

KNS Spring Meeting | 2026.05.07

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

---

방준형, 김종현\*

원자력 계측제어 및 자율운전 연구실 (NICA Lab.)

한국과학기술원, 원자력 및 양자공학과



01

## Introduction

Background and motivation

---

02

## Pre-training

For the feasibility of control rod position

---

03

## Reinforcement Learning

For fine-tuning of pre-trained model

---

04

## Results

Interpretation and comparison

---

05

## Conclusion & Future Work

Summary and next steps

---

# 01

## Introduction

---

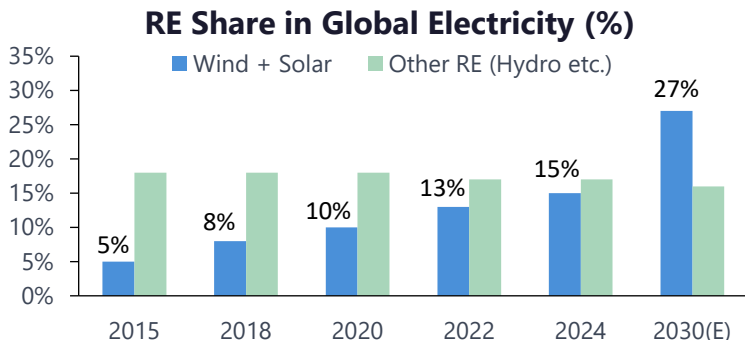
Background, motivation, and literature survey

## 재생에너지 확대에 의한 원전의 탄력운전 요구

재생에너지(풍력·태양광) 비중 확대에 의한 전력계통 수급 불균형 심화. 이로 인해 기존 기저부하 운전이었던 원전의 탄력적인 출력 변동이 요구됨.

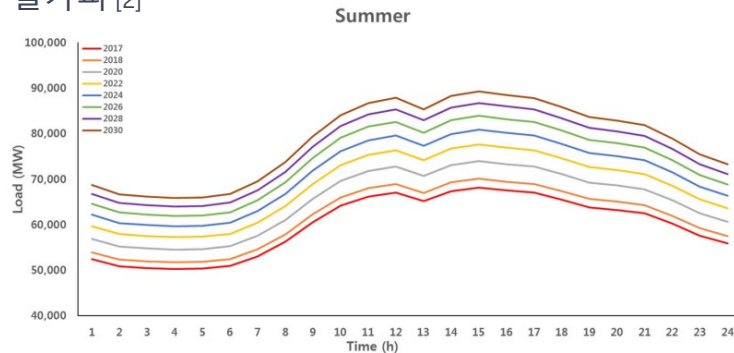
### 1 VRE 비중 급증

2030년 풍력+태양광 비중 27% 전망, 계통 변동성 심화 [1]



### 2 국내 전력망 부하 사례

여름철 피크-저점 부하 갭 연도별 확대, 원전 출력 37~93% 범위 운전 불가피 [2]



기저부하 운전에서 탄력적인 출력 변동을 하는 탄력 운전으로의 전환이 요구됨

## 탄력운전에 대한 설명

- 탄력운전은 외부 전력망의 수요공급에 따라 원자력 발전소의 출력을 그에 맞게 조절하는 운전임
- 출력 증감발을 하기 위해 **제어봉** 및 **붕산농도 조절**로 노심의 반응도 조절
- EUR과 EPRI에서는 노심 출력을 **100%-50%-100%**을 하루주기로 바꾸는 **일일부하추종**에 대해 가이드라인을 제공함
- 그러나 노심은 비선형적이고 복합적인 물리적 상호작용을 하므로 **노심의 반응도를 조절하는 일은 쉽지 않음**

European Utility Requirements (EUR) 권고사항

- 100%-50%-100%
- 일일부하추종
- 1주에 5번
- 1년에 200번

EPRI 권고사항

- 2시간안에 100%에서 저출력 전환
- 저출력을 2시간에서 10시간 유지
- 2시간안에 저출력에서 100% 전환
- 남은 하루동안 100% 출력 유지

	Mode A	Mode G	Mode X
Started	1970	End of 1970	1980
Variables	Boron Concentration	Mode A + Control Rod	Mode G + Axial Power Offset
Limitations	Very Slow Reaction	Core Power Imbalance	

프랑스 부하추종 사례

## 국내 탄력운전 현황

- 국내 신재생 에너지의 비중 증가로 Grid 변동성이 폭증함
- 기존 원전의 base-load만으로는 Grid에 대응 불가
- i-SMR의 기술 개발로 탄력운전에 대한 관심이 커지고 있음

### i-SMR에 의한 탄력운전 수요 증가

- i-SMR의 8대 혁신에 자율운전 + 재생에너지 연계가 명시
- 재생에너지와의 연계를 위해서는 탄력운전이 불가피함

### 대형 원전의 탄력운전 수요 증가

- Base-load 기반인 APR 계열 노형 또한 탄력운전이 요구됨
- 신한울 1호기에서 100%-80%-100%의 탄력운전 테스트 진행
- 국내 대형 원전 또한 탄력운전을 하고자 하는 시도가 계속되며 이에 대한 과제가 활발히 진행

## Project Information

**Title:** 실시간 3차원 노심출력분포 계산 및 운전지원 통합시스템 개발

**Period:** 2025.06 – 2028.12

**Target:** APR형 원전 탄력운전 실증 현장 적용 (신한울 1호기)

### 본 과제 목표

- |                                    |            |
|------------------------------------|------------|
| • 100-50-100 혹은 100-20-100의 일일부하추종 | } EUR 권고사항 |
| • 1주에 5번, 1년에 200번 진행              |            |
| • 시간당 최대 30% 출력 증감율                | EPRI 권고사항  |
| •  ASI <0.3 미만을 만족하는 운전 시나리오 생성    |            |

## 탄력운전시 노심관련 문제점

### Problem

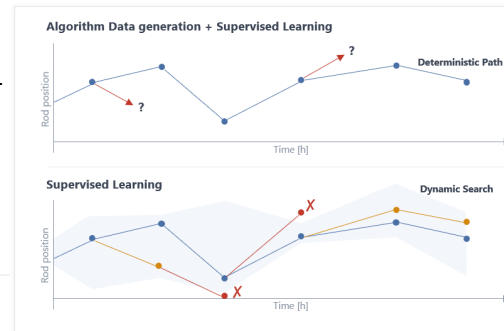
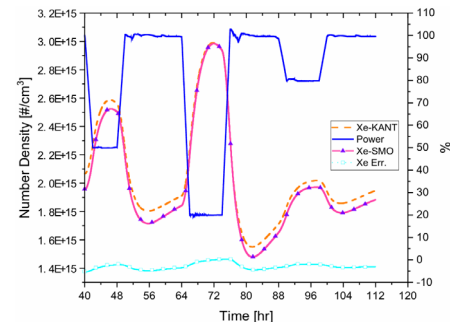
- 노심 제논 축적으로 인해 동일 노심상태라도 **향후 운전 전략이 크게 변함**
- 운전 전략에는 정답루트가 없으며 요구되는 운전 전략 또한 달라 **다양한 시나리오를 요구함**

