Resilient Autonomous Operation for SMR Abnormal Conditions based on the Multi-Agent Reinforcement Learning and Abstraction Hierarchy

Gwanwoo Kim[a] and Jonghyun Kim[b]
[a] Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, kwanwoo1017@kaist.ac.kr
[b] Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea, jonghyun.kim@kaist.ac.kr

Abstract: Small modular reactors (SMRs) are emerging as a strategic energy solution as existing grid infrastructure struggles to meet rapidly growing electricity demand. Their load-following capability complements intermittent renewables such as solar and wind, while providing stable, carbon-free baseload generation. These characteristics make SMRs well-suited for distributed deployment in response to surging demand from sectors such as AI data centers. Despite these advantages, SMRs still inherit a fundamental operational challenge from conventional nuclear plants: responding to abnormal conditions remains one of the most demanding tasks for operators. Large-scale plants such as APR1400 require operators to manage over 80 abnormal operating procedures, each covering multiple event types. Compounding this burden, abnormal operation is the most frequent operating regime and occurs in diverse, unpredictable forms, making consistent response inherently difficult. Indeed, 86% of the 183 unplanned reactor trips at Korean nuclear power plants since 2000 were triggered by abnormal situations. These challenges are further amplified in the SMR context, where a single control room manages multiple modules with as few as three operators. This reduced staffing is compounded by the monitoring complexity that passive safety systems introduce, making high-level automation essential to ensure both safety and operational reliability. This study proposes a resilient abnormal operation automation framework for an integral pressurized water reactor (iPWR), developed through four sequential stages. First, Abstraction Hierarchy (AH) modeling decomposes the iPWR into five functional-physical levels with quantitative formulations for each node. Second, a Hierarchical Multi-Agent Reinforcement Learning (H-MARL) architecture is designed by mapping each AH node to an independent RL agent in a manager-worker hierarchy. Third, a parallel simulation platform is constructed using multiple iPWR simulator instances to enable scalable training. Fourth, training is conducted with randomized fault parameters across selected abnormal scenarios to promote robust generalization.

Keywords: Reinforcement Learning, Small Modular Reactor, Autonomous Operation

1. INTRODUCTION

Small modular reactors (SMRs) are increasingly viewed as a strategic response to the rapid growth in electricity demand that existing grid infrastructure struggles to absorb, driven in part by the expansion of energy-intensive sectors such as artificial intelligence data centers. SMRs are expected to provide a stable, low-carbon electricity supply and, depending on design and operating strategy, may also support flexible operation that complements variable renewable generation. Their modular, factory-built design further enables incremental capacity expansion suited to distributed grids [1, 2].

These same characteristics, however, reshape how the plant is operated. To secure economic viability, SMRs are typically designed for multi-module operation, in which a single control room supervises several reactor units simultaneously with far fewer operators than a large commercial plant; representative concepts assign three operators to four modules or to as many as twelve modules. While passive safety systems reduce the need for active operator intervention, the combination of multi-

module supervision, reduced staffing, infrequent manual intervention, and the difficulty of inferring the state of each unit increases the burden of overall monitoring, so that advanced monitoring and decision support becomes important rather than optional. These factors create strong technical motivation for higher levels of automation and operator-support functions in SMR control rooms [3].

While rule-based automation has proven effective for normal operations [4], abnormal operations remain far more challenging. Large plants such as the APR1400 require operators to manage more than eighty abnormal operating procedures, each covering multiple event types, and operators must be trained to identify the correct event from its alarm and symptom patterns [6, 7]. Abnormal situations also occur in diverse, unpredictable forms, which makes a consistent and accurate response inherently difficult; according to OPIS records for the Korean fleet since 2000, they represent 157 out of 183 unplanned reactor-trip cases, making them the dominant recorded category. Unlike design-basis accidents, which have well-defined mitigation strategies, abnormal states often arise from stochastic failures or unforeseen combinations of malfunctions for which existing procedures may not provide a unique or immediately identifiable response path. In such situations operators must rely on their own diagnostic ability, a task that becomes increasingly error-prone under stress and time pressure [5]. The reduced staffing of SMRs amplifies this risk, making automated abnormal-operation support essential to relieve operator workload and preserve safety.

