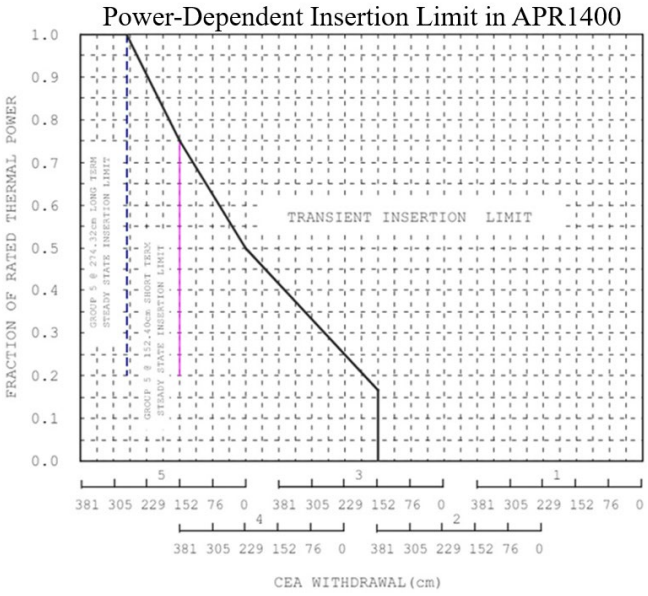
The proposed algorithm aims to generate CEA scenarios that operators can actually use. Since operating procedures specify that the CEAs be moved at a constant speed until the target power is reached [18], CEA insertion and withdrawal occur only during power-transition periods.

Accordingly, this study fixes the full-power period at all-rods-out (ARO) and the low-power holding period at the CEA state of that instant and generates CEA speeds only for the three power-transition periods: (1) 100% to 70%, (2) 70% to 50%, and (3) 50% to 70% state. The transition point is set at 70% since, following the recommended maximum ramp rate of 30% per hour, 70% is the lowest power reachable from full power within one hour [5].

The APR1400 CEAs consists of regulating group (R5-R1), the shutdown group (SA, SB), and the part-strength group (P1-P3), with R5 as the lead bank [19]. The part-strength group moves together and is therefore treated effectively as a single variable, P, whereas the regulating group has fixed spacing between follower banks, so that once R5 is determined, R4 is determined automatically. During DLFO, the shutdown group is kept at ARO. The CEAs that actually moves are R5, its follower bank R4, and P, and the independently controlled variables are R5 and P.

The position of the regulating group follows the Power-Dependent Insertion Limit (PDIL) rule. The PDIL prescribes the insertion limit as a function of power level, following the relationship shown in Fig 2 [20]. According to this relationship, in operation with a minimum power of 50%, the CEAs at R4 and below do not move owing to the insertion limit.

**Figure 3. PDIL of the regulating banks of the APR1400 as a function of power.**




The CEA speed is physically limited to two discrete values, 3 in/min or 30 in/min [19]. However, since the search unit in this study is hourly, a maximum of 1800 in/h covers the entire range of motion. The range of CEA motion is normalized with full insertion as 0 and full withdrawal as 1, and the relationship between the normalized and physical positions is given by Eq. 3.

$$z_{\mathrm{cm}} = 30 + 381 \cdot z_{\mathrm{norm}} \ [\mathrm{cm}] \tag{3}$$

The CEA speed is generated as a random value in the range 0 to 0.3, with insertion during power-decrease periods and withdrawal during power-increase periods. The generated speeds are accumulated, converted to CEA positions, and then corrected by the PDIL. A speed-limiting coefficient $c = 0.3$ is applied to the normalized speed since, without this constraint, the CEAs exhibit an unrealistic pattern of repeated full insertion and withdrawal. This constraint range will be extended in future work based on the learning results.

### 3.2.2 ASI Screening Algorithm

The converted CEA positions are input to RAST-K to compute the ASI, and the algorithm uses |ASI| = 0.2 as the threshold. If |ASI| < 0.2, is satisfied over the entire period, the scenario is stored in the dataset. If it is exceeded at any time step, the procedure returns to the CEA speed generation step. The threshold is set at 0.2 rather than the 0.3 safety limit for two reasons. First, plants are operated with a substantial ASI safety margin and rarely approach the 0.3 limit. Therefore, 0.2 serves as a conservative criterion that conforms to this operating practice while leaving a margin to 0.3. Second, it secures exploration headroom for RL. It is starting from a policy with |ASI| < 0.2, the |ASI| stays within a trainable range even if it rises to 0.3-0.4 during early exploration, whereas without pre-training the |ASI| easily exceeds 0.5, which this study uses as the episode termination condition. Using this algorithm, a total of 12,312 data points ($\mathbb{R}^{10\times12,312}$) satisfying |ASI| < 0.2 are obtained as shown in Table1 and used for pre-training.

