Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 7-8, 2026

Control-Rod Position Prediction for Load-Following Operation via a Reinforcement Learning Platform with a Pre-trained CVAE

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*

***Keywords :** load-following, neural network, artificial intelligence, VAE, reinforcement learning

1. Introduction

Load-following has become an essential operation for nuclear power plants (NPPs) as renewable energy increases in the power grid. Commercial nuclear units were mostly operated as baseload in the past since this method is economical and simple. However, the continuous growth of renewable energy creates a daily need to change power levels. Therefore, load-following which adjusts core power to match electricity demand, is considered a practical option to keep the power grid stable [1,2].

Safe load-following is challenging since a power maneuver drives transient states in the reactor core. Fuel temperature (T_{F}) and moderator temperature (T_{M}) change promptly with power, while Xe 135 and its precursor I 135 evolve more slowly, which can distort reactivity and the spatial power distribution during maneuvers [3]. This multi-time scale coupling makes it difficult to follow grid demand while keeping key operating parameters within allowable ranges [4]. In practice, load-following must satisfy multiple constraints simultaneously, including axial power-shape limits such as axial shape index (ASI) or axial offset (AO), control rod (CR) maneuver rules such as sequencing and overlap, power-dependent insertion limits, and conservative boron management [5].

Previous studies have demonstrated load-following control mainly through rule-based decision logics combined with core analysis calculations. One line of work performed full core simulations for daily load-following. It showed that CR maneuvers with soluble boron can satisfy major operational limits when the CR sequencing and overlap rules are followed and key constraints are monitored [6]. Another study proposed a long-term load-follow strategy based on a preset outlet temperature band. It triggers CR motion when the deviation exceeds the band and refines the logic near axial shape limits by considering axial offset (AO) and boron related parameters [5]. Intelligent techniques have also been investigated for axial power shape regulation, where data-driven core models and supervisory decision modules guide CR maneuvers for AO control [7].

Despite these efforts, existing approaches still show limitations in learning-based optimization under strict control constraints. Rule-based methods require careful tuning of reference bands and trigger conditions. They are designed to keep the NPP within allowable limits, however, they do not directly optimize boron usage and CR motion. This motivates a learning framework that can explore control actions while remaining within feasible CR maneuver patterns from the beginning of training. In this paper, we propose a reinforcement learning (RL) platform coupled with RAST-K. We employ a pre-trained conditional variational autoencoder (CVAE) to generate physically valid CR patterns and ensure stable training progression.

2. Methodology

In this section, we describe the proposed RL platform for load-following control and its key components. The overall workflow consists of (i) the RL formulation and the agent environment interaction (ii) the pre-trained CVAE policy model including input normalization and the training objective with β-warmup (iii) the generation of feasible CR trajectories under maneuver rules.

2.1 Reinforcement learning

We adopt reinforcement learning since load-following control can be formulated as a constrained long-horizon sequential decision-making problem. At each step, the controller must select a CR action that keeps the core response within operational limits while pursuing additional objectives. By contrast, supervised learning typically requires labeled actions and can be ill posed when multiple admissible actions exist for the same state. RL avoids this limitation by learning from interaction outcomes rather than relying on a unique target action.

In RL, the problem is formulated as an interaction between an agent and an environment. In this study, the environment is implemented using the core analysis code RAST-K, and the agent is implemented as a pre-trained CVAE-based policy that generates CR patterns. Without pre-training, early exploration produces near-random CR actions that violate maneuver constraints and terminate episodes after only a few steps, which is the cold start problem. The pre-trained CVAE mitigates this issue by providing a structured action representation learned from physically feasible CR patterns, which restricts exploration to plausible maneuvers from the beginning of training.

We adopt a stochastic policy, which simplifies platform design and is well suited to continuous CR actions. In this work, we focus on demonstrating training progression with the proposed platform. A detailed comparison with proximal policy optimization (PPO) and other advanced on-policy algorithms is left for future work.

We adopt a VAE structure model since its latent space is explicitly regularized toward a prior distribution, which makes latent sampling straightforward and suitable for a stochastic policy representation. We further employ a CVAE since reconstructing CR maneuver patterns from the CR sequence alone did not provide sufficient information for accurate reconstruction in our preliminary tests. By conditioning the decoder on additional state related inputs, the CVAE improves reconstruction fidelity while preserving a sampleable latent distribution for policy exploration.