To address this challenge, this study develops a resilient SMR operation concept whose ultimate goal is to reduce dependence on external intervention by enabling the plant to restore degraded safety-related functions based on real-time operational states. In its ideal form, the concept aims for self-recovery with minimal or no external intervention within a predefined operational envelope. The proposed logic is intended to be triggered when selected operational indicators deviate from predefined normal operating bands, and its defining feature is diagnosis-independent recovery, in the sense that recovery actions are selected based on degraded functional states rather than on explicit classification of a fault scenario: rather than first identifying the specific cause, the system prioritizes restoring the affected safety functions, which strengthens robustness against events that were never explicitly anticipated [8]. Reinforcement learning (RL) is adopted because it provides a data-driven framework for learning feedback policies over continuous plant states and can be combined with scenario randomization to improve robustness across a range of abnormal conditions, rather than relying solely on predefined event-response mappings [9]. RL has already been applied to autonomous power-increase operation [10], multi-objective startup control [11], multi-agent load-following control of microreactors [12], and the handling of multiple emergent accidents [13], and multi-agent RL has likewise been used to enhance resilience in other safety-critical infrastructures [14]. However, these studies generally do not provide a functionally structured mechanism for constraining the action space according to the current safety-function degradation, which limits their applicability to undefined abnormal conditions with large actuator sets. Directly training RL policies for abnormal operation is also difficult because the problem is knowledge-intensive: many actuators are available, only a small context-relevant subset matters at any moment, and naive exploration wastes data and destabilizes learning. The proposed framework is developed and validated on the iPWR simulator, an educational integral pressurized water reactor simulator distributed by the IAEA.

This paper makes two contributions. The first contribution is an operational concept for diagnosis-independent self-recovery, in which abnormal-operation control is formulated around the restoration of degraded safety functions rather than explicit event classification. The second is the quantitative operationalization of the Abstraction Hierarchy (AH), in which each AH node is assigned a mathematical formulation that is used directly as a learning signal. These two ideas are combined into a four-stage development framework: (1) AH-based functional decomposition with quantitative node formulations, (2) design of a hierarchical multi-agent RL (H-MARL) architecture, (3) construction of a parallel simulation platform, and (4) training with randomized fault parameters. The remainder of this paper follows these stages: Section 2 presents the AH modeling, Section 3 the H-MARL design, Section 4 the case study and results, and Section 5 the conclusion. The present study focuses on framework development and provides an initial validation using a preliminary model of the steam-generation subsystem.

2. AH-BASED iPWR FUNCTIONAL DECOMPOSITION

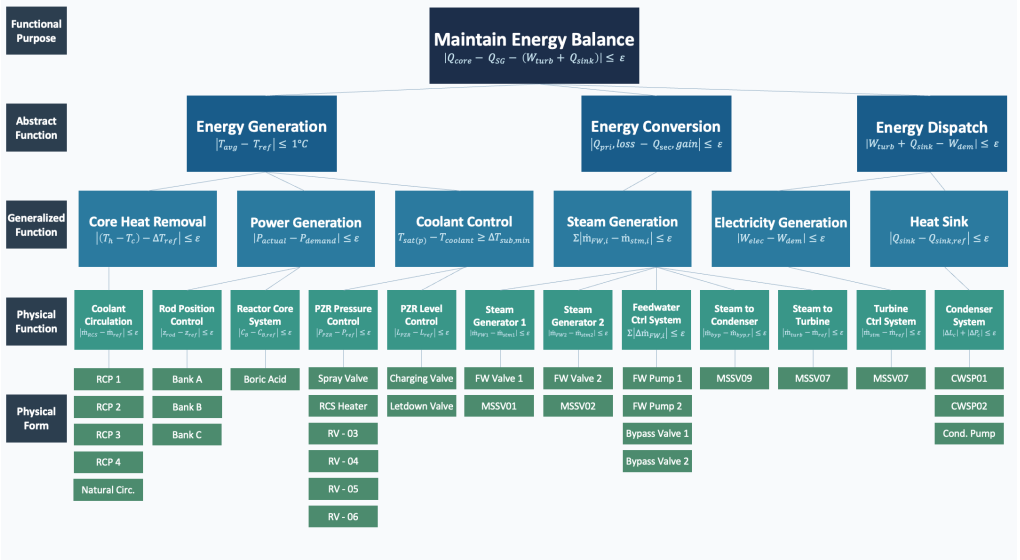
A resilient SMR must act without explicit fault diagnosis, which means its control logic should prioritize maintaining functional purposes over monitoring individual physical components. This requires a structured representation that connects raw physical measurements to the higher-level functions they serve. Work Domain Analysis (WDA) provides such a representation by describing the plant in terms of its invariant purposes, governing physical principles, and means-ends relations, so that control is anchored to the safety-relevant functions that must be maintained rather than to a fixed procedure library [17]. Anchoring control to functions in this way is what allows the system to remain valid under uncertainty, even when the initiating event is unclear or compounded.