### 제논 축적 문제

- 원자력발전소의 노심은 운전 전략에 따라 노심에 축적되는 제논의 분포가 달라짐
- 현재 노심의 제논 분포에 따라 향후 수행해야 할 운전 전략은 크게 변함
- 그러나 현재 노심의 제논 분포는 측정할 수 없음
- 그러므로 과거 운전 이력을 고려하여 현재 노심 상태를 추정하고 운전 전략을 세워야 함

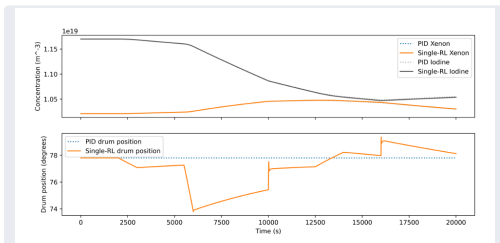
### 시나리오 다양성 문제

- 같은 노심 상태라도 운전 전략을 어떻게 세우느냐에 따라 향후 노심 상태는 큰 폭으로 변함
- 노심은 비선형성을 따르므로 현재의 운전 전략이 미래 어떤 영향을 미칠지 예측이 어려움
- 그러므로 현재 노심 상태에 대해 결정론적인 단일 운전 전략 제공 보다 미래를 고려한 다양한 운전 전략 경우를 모두 고려해야 함



## 과거 연구 사례

[Tunkle, L et al., 2025] [4]



### Motivation

- 마이크로리액터의 자율제어를 통해 운영비 절감
- 탄력운전 시나리오에서 강화학습으로 PID/MPC를 대체할 수 있는지 증명.

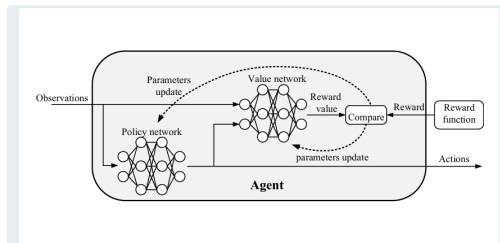
### Method

- 마이크로리액터의 8개 control drum을 PPO + multi agent (MARL)로 제어.
- PID baseline과 single 및 multi-drum 시나리오에서 비교, noise 견고성 평가.

### Contribution

- 마이크로리액터에 real-time RL 처음 적용
- MARL로 single-agent가 못하던 multi-drum달성
- Gaussian noise하에 PID보다 낮은 error

[Chen, J et al., 2025] [5]



### Motivation

- 복잡한 노물리로 인해 전통 PID제어로 한계가 있음
- 변수간 coupling effect를 다루기 위한 다변수 통합 제어 방법 필요.

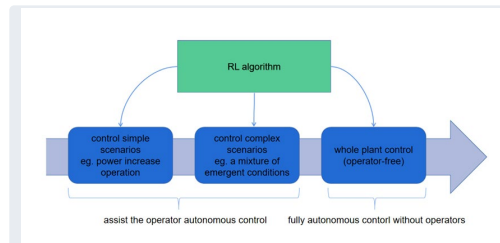
### Method

- Small PWR의 Nuclear Steam Supply System (NSSS)에 Deep Deterministic Policy Gradient (DDPG) 적용.
- 단일 controller가 다변수 동시 제어.

### Contribution

- NSSS 다변수 통합 제어를 DDPG로 디자인
- 모든 평가 지표에서 PID 대비 15.5%개선
- RL기반 다변수 노심 제어의 이론적틀 제시

[Gong, A et al., 2024] [6]



### Motivation

- 다른 Complex system에서 완성된 강화학습을 원자력 발전소에 적용할 수 있을지 리뷰.

### Method

- Mainstream RL 알고리즘,
- RL이 적용된 다른 complex system, 원자력 발전소 task별 강화학습 적용 사례 정리
- 이후 향후 적용 방향과 challenges 논의.

### Contribution

- RL mainstream framework의 advantage, limitation 정리
- Complex system과 NPP간 RL 적용 가능성 맵핑
- 원자력발전소 Task별 RL 적용 가능성 종합

[4] Tunkle, L., Abdurraheem, K., Lin, L., & Radaideh, M. I. (2025). Nuclear microreactor transient and load-following control with deep reinforcement learning. *Energy Conversion and Management: X*, 25, 101090.

[5] Chen, J., et al. (2025). A Multi-Variable Coupled Control Strategy Based on a Deep Deterministic Policy Gradient Reinforcement Learning Algorithm for a Small Pressurized Water Reactor. *Energies*, 18(6), 1517

[6] Gong et al. (2024). Possibilities of reinforcement learning for nuclear power plants: Evidence on current applications and beyond. *Nuclear Engineering and Technology*, 56, 1959–1974.

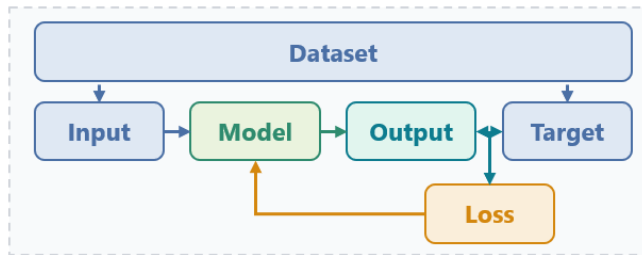
## 본 발표의 개요

### Objective

- Pre-trained Network 기반 강화학습을 적용한 탄력운전 시나리오 생성

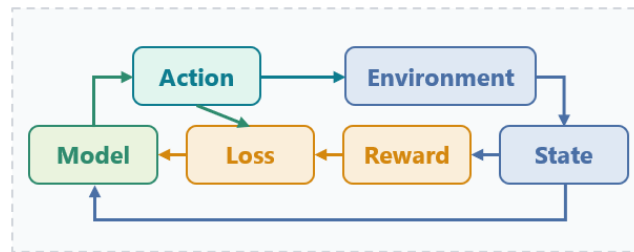
### Step 1. 사전학습 (Pre-training)