**Table 1: State variables recorded per hour in the dataset**

| Variables | Description | Unit |
|---|---|---|
| POWER | Core Power | % |
| BURNUP | Mean Burnup | MWD/MTU |
| PPM | Soluble Boron Concentration | PPM |
| ASI | Axial Shape Index | |
| TF_MIN | Minimum Fuel Temperature | C° |
| TM_AVG | Average Moderator Temperature | C° |
| TM_IN | Moderator Inlet Temperature | C° |
| TM_OUT | Moderator Outlet Temperature | C° |
| R5 | R5 Regulating Bank Position | norm |
| P | P Bank Position | norm |

### 3.3. Input and Output of Agent Network Architecture

The encoder in actor network performs self-attention over the embedding state and the current core state to produce the current context, and the decoder in actor network generates the CEA speeds for the three sections through cross-attention between this context and the future power plant (the decoder query). The model input and output format is shown in Table 2.

### 3.4. Pre-training results

Pre-training is performed by SL based on the MSE loss, using the dataset of Section 2.2 and the actor network of Section 2.3. Starting from an arbitrary time point in the dataset, the CEA speeds generated by the model are compared with the actual speeds that satisfied |ASI| < 0.2, and the loss is computed as in Eq. 4.

$$\mathrm{MSE} = \frac{1}{N}\sum_{k=1}^{N}(\hat{a}_k - a_k)^2 \qquad (4)$$

Here, $\hat{a}_k$ is the predicted CEA speed, $a_k$ is the actual CEA speed, and $N$ is the number of data points. The optimizer is AdamW, and the detailed parameters are listed in Table 3. As shown in the Fig. 4, the validation loss decreases sharply from the beginning and converges to approximately 0.0229 indicating that the error between the predicted and actual CEA speeds is on the order of 3 to 5 cm. For scenarios starting from the full-power ARO initial condition, the pre-trained network generated CEA behavior satisfying |ASI| $\lesssim$ 0.2 over the entire period which is the maximum |ASI| is 0.216 as shown in Fig. 5. This achieves the goal stated in Section 3.1 of securing and initial policy close to the |ASI| constraint. In the RL stage, however, the search space is broadened, so |ASI| may exceed the criterion in some cases during early training.

## 4. REINFORCEMENT LEARNING

### 4.1. Reinforcement Learning Platform Coupled with RAST-K

RL consists of an agent and an environment, and the agent improves so as to obtain a higher reward through interaction with the environment as shown in Fig. 6. In this platform, the environment is the core analysis code RAST-K. The action generated by the agent is passed to RAST-K to produce a state, which is fed back to the agent to compute the next action, and this process is repeated. The reward, computed from the state produced by RAST-K, indicates to the agent whether the action was good or bad. The RL scenario maintains full-power ARO for 24 hours at BOC and then repeats the 100-50-100 DLFO for three days. The agent's action is the CEA for the three sections which is 100% to 70%, 70% to 50% and 50% to 70% which is converted to positions and input to RAST-K as a 24-hour scenario as shown in Fig. 7.

**Table 2: Input and Output variables of Network**

| | Variables | Description | Normalization |
|---|---|---|---|
| Encoder Input (LSTM Input) | POWER | Core Power | $P/100$ |
| | BURNUP_M | Min-Max Burnup | Min-max |
| | BURNUP_Z | Z-Score Burnup | Z-score |
| | PPM | Soluble Boron Concentration | Z-score |
| | ASI | Axial Shape Index | ASI/0.3 |
| | TF_MIN | Minimum Fuel Temperature | Z-score |
| | TM_AVG | Average Moderator Temperature | Z-score |
| | TM_IN | Moderator Inlet Temperature | Z-score |
| | TM_OUT | Moderator Outlet Temperature | Z-score |
| | R5 | R5 Regulating Bank Position | |
| | P | P Bank Position | |
| Decoder Query | START_POWER | Power at initial point | $P/100$ |
| | END_POWER | Power at final point | $P/100$ |
| | DELTA_POWER | Power difference with initial and final point | $P/100$ |
| | DIRECTION | Withdrawal (+1), Insertion (-1) | Boolean |
| | END_POWER_HOLD_TIME | The duration for the power at final point | |
| | START_BURNUP_M | Min-max normalized burnup at start point | Min-max |
| | START_BURNUP_Z | Z-scored scaled burnup at start point | Z-score |
| Encoder Output | CONTEXT | Context of the current state | |
| Decoder Output | ALPHA_SPEED | Alpha value for the beta distribution | |
| | BETA_SPEED | Beta value for the beta distribution | |