The structural backbone of WDA is the Abstraction Hierarchy (AH) proposed by Rasmussen, a multilevel framework that decomposes a system into five levels ranging from physical form to functional purpose [15]. Adjacent levels are connected by means-ends links: the downward (how) direction explains how a higher-level function is realized through lower-level functions and physical resources, while the upward (why) direction explains why a given component or function is required to achieve a higher-level goal. The AH has been widely used to model process plants in the power domain [16] and to translate analysis insights into control and interface design [17].

2.1. AH Model of the iPWR

Based on WDA, this study constructs an AH of the iPWR simulator with five levels: Functional Purpose, Abstract Function, Generalized Function, Physical Function, and Physical Form. Figure 1 presents the overall AH diagram. At the top level, the Functional Purpose is defined as maintaining the overall energy balance across the core, steam generators, and secondary side. This purpose is realized through three Abstract Function branches: Energy Generation, which covers primary-side thermal-hydraulic processes including core heat removal, power generation, and coolant control; Energy Conversion, which covers secondary-side steam generation and feedwater heat exchange; and Energy Dispatch, which covers electricity generation and ultimate heat rejection. Each branch is decomposed through the Generalized Function and Physical Function levels down to the Physical Form level, where individual actuators such as reactor coolant pumps, control rods, spray, charging and letdown valves, and feedwater pumps are identified as control endpoints. This decomposition maps high-level control objectives to process-level functions and ultimately to component-level actuation pathways, providing the structural scaffold for state interpretation, reward construction, and the organization of control-relevant variables.

Figure 1: Overall Abstraction Hierarchy of the iPWR simulator



2.2. Quantitative Formulation of AH Nodes

The complete AH provides a qualitative functional structure of the iPWR simulator, as shown in Figure 1. For autonomous control, however, the AH nodes relevant to the target subsystem must be made computable. Therefore, this study defines quantitative functional indicators for selected nodes along the Energy Conversion path used in the preliminary case study. Each selected node is assigned an operational indicator and a tolerance margin that together represent the degree of functional deviation under the present simulator-based setting; these formulations are intended not as complete first-principles thermal-hydraulic models, but as operational indicators that translate functional deviations into learning-relevant information. They serve two roles. First, they provide quantitative criteria for identifying which functional branch in the hierarchy is deviating from its intended operating condition. Second, they are used as inputs to the reward design of the RL agents, allowing the steam-generation subsystem to be evaluated across multiple abstraction levels rather than only through component-level variables.

While the complete iPWR AH is presented in Figure 1, Table 1 summarizes representative quantitative formulations for the selected AH nodes along the Energy Conversion path, which contains the steam-generation subsystem used for preliminary validation. The variables are defined as follows: Q denotes a heat or power rate (Q_{core} for core thermal power, Q_{SG} for steam-generator heat transfer, $Q_{pri,loss}$ and $Q_{sec,gain}$ for primary heat loss and secondary heat gain, Q_{sink} and $Q_{sink,ref}$ for heat rejection and its reference); W denotes work or electrical power (W_{turb} for turbine work, W_{elec} for electrical output, W_{dem} for demand); $\dot{m}$ denotes a mass flow rate ($\dot{m}_{FW,i}$ and $\dot{m}_{stm,i}$ for the feedwater and steam flows of steam generator i, $\dot{m}_{byp}$ and $\dot{m}_{byp,r}$ for the actual and reference bypass flows, $\dot{m}_{turb}$ and $\dot{m}_{ref}$ for the turbine steam flow and its reference); $\Delta\dot{m}_{FW,i}$ is the feedwater flow deviation; and ε is the tolerance margin used for steady-state verification. For reference, the remaining Generalized Function nodes are formulated analogously, for example Core Heat Removal as $|(T_h - T_c) - \Delta T_{ref}| \leq \varepsilon$, Power Generation as $|P_{actual} - P_{demand}| \leq \varepsilon$, and Coolant Control as $T_{sat(p)} - T_{coolant} \geq \Delta T_{sub,min}$.