- 사전학습의 목적
- 사전학습의 방법
- 지도학습을 하기 위한 데이터셋 생산
- 데이터셋 생산을 위한 데이터셋 생산 알고리즘 설계



### Step 2. 강화학습 (Reinforcement Learning)

- 강화학습의 목적
- 강화학습 방법
- PPO 알고리즘을 기반으로 한 강화학습 플랫폼 설계 (노심해석코드인 RAST-K 연동)
- LSTM + Transformer 기반 Actor network, MLP 기반 Critic network 설계
- ASI기준 보상함수 설계
- 학습 진행 및 결과



# 02

## Pre-training

---

For the feasibility of the control rod position

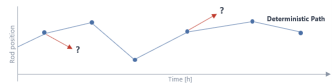
## Pre-trained의 목적 - 강화학습에 활용되는 Network 모델 개발

### 데이터셋 생산

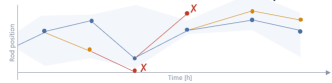
#### 2.1 사전학습의 필요성

- 강화학습 시 탐색이 랜덤으로 진행될 수 있음
- 랜덤 탐색은 학습 시간이 오래걸리고 불안정하며 학습 실패 가능성이 있음

Algorithm Data generation + Supervised Learning



Supervised Learning



Cold-Start Problem

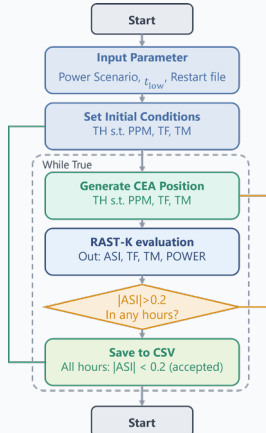


Warm-Start



#### 2.2 데이터셋 생산

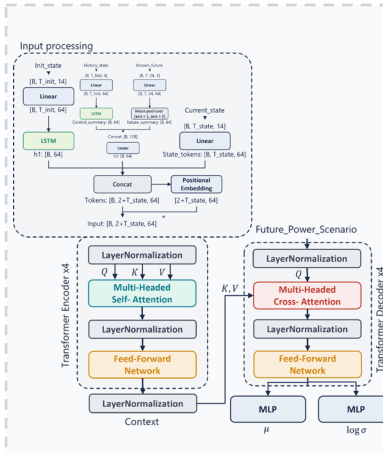
- 제어용 운전 제약 조건을 도입하여 제어용 움직임을 제약
- BOC부터 EOC까지 전주기에 이르는 1.5년치 1시간단위의 데이터셋 생산



### 사전학습 진행

#### 2.3 모델 구축

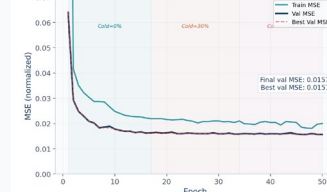
- 강화학습의 Actor Network가 될 모델로 Transformer 기반 모델을 구성
- LSTM을 통해 과거 운전이력 반영



#### 2.4 모델 학습

- MSE Loss, AdamW Optimizer 기반 지도학습
- Val MSE 0.015부근 수렴, 사전학습으로서의 목표 달성

Training convergence



LSTM weight change - TBPTT verification



### 2.1: 사전학습의 필요성

#### 랜덤 탐색 문제



- 초기정책의 무작위성으로 무의미한 탐색이 반복됨
- 제약 위반이 빈번하게 발생하여 제대로 된 탐색 불가

→ 초기 학습 시간 폭증 및 탐색 자체가 불가능해질 수 있음

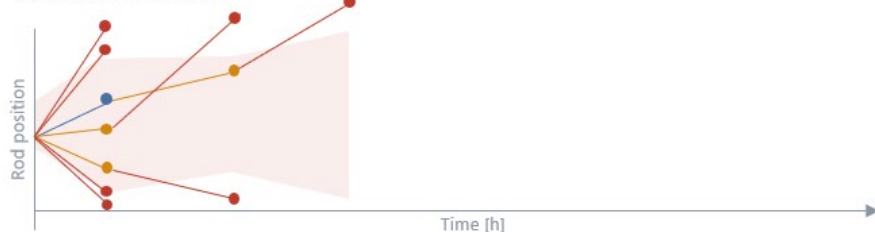
#### 사전학습을 통한 규제



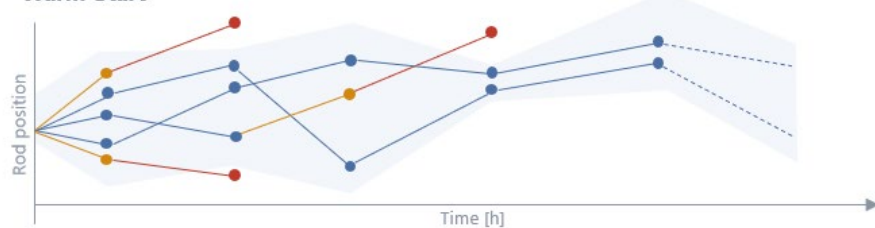
- Actor network를 pre-train하여 학습공간 규제

→ 탐색 공간을 축소시켜 학습 방향성 제공

#### Cold-Start Problem



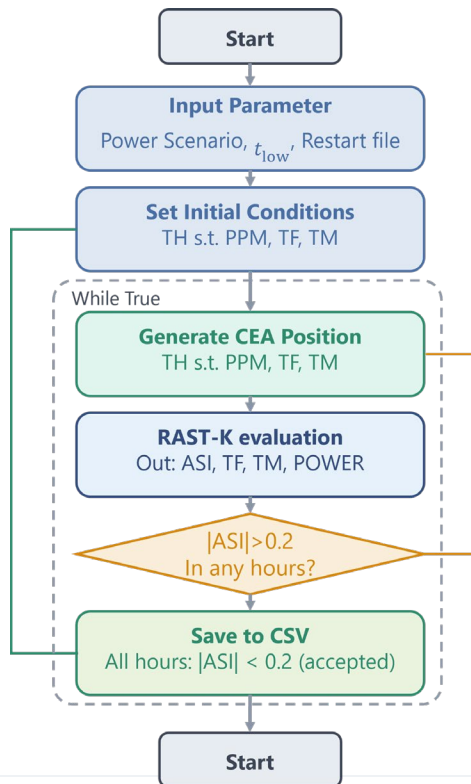
#### Warm-Start



#### → 유의미한 운전 시나리오만 생산하도록 탐색 공간 규제

- 운전 시나리오 (제어봉)이 랜덤하게 출력되어 물리적 규칙을 위반하면 시나리오 조기 종료로 학습 실패 확률이 높아짐

## 2.2: 데이터 생산 - 생산 알고리즘 설계



- 0 Step 0: Input**
- 이전 일차 종료 시점의 노심상태(붕소 농도, 연료온도, 감속재온도)를 로드
  - 첫째 날은 정상 전출력(ARO) 조건에서 시작