2.2 CVAE

We describe methodology for the CVAE training procedure, including feature normalization and the loss function used for pre-training. Before training, input features are normalized to ensure consistent scaling across variables. Min-max scaling depends on the selected minimum and maximum values and there is no unique standard for choosing them. When operational bounds are used for min-max scaling, the effective variation of some features can be overly compressed, and the normalized deviation may converges to zero. In contrast, z-score standardization transforms each feature to have mean zero and unit standard deviation and is generally more robust to changes in data distribution. Therefore, we adopt z-score standardization for normalization.

The CVAE is trained using the standard VAE objective. Let x denote the target sequence to be reconstructed and let z denote the latent variable. The encoder $q_\phi(z|x)$ with parameters ϕ approximates the posterior distribution of z conditioned on x. The decoder $p_\theta(x|z)$ with parameters θ defines the conditional likelihood of reconstructing x from z. The reconstruction loss $\mathcal{L}_{\text{recon}}$ in Eq. (1) is the negative expected log likelihood under the encoder distribution and it encourages accurate reconstruction of the input sequence.

$$\mathcal{L}_{\text{recon}} = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \cdots\cdots\cdots \quad (1)$$

The Kullback-Leibler divergence loss $\mathcal{L}_{\text{KL}}$ in Eq. (2) regularizes the encoder output toward the prior distribution $p(z)$. Here $p(z)$ is typically chosen as a standard normal distribution. The operator $D_{\text{KL}}(\cdot\,||\,\cdot)$ measure the divergence between two distributions. Minimizing $\mathcal{L}_{\text{KL}}$ makes the latent space smooth and sampleable, which is required for stochastic policy generation.

$$\mathcal{L}_{\text{KL}} = D_{\text{KL}}\left(q_\phi(z|x)\,||\,p(z)\right) \cdots\cdots\cdots\cdots\cdots \quad (2)$$

The total loss $\mathcal{L}_{\text{tot}}$ in Eq. (3) is a weighted sum of the reconstruction and KL terms. The weight $\beta(t)$ controls the relative strength of the KL regularization at training step or epoch t

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{recon}} + \beta(t)\mathcal{L}_{\text{KL}} \cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (3)$$

A strong KL term in early epochs can hinder reconstruction and destabilize training. Therefore, we apply a β-warmup schedule. During the initial training stage, β is set to zero to focus on reconstructing physically reasonable CR maneuver patterns from the dataset. After epoch 50, β is gradually increases to regularize the latent space toward a meaningful distribution.

2.3 Data generation for feasible CR pattern

CR operations follow predefined maneuver rules, including bank sequencing and overlap. During insertion, the regulating banks are moved sequentially from R5 to R1, and the withdrawal order is reversed. Overlap is initiated when the lead bank reaches a preset insertion depth, for example 30 cm, after which the adjacent bank starts moving together with the lead bank. During withdrawal, the same overlap rule is applied with the bank order reversed [6].

To cover a wide range of maneuver patterns, we generate candidate CR trajectories by random sampling conditioned on the target core power profile. Some sampled trajectories can violate the maneuver rules, therefore, they are corrected to satisfy the sequencing and overlap constraints to obtain feasible CR patterns.

Each feasible rod trajectory is provided to RAST-K as an input, and the corresponding core state variables are computed. The RAST-K outputs are stored together with the input CR position sequences to construct the dataset. Since the data are time sequential, fixed length windows with $T = 16$ are used for sequence modeling, yielding a total of 14,197 samples.

3. Design of RL platform

In this section, presents the design of the proposed RL platform for load-following control. The platform design consists of (i) the agent design based on a pre-trained CVAE stochastic policy (ii) the reward function design that enforces the ASI constraint while encouraging boron reduction (iii) the environment design implemented using the RAST-K core analysis code.

3.1 Design of agent

In our framework, the agent represents a stochastic policy $\pi_\theta(a_t|h_t)$ using a pre-trained CVAE, shown in Fig. 1. The CVAE employs an LSTM-CNN encoder and