Table 1: Representative quantitative formulations for the selected AH nodes

AH Level	Node	Functional Indicator
Functional Purpose	Maintain Energy Balance	$\lvert Q_{core} - Q_{SG} - (W_{turb} + Q_{sink})\rvert \leq \varepsilon$
Abstract Function	Energy Conversion	$\lvert Q_{pri,loss} - Q_{sec,gain}\rvert \leq \varepsilon$
Generalized Function	Steam Generation	$\Sigma\lvert\dot{m}_{FW,i} - \dot{m}_{stm,i}\rvert \leq \varepsilon$
Physical Function	Steam Generator 1	$\lvert\dot{m}_{FW1} - \dot{m}_{stm1}\rvert \leq \varepsilon$
Physical Function	Steam Generator 2	$\lvert\dot{m}_{FW2} - \dot{m}_{stm2}\rvert \leq \varepsilon$
Physical Function	Feedwater Control System	$\Sigma\lvert\Delta\dot{m}_{FW,i}\rvert \leq \varepsilon$
Physical Function	Steam to Condenser	$\lvert\dot{m}_{byp} - \dot{m}_{byp,r}\rvert \leq \varepsilon$
Physical Function	Steam to Turbine	$\lvert\dot{m}_{turb} - \dot{m}_{ref}\rvert \leq \varepsilon$

3. HIERARCHICAL MULTI-AGENT REINFORCEMENT LEARNING DESIGN

3.1. Background

Standard single-agent RL struggles in complex multi-system environments because the joint action space grows high-dimensional and its exploration becomes inefficient and unstable. The iPWR is exactly such an environment, containing multiple interacting systems—the reactor coolant system, feedwater system, chemical and volume control system, and others—so that controlling all available actuators with a single monolithic agent would lead to a large and weakly structured action space, making exploration inefficient and potentially unstable. Multi-agent RL (MARL) can mitigate this difficulty by distributing control across several cooperating agents, each responsible for a subset of the action space. Hierarchical RL (HRL), in turn, introduces a goal-conditioned structure in which a higher-level agent sets subgoals and lower-level agents act to achieve them [19, 20]. Hierarchical multi-agent RL (H-MARL) combines the two, so that multiple agents cooperate within a hierarchical means-ends

structure [18]; related work has also coupled such hierarchies with safety mechanisms for safety-critical autonomous systems [21].

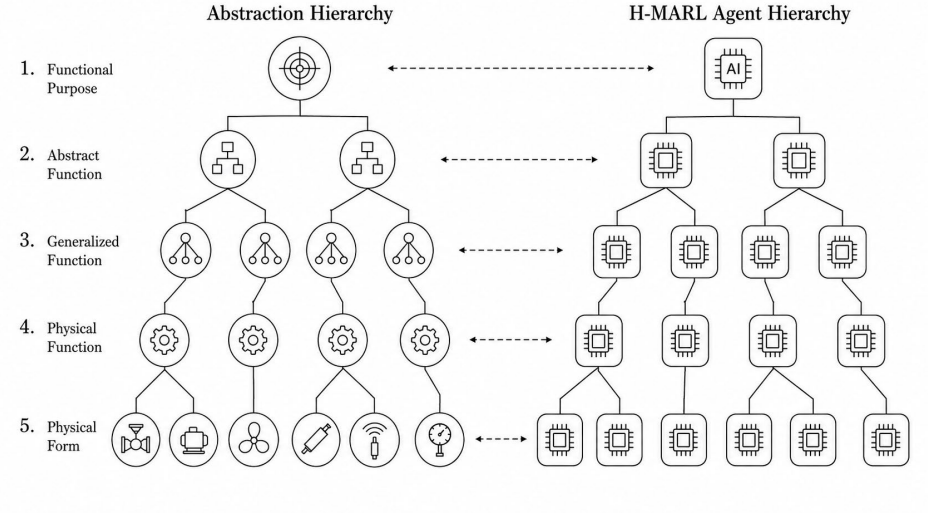
3.2. Mapping the AH onto an H-MARL Architecture

The central design choice of this study is to instantiate each AH node as a computational RL agent, so that the means-ends structure of the AH becomes the communication and control structure of the H-MARL system. Each agent is associated with a specific AH node and receives observations relevant to that node, including raw simulator variables and AH-derived functional indicators. The output of each agent depends on its abstraction level: higher-level agents output subgoals or reference targets to lower-level agents, whereas lower-level agents output actuator-level commands to the simulator.

Following the means-ends interpretation of the AH, downward links specify how higher-level functional objectives are realized through lower-level functions and physical resources, while upward links indicate why lower-level actions are relevant to higher-level functional objectives. In the proposed H-MARL architecture, this relation is implemented as downward subgoal propagation and upward state and performance feedback.