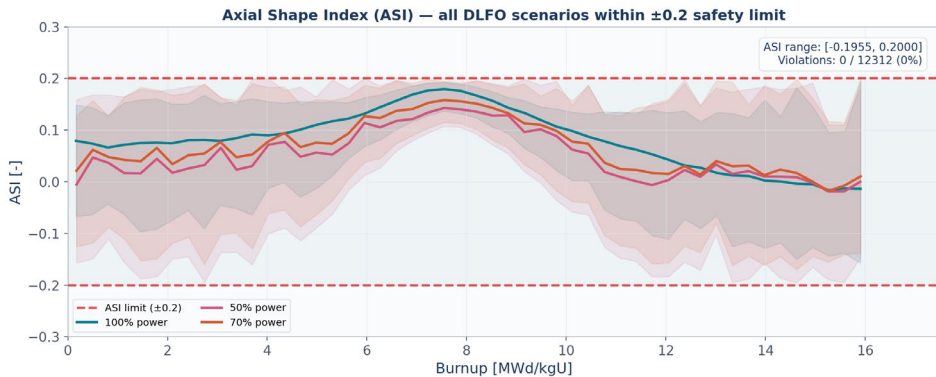
- 1 Step 1: Control Element Assembly (CEA) Position Generation**
- 제약조건을 만족하는 제어봉을 허용 범위내에서 확률적으로 생성
    1. Power Dependent Insertion Limit (PDIL)
    2. Bank Order (P>R5>R4>R3>...)

- 2 Step 2: RAST-K simulation**
- 생성된 제어봉 패턴을 노심해석코드인 RAST-K 3D quasi-static 모듈에 입력
  - 24시간 시나리오를 모사.
  - 매 시간 단위로 ASI, 출력분포, 연료 및 감속재 온도를 산출

- 3 Step 3: ASI determination**
- 24시간 전 구간에서  $|ASI| < 0.2$  여부를 판정.
  - 단 한시간이라도 초과하면 불합격하여 Step 1로 복귀 후 새 패턴 재생성

- 4 Step 4: Save**
- 전 시간대 합격 시 해당 일차의 제어봉 패턴, ASI, PPM, 연료 및 감속재 온도를 CSV로 저장.
  - 종료 시점 상태를 다음 일차의 초기조건으로 전달

## 2.2: 데이터 생산 – 데이터 생산 결과

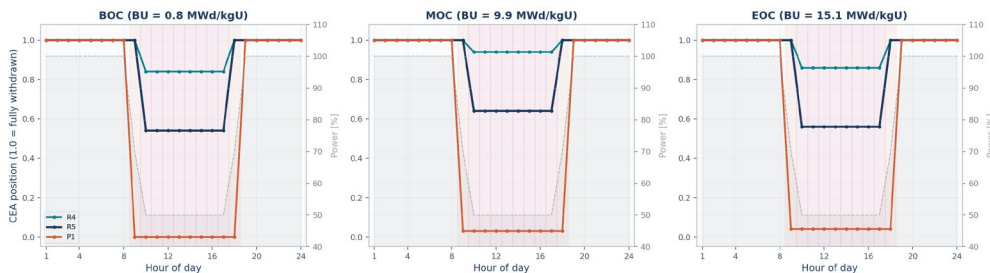


### 데이터 셋 형태

- 1.5년 (BU = 0 ~ BU=16) Full-Cycle에 대해 매일 100-50-100 출력 증감발 운전을 진행
- 시간당 최대 30%의 출력증감률 적용
- 12,312개의 데이터 포인트 보유

→ 전 주기에 걸쳐 ASI가  $\pm 0.2$  이내로 안정 유지되어 전주기에 이르는 안정적인 일일부하추종 시나리오에 대한 데이터셋을 얻음

CEA positions during daily load-following cycle (BOC → MOC → EOC)

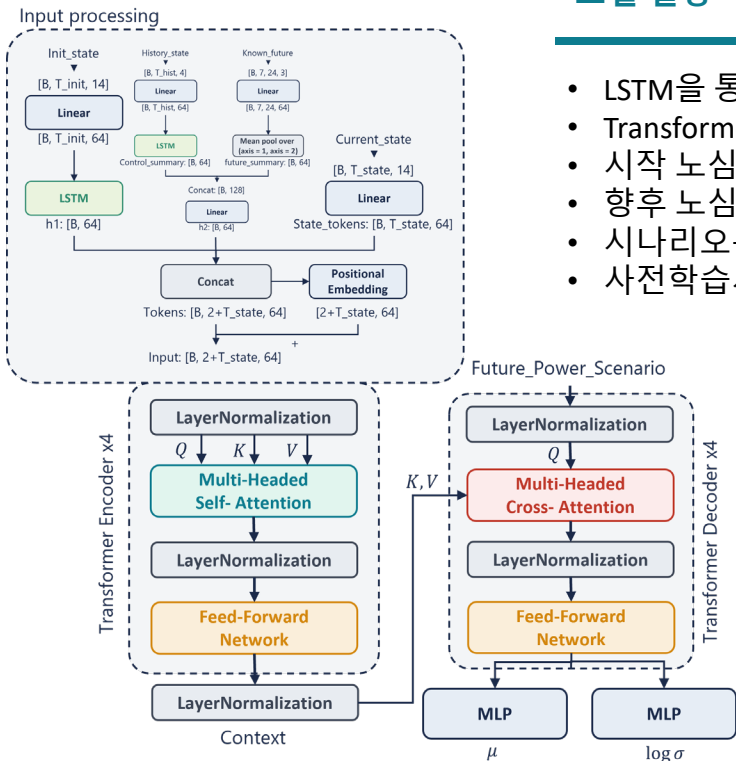


### 제어봉 운전 형태

- P뱅크 제어봉을 우선 삽입 후 고정
- 이후 R5, R4 뱅크 제어봉을 삽입
- PDIL로 인해 50%출력 증감발 운전에서는 R3뱅크까지 움직이지 못함