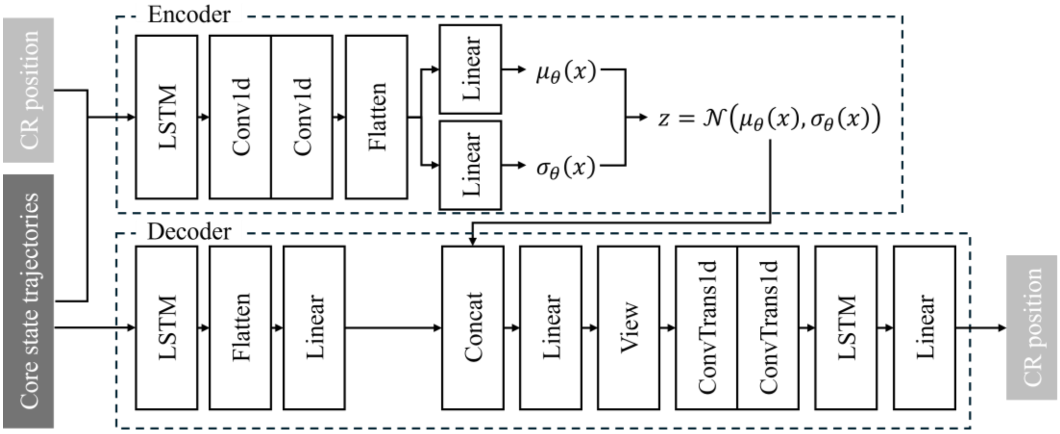


Fig. 1. CVAE architecture used for CR pattern generation.

decoder. The LSTM layers capture temporal dependencies in the CR trajectories, while the convolutional layer extract local pattern features along the sequence.

Each decision step t corresponds to one hour of operation. At step t, the agent observes a history tensor $h_t \in \mathbb{R}^{T\times F}$ with $F = S + C$, where $S = 23$ denotes the number of core state features and $C = 7$ denotes the number of CR position features. Conditioned on h_t, the encoder outputs a latent representation and a latent vector z_t is drawn to introduce stochastic exploration. The decoder then maps z_t to the next CR action a_t, which is defined as the next step CR positions. The agent environment input and output interface is summarized in Table I.

Table I: Inputs and outputs of the RL platform components

	Agent	Environment
Input	$h_t \in \mathbb{R}^{T\times F}$ $T = 16, F = S + C$	CR positions (a_t), T_F, T_M, power
Output	$a_t \in \mathbb{R}^C \rightarrow$ next CR positions	Burnup cycle, soluble boron concentration (ρ_B), ASI, k_{eff}, reactivity, T_F, T_M, DNBR margin, other thermal hydraulic variables

To preserve feasibility during RL training, we freeze the decoder during RL training and update only the encoder parameters. The fixed decoder acts as a constrained action generator that produces CR actions consistent with the feasible pattern manifold learned during pre-training. Therefore, RL updates focus on the encoder so that it selects latent variables that yield load-following maneuvers while maintaining physical feasibility.

We also refine the decoder design to avoid KL collapse during CVAE training. When the decoder directly reuses the input sequence as a strong reference signal, reconstruction can rely on the re-input path and the KL divergence can collapse toward zero. To mitigate this issue, we introduce an input compression module that reduces the information strength of the re-input signal and encourages the decoder to utilize the latent variables.

3.2 Design of reward function

The reward function is a key component of the proposed RL platform. we construct the reward using two environment outputs, the ASI and the soluble-boron concentration (ρ_B). We keep the reward design simple to avoid competing terms that can reduce the effective learning signal.

The safety objective is represented by the ASI term r_{ASI} in Eq. (4). Here r_{ASI} assigns a strong penalty when the ASI limit is violated. An episode is terminated when |ASI| > 0.3, and a large weight w_{ASI} is used in Eq. (6) to emphasize strict compliance. In Eq. (4), sign(·) is the sign function. When |ASI| $\leq$ 0.3, the term becomes $r_{\text{ASI}} = 0$. When |ASI| > 0.3, the term becomes $r_{\text{ASI}} = -2$, which yield a steep penalty near the threshold.

We also include a boron term r_B in Eq. (5) to discourage high boron concentration. ρ_{target} denotes a reference ρ_B used for normalization. The total reward is computed as a weighted sum of the two components in Eq. (6), where w_B controls the relative importance of boron management.

$$r_{\text{ASI}} = \text{sign}(0.3 - |\text{ASI}|) - 1 \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (4)$$

$$r_B = \left(\frac{\rho_B - \rho_{\text{target}}}{\rho_{\text{target}}}\right)^3 \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (5)$$

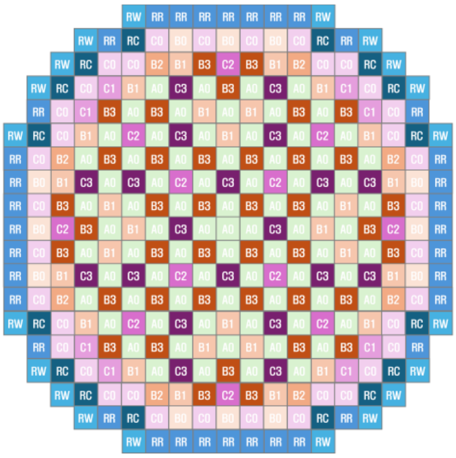
$$r_{\text{total}} = w_{\text{ASI}} r_{\text{ASI}} + w_B r_B \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (6)$$

When the ASI limit is violated, the episode is terminated and a large terminal penalty of -1000 is assigned.