**Table 3: SL hyperparameters**

| Parameter | Value |
|---|---|
| Action Distribution | Beta distribution |
| Loss Function | MSE between beta mean and target speed unit |
| Optimizer | AdamW |
| Learning Rate | 3e-4 |
| Batch Size | 128 |
| Epochs | 200 |
| Train/Validation Split | 90/10 |
| Best Epoch | 184 |
| Best Validation MSE | 0.02286 |

**Figure 4. SL loss curve over 200 epochs.**

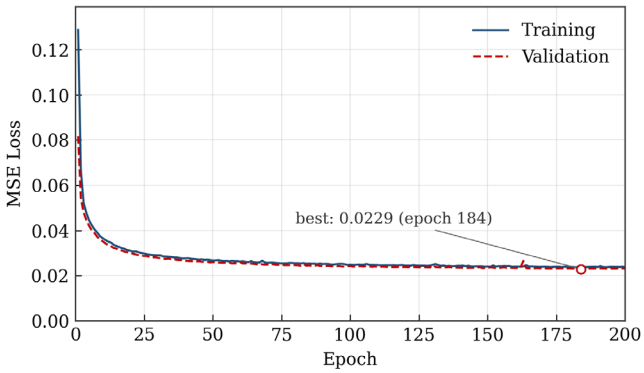



**Figure 5. Validation of the pre-trained actor on a DLFO scenario**

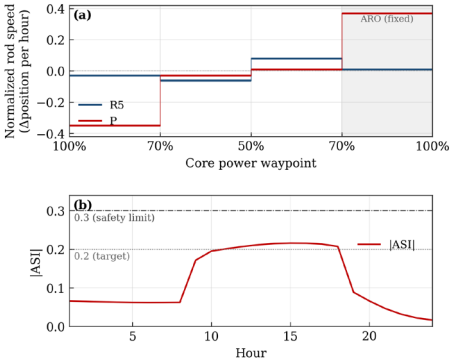



**Figure 6. RL platform coupled with RAST-K**

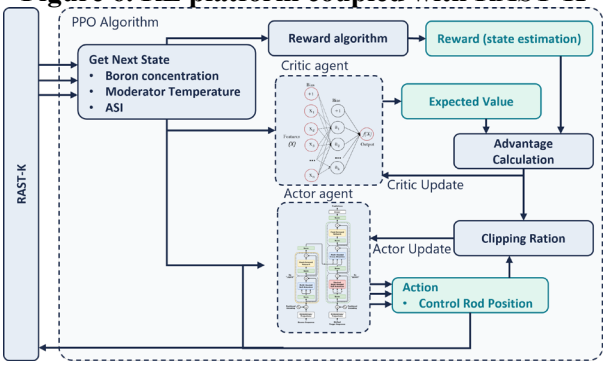
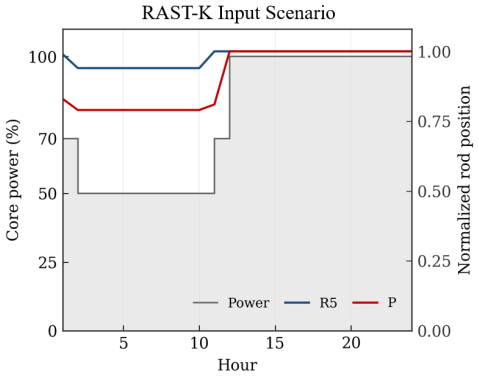



**Figure 7. DLFO Scenario from action**




## 4.2. Proximal Policy Optimization Design

PPO is an actor-critic RL method characterized by applying clipping to the loss computed from the reward, thereby limiting the size of each update. In ordinary policy-based RL without clipping an update can become excessively large and unstable, whereas PPO limits the magnitude of the policy change to induce cautious updates. Owing to this stability, PPO and Soft Actor-Critic (SAC) is used as a standard RL method, and this study also trains based on PPO. PPO constructs the loss from the probability of the action generated by the model and the advantage, as given in Eq. 5 [14].