→ 단순한 제어봉 패턴으로 데이터 생산 시 로직 간소화에 도움, 사전모델 학습 시 노이즈 없는 명확한 정답라벨 제공

## 2.3: 모델 구축



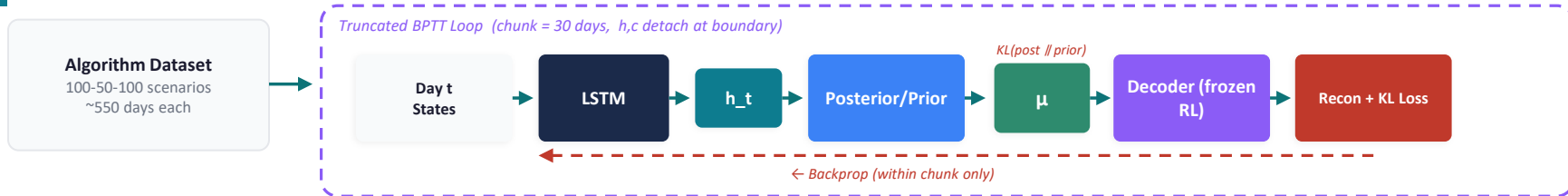
### 모델 설명

- LSTM을 통해 과거 이력 반영
- Transformer based 모델을 통해 현재 운전 맥락에 맞는 시나리오 생성
- 시작 노심 상태, 과거 노심 상태, 제어봉 운전 이력을 입력받아 현재의 Context 생산
- 향후 노심 출력 시나리오를 기반으로 Context와 Cross-Attention하여 시나리오 생성
- 시나리오는 향후 7일간 1시간 단위의 제어봉 위치 (R4,R5,P)임
- 사전학습시  $\log \sigma$ -head는 비활성, 강화학습 시 활성화

### 입출력 관계

- Init\_state: 초기 시작 시점 상태
    - 연료 온도, 냉각재 온도, 과거 봉산, ASI이력
  - History\_state: 누적 제어봉 위치 이력
    - 과거 R4,R5,P 제어봉 위치 이력
  - Current\_state: 현재 노심 상태
    - 현재 연료 온도, 냉각재 온도, 과거 봉산, ASI이력
  - Known\_future: 미래 출력 계획
    - 향후 출력 계획
- Context
- 제어봉 위치 •  $\log \sigma$  (사전학습때는 비활성)

## 2.4: 모델 학습

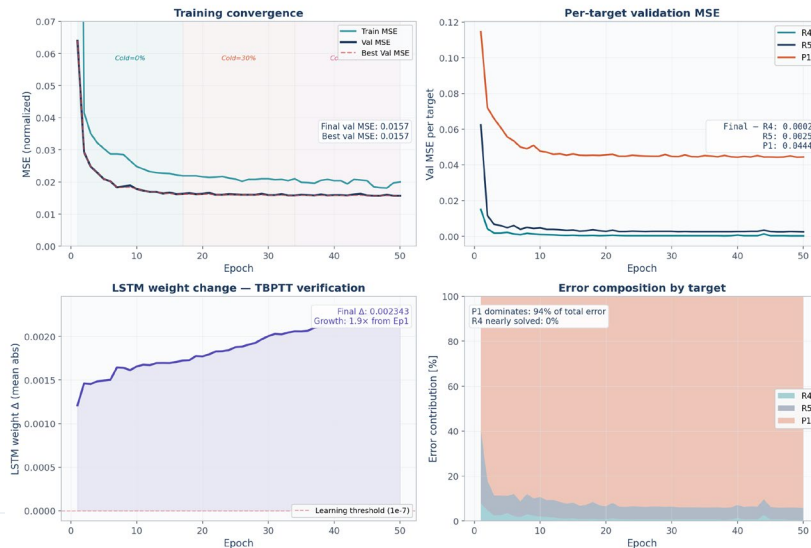


SL pre-training results (50 epochs, TBPTT, 3 targets: R4, R5, P1)

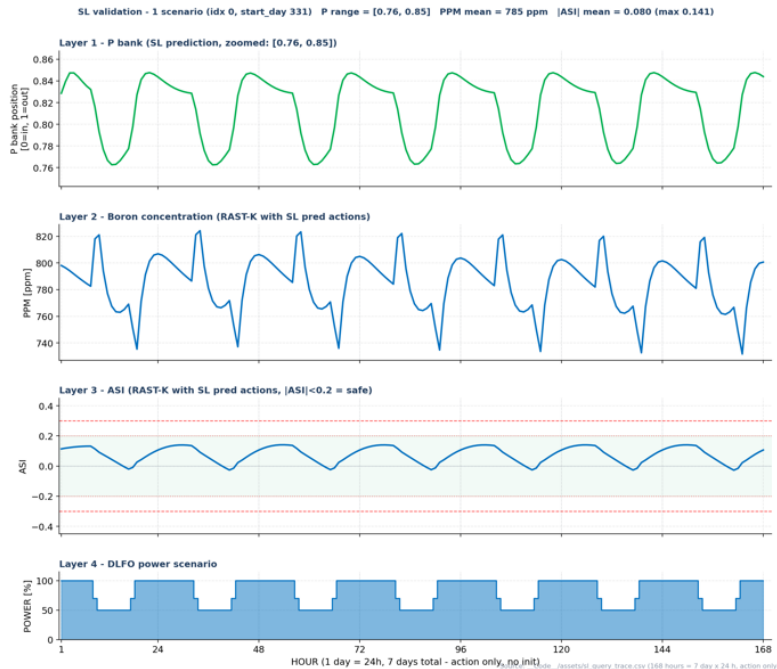
### Training Configuration

- **Method:** Supervised Learning
- **Loss function:**  $MSE(\mu_{pred}, y_{target})$
- **Prediction window:** 7 days
- **Start point:** random
- **Optimizer:** AdamW, lr=1e-4, gradient clip = 1.0
- **Epochs:** 50
- **Batch:** 32

CPU: Intel(R) Core(TM) Ultra 9 285K  
RAM: 32 GB



## 2.4: 모델 학습 - 결과



- 1 제어봉 시나리오 생성
- PDIL과 같은 제어봉 운전 제약에 맞는 시나리오 생성
  - 랜덤한 움직임이 없는 물리적으로 타당한 제어봉 시나리오 생성