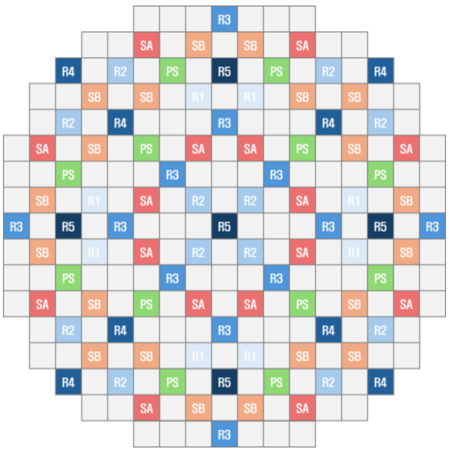
3.3 Design of environment

We use the core analysis code RAST-K as the environment in the proposed RL platform. RAST-K is a nodal diffusion code for core neutronics analysis and allows flexible configuration for simulation studies. In this work, we adopt the APR1400 core configuration shown in Fig. 2. The control rods are grouped into five regulating banks (R1-R5), one part-strength bank (PS), and two shutdown banks (SA and SB). RAST-K requires multigroup cross-section data, which are generated using STREAM 2D, a reactor analysis code developed at UNIST.

RAST-K computes core state variables including ASI and the ρ_B, and it also provides temperature related quantities such as T_F and T_M used in the calculations, as summarized in Table I. Since these outputs are used to construct the next state in the RL loop, the environment can be executed iteratively at each decision step.



(a) Core batch pattern



(b) CR bank pattern

Fig 2. APR1400 core and CR bank configuration.

4. Implementation

In this section, we describe the practical implementation of the proposed platform, including the CVAE pre-training setup and its integration into the RL loop.

The CVAE is trained on the time window dataset described in Section 2.3. The model is trained for 250 epochs with a learning rate of 3×10^{-3}. The batch size is 128 and the latent dimension is 128. Z score standardization is applied to input features prior to training. Model performance is evaluated using reconstruction-based metrics, including the KL divergence term and the mean absolute error reported in Table II. The resulting MAE is 0.0162, which corresponds to an average control rod position deviation of approximately 6 cm in the physical scale. Training convergence is monitored using the KL divergence trajectory under the beta warmup schedule. As shown in Fig. 3, the KL divergence decreases rapidly after epoch 50 as the β term increases and converges to 3.9305.

After pre-training, the CVAE is used to initialize the RL agent. During RL training, the decoder is kept fixed and only the encoder parameters are updated to select latent samples that generate feasible control rod actions. The RAST-K environment is executed iteratively for each decision step, and the reward and termination conditions described in Section 3.2 are applied to collect trajectories for policy updates.

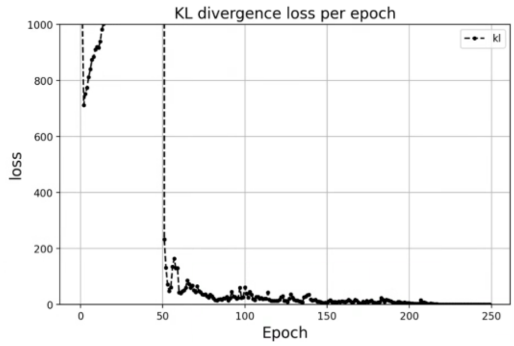


Fig. 3. KL divergence during CVAE training with β-warmup. The KL divergence decreases rapidly after epoch 50 as the β term is increased.

Table II: CVAE performance metrics.

	CVAE model
$\mathcal{L}_{KL}$	3.9305
MAE	0.0162

5. Results

As a result, Fig. 4 presents a representative load-following episode under a 100-50-100 percent power scenario with a target ramp rate of 25 percent power per hour. The trends of ρ_B and CR position change in response to the power maneuver, indicating that the agent learns to generate coordinated control actions that follow the load-following trajectory. Fig. 4 also shows that episode survivability increases as training progresses, which suggests that the proposed platform enables learning without early termination caused by infeasible control rod actions. This behavior indicates that the pre-trained CVAE initialization mitigates the cold start problem and constrains exploration to feasible rod patterns under the maneuver rules.

Although a boron reduction term is included in the reward function, the boron usage tends to increase as the episode progresses in later stages. This result suggests that the current policy still relies on boron adjustment to achieve the maneuver rather than shifting the control effort toward control rod actions. Further reward shaping and additional penalties on boron usage and control rod movement will be required to promote balanced multi objective control.