Here, $r_t(\theta)$ is the probability ratio of the policy before and after the update, $\hat{A}_t$ is the advantage, and $\epsilon$ is the clipping range. The clip term prevents $r_t(\theta)$ from deviating beyond $1 \pm \epsilon$, thereby suppressing excessive updates.

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_l(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right],\ r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{5}$$

The advantage $\hat{A}_t$ indicates how much better a particular action is than the baseline value, and this study estimates it using Generalized Advantage Estimation (GAE). GAE defines the temporal-difference (TD) residual $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ from the value $V(s_t)$ estimated by the critic and exponentially weights it by $\gamma\lambda$ to obtain the advantage which is given in Eq. 6.

$$\hat{A}_t = \sum_{t=0}^{\infty}(\gamma\lambda)^l \delta_{t+l},\ \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \tag{6}$$

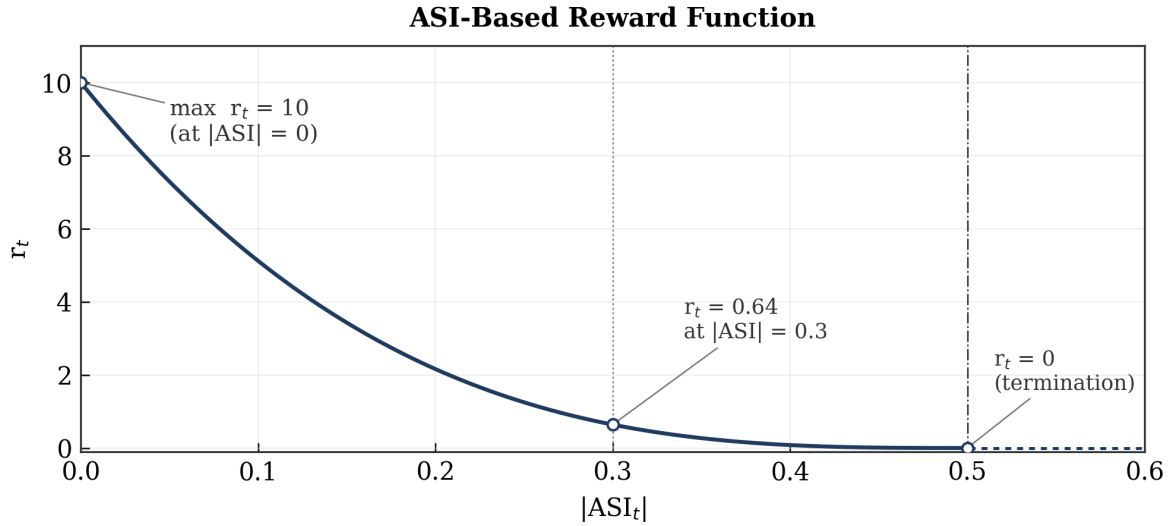
Here, $\gamma$ is the discount factor and $\lambda$ is a parameter that balances bias and variance, so that GAE interpolates between the two extremes to estimate the advantage stably [21].
All reward computations depend on the form of the reward function, which is described in the next section.

### 4.3. Reward Function for ASI

The reward function in this study depends on the ASI. If several quantities such as CEA position or boron concentration are mixed into the reward, the effects of the different terms can cancel out. For example, the reward for a case with good boron but poor CEA behaviour can equal that for a case with poor boron but good CEA behaviour, so the model cannot distinguish which variable to improve. Since the objective of this study is to generate operating scenario that satisfy the ASI safety constraint, the reward is restricted to the ASI so that the model focuses on improving it. The form of the reward function is given in Eq. 7 and Fig. 7.

$$r_t = 10\left(\max\left(0, \frac{0.5-|\mathrm{ASI_t}|}{0.5}\right)\right)^3 \tag{7}$$

**Figure 8. ASI-based reward function defined in Eq. 7**




The reward function is positive over its valid range (|ASI|<0.5) and increases sharply as |ASI| approaches zero. It is designed to be positive in order to avoid the early-termination problem. This platform uses |ASI| > 0.5 as the episode termination condition. If negative rewards were allowed, the agent could find it more advantageous, in terms of cumulative reward, to deliberately violate the constraint and end the episode early rather than to keep accumulating negative rewards [22]. In other words, learning would be distorted toward violating the safety constraint. Conversely, when the reward is always positive, more reward accumulates the longer the episode lasts, so the agent is induced to keep |ASI| within the constraint and continue operation. Based on this ASI-based reward function, the next section performs RL and analyses the results.