- 2 붕소 농도
- 생성된 제어봉 시나리오에 대응되는 붕소 농도 계산
  - 출력 증감발에 따른 붕소 동향이 일관적임

- 3 ASI 범위
- 사전학습 목적인  $|ASI| < 0.2$ 를 전구간 만족
  - 생성된 제어봉과 붕소 농도 시나리오는  $|ASI| < 0.2$  조건을 만족함을 확인

- 4 출력 프로파일
- 생성된 제어봉 시나리오와 계산된 붕소 농도가 100-50-100의 탄력운전을 가능하게 함을 확인
  - $|ASI| < 0.2$ 를 만족하는 100-50-100 운전 시나리오 생성을 확인

➔ 유의미한 운전 시나리오만 생산하도록 탐색 공간을 규제하는 사전학습 목표 달성

# 03

## Reinforcement Learning

---

For fine-tuning of pre-trained model

## 강화학습의 목표 - $|ASI| < 0.2$ 만족 100-50-100 제어봉 조절 시나리오 개발 강화학습 Fine-tuning 진행

- ### 3.1: 플랫폼 개발

  - 노심해석코드와 Actor의 연결

### 3.2: 강화학습 디자인

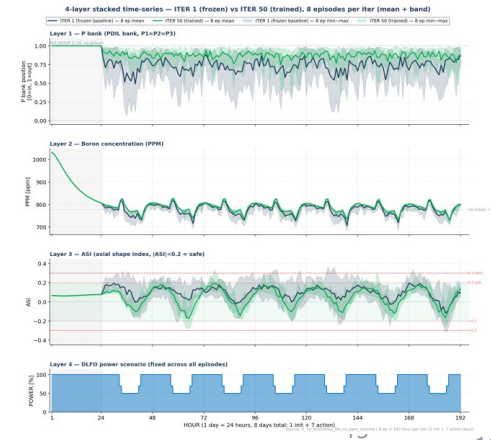
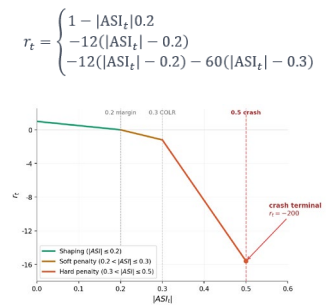
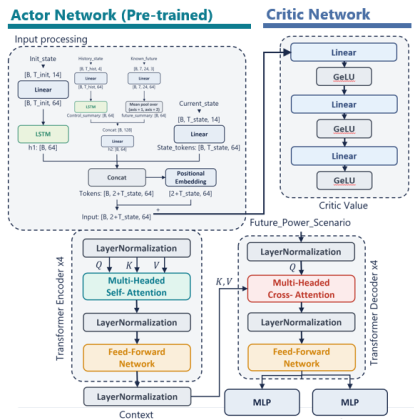
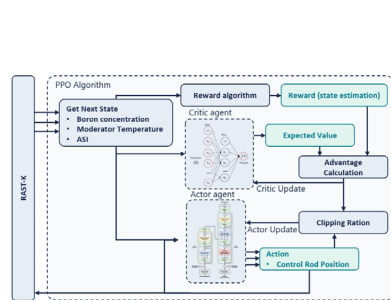
  - Loss 설계, PPO 방식
  - 7일에 대한 시나리오 생성 및 이에 대한 노심해석코드 검증

### 3.3: 리워드 함수 설정

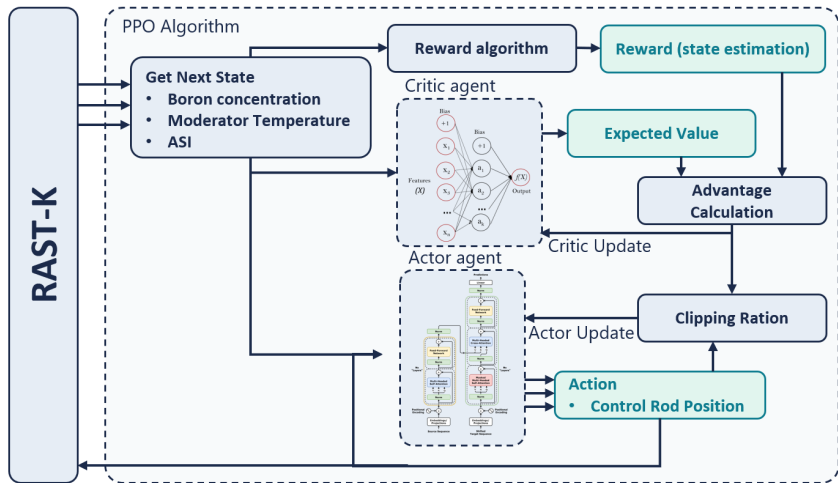
  - $|ASI|$  구간별 패널티 값 설정
  - $|ASI|$  중심의 리워드 함수 설계

### 3.4: 학습 진행

  - 학습 리워드 추세 분석
  - 생성된 제어봉 위치 시나리오에 따른 출력 조절 결과



## 3.1: 강화학습 플랫폼 개발



### 노심해석코드(RAST-K)와 연동

- Action값 기반 RAST-K 입력 파일 자동 생성 → 호출 명령어로 RAST-K 실행
- RAST-K 실행 후 생성된 출력 파일을 자동 파싱 → Next State로 변수 할당

→ subprocess 라이브러리를 통한 RAST-K 실행 및 파싱으로 강화학습 플랫폼에 연동

### PPO 알고리즘

- Value-based 강화학습은 연속적인 제어봉 위치 출력에 적합하지 않음
- Vanilla stochastic policy는 학습이 불안정함, → 업데이트를 Clipping하는 PPO 알고리즘 도입

→ RAST-K의 출력(ASI)을 기반으로 Reward 및 Critic Value (Reward 판단 기준) 생성, 이를 기반으로 Advantage를 계산하여 Loss 피드백

### Action 생성

```
with torch.no_grad():
    rollout = actor.rollout(
        init_states,
        history_lengths,
        init_states,
        scenario_batch,
        sample=True,
    )
```