Figure 2 shows the resulting framework. Higher-level agents, spanning from the Functional Purpose toward the Generalized Function levels, monitor the functional balance and reassign subgoals when a deviation is detected, while lower-level agents, spanning from the Physical Function to the Physical Form levels, drive the actuators to satisfy the subgoals assigned to them. Each agent evaluates the condition of its own node using the quantitative formulation defined in Section 2 and acts toward its assigned goal, allowing the system to assess and control the plant coherently across abstraction levels.

Figure 2: One-to-one mapping between Abstraction Hierarchy and H-MARL agent structure



3.3. RL Problem Formulation

Each agent receives an agent-specific observation vector composed of simulator measurements relevant to its associated AH node, such as temperatures, pressures, flow rates, and component states, together with AH-derived functional indicators; the observation dimensions of the active agents are listed in Section 4. The action space differs by agent level. Manager agents do not directly manipulate plant actuators; instead, they emit continuous subgoal parameters or reference targets for their worker agents. Worker agents associated with continuously modulated devices, such as feedwater and steam valves, use discrete delta actions selected from {−2%, 0, +2%} of valve opening, while worker agents associated with latched devices, such as the heat-exchanger bypass valve and feedwater pumps, use open/close or on/off actions.

Rewards are hierarchical. Both manager and worker agents construct their rewards from the tolerance margins of their associated AH nodes. In this preliminary implementation, the tolerance margin was set to 3% as a design parameter for defining functional deviation during simulator-based training. A worker agent is rewarded for reducing deviation from its assigned reference condition, while a manager agent is rewarded for maintaining or restoring the functional balance of its branch.

3.4. Agents and Learning Algorithm

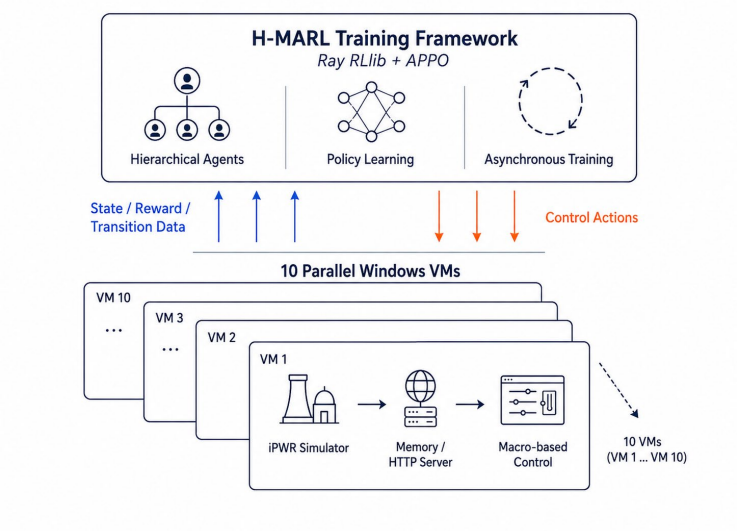
Under this design, a manager agent monitors the functional balance of its branch and reassigns subgoals when a deviation occurs, and the worker agents drive their assigned actuators to meet those subgoals. The agents are trained with Asynchronous Proximal Policy Optimization (APPO) using the Ray RLlib library. APPO is a distributed variant of Proximal Policy Optimization (PPO) [22] that collects samples asynchronously and uses importance sampling to mitigate the off-policy bias introduced by asynchronous updates [23]. APPO is used because its asynchronous actor-learner structure reduces the bottleneck caused by slow simulator instances and enables training across many parallel iPWR simulator environments, following the principle used in scalable distributed RL architectures [24], as described in Section 4.

4. TRAINING AND RESULTS

4.1. Parallel Simulation Platform

The training environment is the iPWR simulator, an integral pressurized water reactor simulator. A key practical constraint is that the simulator does not support step-based control and runs only in real time; acceleration beyond roughly twice real time becomes unstable, making a time-acceleration approach impractical. Under these constraints, the most practical way to increase training throughput was to run multiple independent simulator instances in parallel. To this end, the platform runs ten Windows virtual machines concurrently on a single server equipped with two Intel Xeon Silver processors, 128 GB of RAM, and an NVIDIA L40S GPU. Each VM hosts an isolated iPWR simulator instance, providing separate simulator states and independent episode execution. Communication is handled by an HTTP server running inside each VM that reads and manipulates the simulator memory to implement the RL actions; actions that cannot be realized through memory manipulation are implemented as macros. The Ray RLlib training process communicates with each VM over HTTP, collecting observations and dispatching actions asynchronously, following the distributed actor-learner principle described in Section 3 [24]. Figure 3 shows the overall platform architecture.