### 4.4. Training

The RL initialized by pre-training converges to an average reward of 463 at iteration 50, while the model trained without pre-training reaches 451 over the same horizon as shown in Fig 9a. Although final reward is comparable, two differences are observed.

First, pre-training raises the starting reward, at iteration 1 the average reward is 225 with pre-training and 121 without, indicating that the pre-trained policy starts from a more physically informed point, in line with the purpose stated in Section 2.1. Second, episode termination caused by |ASI| > 0.5 is suppressed throughout training. Over the first five iterations, the termination rate averages 10.0% with pre-training and 40.0% without. Over the full 50 iterations, the means are 1.0% a d 5.4%, respectively. By iteration 50, max |ASI| is 0.247 with pre-training and 0.316 without as shown in Fig 9b. the former remains within the 0.3 safety limit, whereas the latter still slightly exceeds it.

## 5. RESULTS

The trained agent generates the CEA speed pattern shown in Fig. 9a across the three controlled power-transition periods. The 70% to 100% return is fixed at ARO. The CEAs move at a constant speed within each transition, so the resulting scenario can be executed directly according to the operating procedures. The corresponding |ASI| trajectory in Fig. 9b remains within the 0.3 safety limit over the entire 96-hour |ASI| stays within the 0.2 conservative bound at every time step. The generated scenario therefore satisfies the ASI safety constraint, consistent with the objective of supporting safe DLFO operation.

As training progresses, however, the agent tends to reduce CCEA motion. When CEA movement decreases, the boron concentration computed by RAST-K to match the target power increases, since the reward depends only on |ASI| and does not constrain boron usage. This strategy still satisfies the ASI safety constraint but conflicts with the objective of minimizing boron usage.

**Figure 9. Training of the RL agent with and without pre-training**

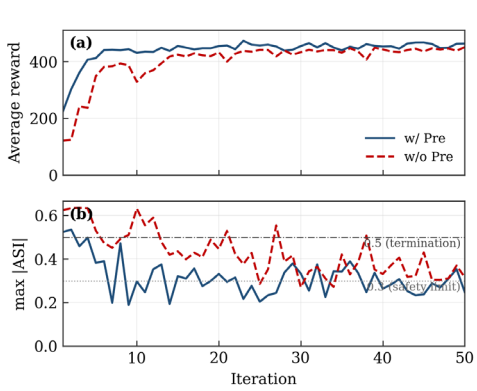
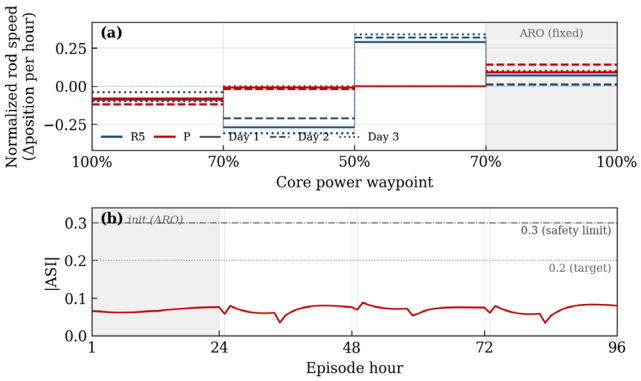



**Figure 10. Validation of the trained RL agent on an init plus three-day DLFO operation**




## 6. CONCLUSION & FUTURE WORK

This study presented a framework that, through supervised pre-training, constructs the actor network of PPO RL to generate 100-50-100 load-following CEA scenarios satisfying the |ASI| safety constraint. Pre-training mitigated the training instability caused by random exploration and enable stable convergence of RL, thereby automatically generating safety-constraint-satisfying scenarios in a form usable by operators.