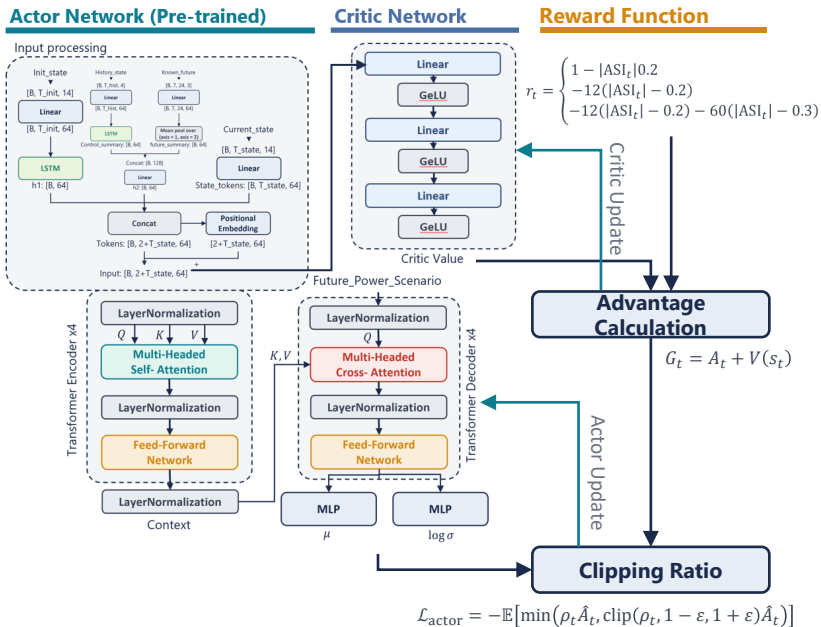
### RAST-K 호출

```
msgs = [{"verify_chunk",
        scenario_batch[w, chunk_start:chunk_start + chunk_days]
        .reshape(chunk_hours, -1).detach().cpu().numpy(),
        rollout["action"][w, chunk_start:chunk_start + chunk_days]
        .reshape(chunk_hours, -1).detach().cpu().numpy(),
        prev_cr[w]} for w in alive]
pool.send(alive, msgs)
results = pool.recv(alive)
```

### PPO 알고리즘

```
ratio = torch.exp(new_log_prob - old_log_prob)
surr1 = ratio * advantages
surr2 = torch.clamp(ratio, 1.0 - clip_eps, 1.0 + clip_eps) \
    * advantages
actor_loss = -torch.min(surr1, surr2).mean()
value_loss = F.mse_loss(value_preds, returns)
total_loss = actor_loss + value_coef * value_loss \
    - entropy_coef * entropy
```

## 3.2: 강화학습 모델 디자인



### 모델 설명 (PPO)

- Proximal Policy Optimization (PPO) 방식 사용
- Policy update를 한번에 멀리가지 않도록 Clipping하는 방법

### Loss

$$\mathcal{L}_{actor} = -\mathbb{E}[\min(\rho_t \hat{A}_t, \text{clip}(\rho_t, 1 - \epsilon, 1 + \epsilon) \hat{A}_t)]$$

$$\mathcal{L}_{critic} = \mathbb{E}[(V(s_t) - G_t)^2]$$

$$\mathcal{L} = \mathcal{L}_{actor} + 0.5 \cdot \mathcal{L}_{critic}$$

- Loss 함수에 Clipping을 적용하여 PPO 구현
- $A_t$ 는 행동이 평균보다 얼마나 좋은가를 의미
- PPO 성능은  $A_t$ 의 품질에 달려있으며 GAE를 통해 안정적으로 추정

### Advantage (Generalized Advantage Estimation (GAE))

$$G_t = A_t + V(s_t)$$

$$A_t = \delta_t + \gamma \lambda A_{t+1}, \quad \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

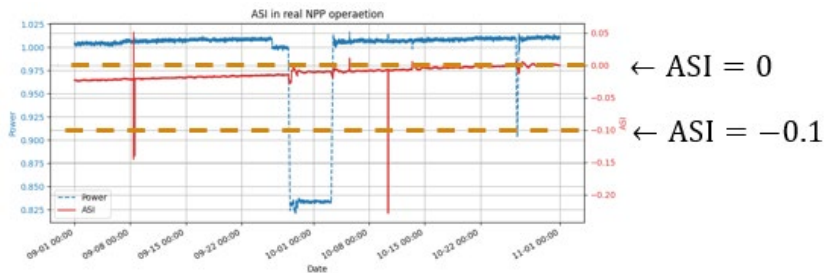
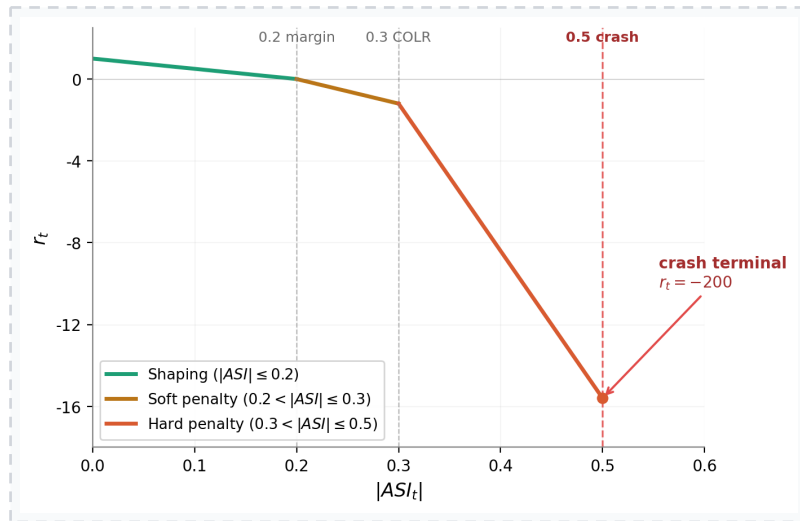
- 학습 초반엔  $v$ 가 랜덤이라 학습이 망가지기 쉬웠으나 학습 후반에는 너무 noisy하여 학습이 느림
- 이를 극복하기 위해 두 극단사이를 보간하여 조절함

## 리워드함수 설정

### ASI Reward

$$r_t = \begin{cases} 1 - |ASI_t|/0.2 & \text{if } 0 \leq |ASI_t| \leq 0.2 \\ -12(|ASI_t| - 0.2) & \text{if } 0.2 < |ASI_t| \leq 0.3 \\ -12(|ASI_t| - 0.2) - 60(|ASI_t| - 0.3) & \text{if } 0.3 < |ASI_t| \leq 0.5 \end{cases}$$

- $|ASI|$  0.2 구간: 목표로 하는 보수적인 안전구간으로 탄력운전 중 모든 ASI를 0.2 이내로 유지하고자 목표
- $|ASI|$  0.3 구간: 운전조건에서 명시한 최대 ASI구간, 이 이상으로는 절대 넘어가면 안됨
- $|ASI|$  0.3 이상: 사용할 수 없는 시나리오로 0.5가 될 시 노심해석코드 연산이 불가능하여 종료됨