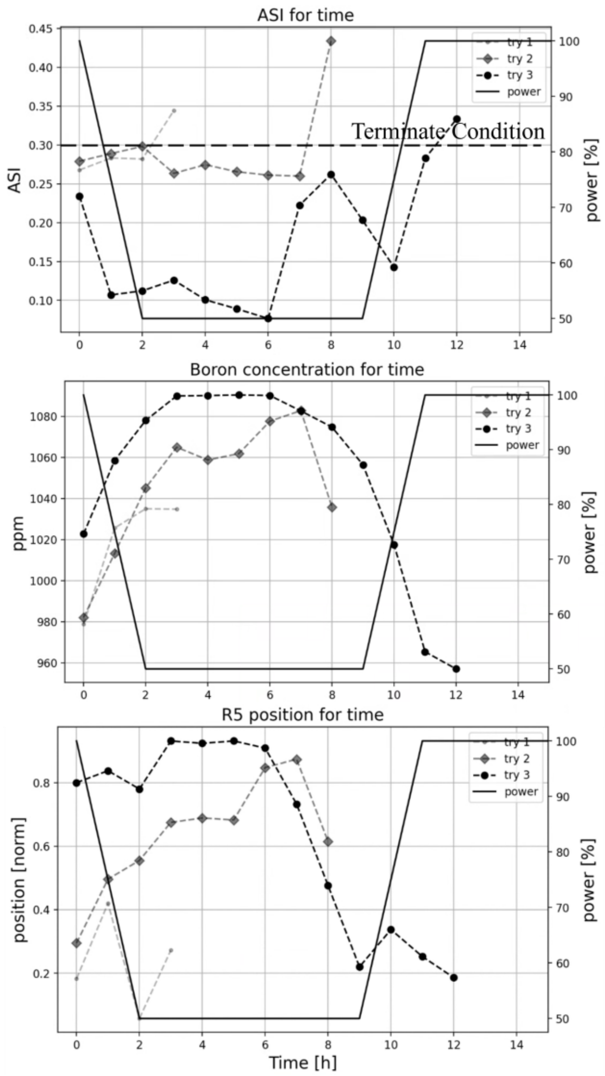


Fig. 4. Representative RL maneuver trajectories under the ASI termination constraint. The time histories of ASI, boron concentration, and the R5 bank position are shown for a representative episode.

4. Conclusion and Future Works

We developed a CVAE to generate CR patterns that satisfy maneuver constraints for load-following operation. To train and evaluate the proposed approach, we implemented an RL platform coupled with the core analysis code RAST-K. The pre-trained CVAE was used to produce feasible Cr patterns, which allowed the learning loop to proceed without early termination.

Future work will refine the reward formulation to induce more meaningful CR movements instead of relying on boron adjustment. In particular, we will adjust the boron reward weight w_B , incorporate power dependent insertion limits (PDIL), and include explicit penalties to minimize CR motions. We will also investigate improved policy optimization methods, such as PPO, to enhance training stability and performance.

Acknowledgement

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] C. Saigustia and P. Pijarski, Time Series Analysis and Forecasting of Solar Generation in Spain Using eXtreme Gradient Boosting: A Machine Learning Approach, Energies, Vol. 16, No. 22, Article 7618, 2023.
[2] E. Graham, N. Fulghum, and K. Altieri, Global Electricity Review 2025, Ember, April 8, 2025.
[3] G. Žerovnik, D. Čalič, S. Gerkšič, M. Kromar, J. Malec, A. Mihelčič, A. Trkov, and L. Snoj, An overview of power reactor kinetics and control in load-following operation modes, Frontiers in Energy Research, Vol. 11, 1111357, 2023.
[4] G. Chen, M. Li, Y. Zou, and H. Xu, Analysis of load-following operation characteristics of liquid fuel molten salt reactor, Progress in Nuclear Energy, Vol. 150, 104308, 2022.
[5] K. Park, T. Park, S. K. Zee, and B. S. Koo, Control Rod Driven Long-Term Load-Follow Operation in Small Modular Reactor, Annals of Nuclear Energy, Vol. 182, Article 109620, 2023.
[6] M. Boroushaki, M. B. Ghofrani, C. Lucas, M. J. Yazdanpanah, and N. Sadati, Axial Offset Control of PWR Nuclear Reactor Core Using Intelligent Techniques, Nuclear Engineering and Design, Vol. 227, No. 3, pp. 285–300, 2004.
[7] M. Park, J. Choe, D. Lee, and E. Lee, Application of RAST-K to Simulation of OPR1000 Daily Load Follow Operation, Transactions of the Korean Nuclear Society Autumn Meeting, Pyeongchang, Korea, October 30–31, 2014.