However, since the reward function depends only on the ASI and does not constrain CEA behaviour or boron usage, the agent tended to move the CEA less and use more boron. Future work should add boron-minimization and CEA behaviour terms to the reward function to correct this tendency. In addition, since the EUR recommends performing DLFO five times per week, the scenario which is currently three days should be extended to one week or longer and generalized to include diverse initial conditions and burnup ranges.


### Acknowledgements

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean government (Ministry of Trade, Industry and Resources, MOTIR) (No. RS-2025-16063033).


### References


[1] International Energy Agency, "*Renewables 2025*", IEA, Paris, (2025).

[2] H. Kim, S. Choi, and B. Hwang, "*Analysis of challenges due to changes in net load curve in South Korea by integrating DERs*", Electronics, 9(8), 1310, (2020).
[3] A. Lokhov, "*Technical and Economic Aspects of Load Following with Nuclear Power Plants*", OECD Nuclear Energy Agency, Paris, (2011).
[4] European Utility Requirements, "*The European Utility Requirement (EUR) document, Volume 2 revision C*", (2001).
[5] A. Sowder, S. Bernhoft, and D. Moneghan, "*Expanding the Concept of Nuclear Flexibility for the Current Fleet and the Next Generation of Advanced Reactors*", EPRI, Palo Alto, CA, (2020).
[6] H. Khalefih, Y. Jeong, and Y. Kim, "*Daily Load-Follow Operation in LEU+-Loaded APR1400 Using Mode-K+ Control Logic*", International Journal of Energy Research, vol. 2023, 1853535, (2023)
[7] W. M. Stacey, "*Nuclear Reactor Physics*", 2nd ed., Wiley-VCH, Weinheim, (2007).
[8] J. Tian, H. Zheng, J. Liu et al., "*Axial offset control of PWR nuclear reactor core using intelligent techniques*", Annals of Nuclear Energy, 30(16), pp. 1645-1661, (2003).
[9] J. R. Lamarsh and A. J. Baratta, "*Introduction to Nuclear Engineering*", 3rd ed., Prentice Hall, Upper Saddle River, NJ, (2001).
[10] Y. -S. Bang et al., "*Design characteristics of nuclear steam supply system and passive safety system for Innovative Small Modular Reactor (i-SMR)*", Nuclear Engineering and Technology, (2025).
[11] S. -B. Koo, B. -J. Yoon, S. -W. Lee, J. H. Park, and H. C. Kim, "*Design of a model predictive load-following controller by discrete optimization of control rod speed for PWRs*", Annals of Nuclear Energy, 71, pp. 313-322, (2014).
[12] Y. Ahmed et al., "*A Study on Xenon Estimation During Load-Following Operation Using Sliding Mode Observer in APR1400 for a Predictive Soluble Boron Control*", Arabian Journal for Science and Engineering, (2024).
[13] R.S. Sutton and A. G. Barto, "*Reinforcement Learning: An Introduction*", 2nd ed., MIT Press, Cambridge, MA, (2018)
[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "*Attention is all you need*", Advances in Neural Information Processing Systems, vol. 30, (2017).
[15] T. P. Lillicarp, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "*Continuous control with deep reinforcement learning*", Proc. International Conference on Learning Representations (ICLR), (2016).
[16] J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "*Trust region policy optimization*", Proc. International Conference on Machine Learning (ICML), pp. 1889-1897, (2015).
[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "*Proximal policy optimization algorithms*", arXiv:1707.06347, (2017).
[18] M. Kromar et. al., "*An overview of power reactor kinetics and control in load-following operation modes*", Frontiers in Energy Research, vol. 11, 1111357, (2023).
[19] Korea Hydro & Nuclear Power Co., Ltd., "*APR1400 Design Control Document, Tier 2*", Revision 3, U.S. Nuclear Regulatory Commission Docket No. 52-046, (2018).
[20] I. Panciak and A. Diab, "*Dynamic Multiphysics Simulation of the Load-Following Behavior in a Typical Pressurized Water Reactor Power Plant*", Energies, 17(24), 6373, (2024).
[21] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "*High-dimensional continuous control using generalized advantage estimation*", Proc. International Conference on Learning Representations (ICLR), (2016).
[22] A. Y. Ng, D. Harada, and S. J. Russell, "*Policy invariance under reward transformations: theory and application to reward shaping*", Proc. International Conference on Machine Learning (ICML), pp. 278-287, (1999).