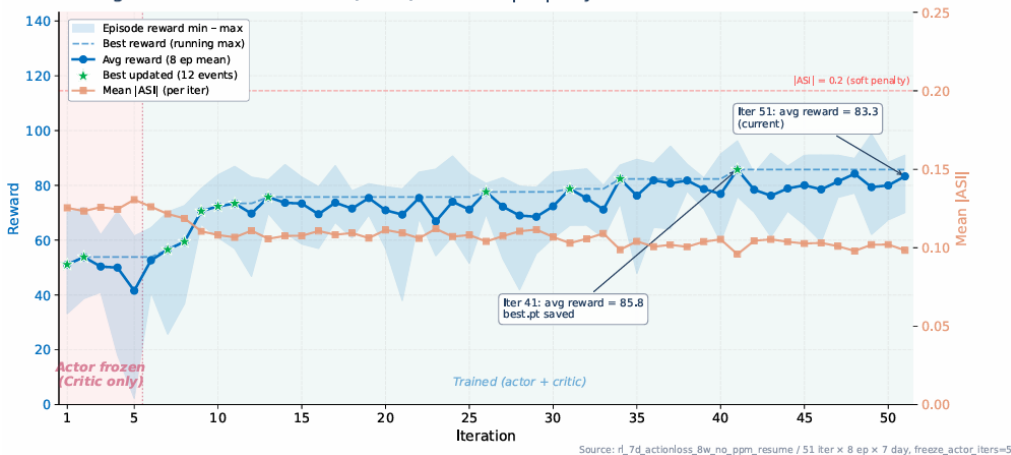


### 실제 원자력발전소 운전 데이터기반 ASI 범위 설정

- 실제 원자력발전소에서는 안전상의 이유로  $|ASI|$ 를 최대 0.1이상으로 넘지 않도록 하기 위해 노력함
- 100-80-100운전 데이터에서  $|ASI|$ 를 0에 가깝게 유지하려는 기동이 보임

## 모델 학습

RL training: reward rises 51.1 → 83.3 (+63%) and mean |ASI| stays bounded — actor unfreezes after iter 5



- Reward에 큰 변동이 있으나 소폭 상향세의 trend를 가짐.
- Reward 80대의 의미는 0.2를 넘는 ASI가 5%의 미만이며 0.3을 넘는 ASI가 극소량에 불과하다는 의미로 해석됨.
- 추후 더 많은 시간을 가지고 Reward를 향상시킬 여지가 있음

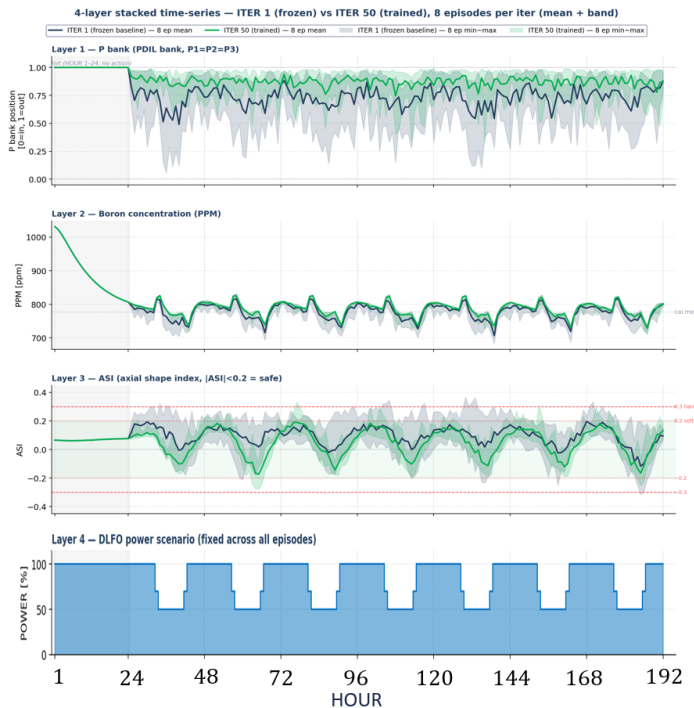
# 04

## Test

---

Approach and key variables

## 강화학습의 결과



- 1 **제어봉 시나리오 생성**
  - 사전학습에 비해 노이즈가 많아진 제어봉 움직임을 보임
  - ITER가 진전될수록 제어봉 움직임을 최소화 하려는 움직임을 보임
  
- 2 **봉소 농도 (자동 계산)**
  - 제어봉 위치가 입력되면 목표 출력에 맞는 봉소 농도를 RAST-K가 계산
  - ITER가 진전될 수록 봉소 농도의 노이즈가 줄어듦
  - 그러나, ITER가 진전될 수록 봉소의 사용량이 증가하는 경향을 보임
  
- 3 **ASI 범위**
  - ITER 초반에는 ASI 노이즈가 많아 |ASI|<0.2를 위반하는 경우가 다수 존재함
  - ITER 후반에 갈 수록 노이즈는 줄어 |ASI|<0.2 위반 수가 확연히 줄어듦
  
- 4 **출력 프로파일**
  - 예측된 제어봉 및 계산된 봉소 농도를 기반으로 100-50-100 일일부하추진 운전 성공

➔ 100-50-100 탄력운전을 하기 위한 시나리오 생산에 성공

# 05

## Conclusion & Future Work

---

Approach and key variables

## Conclusion

- 탄력운전을 하기 위한 제어봉 시나리오 생성 알고리즘 개발, 이를 통해 사전학습 데이터셋 생성
- 지도학습으로 학습된 모델을 강화학습으로 Fine tuning하여 탐색 공간을 넓혀 다양한 제어봉 시나리오를 생성
- 강화학습을 통해 BOC에서 7일간 안정적 ( $|ASI| < 0.2$ )으로 탄력운전을 하는 제어봉 시나리오 생산
- 다만, 제어봉 관련 리워드 함수가 없어 사전학습에 비해 제어봉 위치의 노이즈가 심해짐.

## Future Work



### Reward 함수 개선

ASI뿐만이 아닌 붕소 농도나 제어봉 움직임 관련을 Reward함수에 반영하여 개선



### Model Validation

강화학습으로 학습된 모델의 결과를 검증

# Thank You

---

Questions & Discussion

Junhyeong Bang | [bjun201goseguly@kaist.ac.kr](mailto:bjun201goseguly@kaist.ac.kr)

NICA Lab · Dept. of Nuclear and Quantum Engineering · KAIST