Figure 3: Parallel H-MARL training platform built from ten iPWR simulator instances



4.2. Abnormal Scenarios and Randomization

Two abnormal scenarios were selected to train the preliminary model: an intermediate heat-exchanger tube rupture and a feedwater-pump performance degradation, the latter included as a scenario common to the participating institutions. Both scenarios involve the feedwater control system as the primary responding subsystem and are well suited to demonstrating the H-MARL learning objective. To reduce overfitting to a fixed disturbance profile, the ramp time and severity of each fault were randomized on a per-episode basis. This randomization encourages the agents to learn recovery policies that remain effective across the sampled range of fault intensities and onset rates within the selected scenario classes. This use of randomized fault parameters follows the principle of domain randomization for robust and safe RL [25, 26]. Table 2 lists the scenarios and their randomization ranges.

Table 2: Abnormal scenarios used for training.

Component	Failure Type	Reactor Symptom	Ramp Time	Severity
Heat Exchanger 2	Tube rupture	Feedwater leakage	0–30 s	30–60%
Feedwater Pump 1	Performance reduction	Feedwater flow reduction	0–30 s	50–100%

4.3. Agent Configuration

The full AH model yields 22 agents spanning levels L1–L4. The Physical Form level, L5, is excluded because physical components are treated as actuator endpoints rather than independent decision-making agents in the present implementation. In the current preliminary scope, the six agents of the steam-generation subsystem are active, while the remaining sixteen are retained in the framework for future scenario expansion. Within the active hierarchy, the Steam Generation node acts as the manager, emitting continuous subgoals to five worker agents and dynamically adjusting their operating targets. The workers are Steam Generator 1, Steam Generator 2, the Feedwater Control System, Steam to Turbine, and Steam to Condenser. Continuously modulated devices such as the feedwater and steam valves are adjusted by the discrete delta actions defined in Section 3, while latched devices such as the heat-exchanger bypass valve and the feedwater pump use open/close or on/off actions. Table 3 summarizes the configuration of the active agents.

Table 3: Configuration of the active agents (steam-generation subsystem)

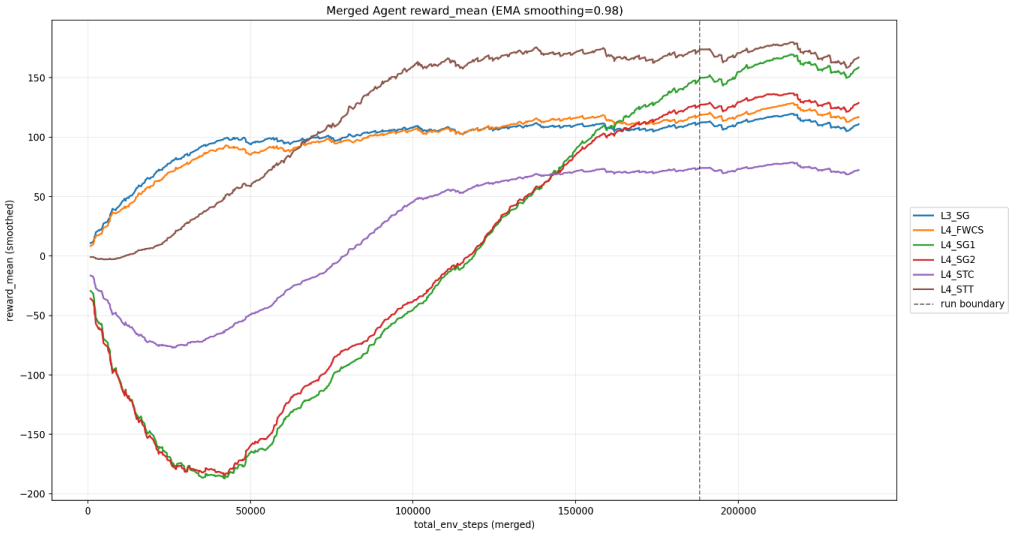
Agent	Role	Obs. Dim.	Hidden Layer	Actuators	Action Space
Steam Generation	Manager	20	[128, 64]	–	Continuous (5)
Steam Generator 1	SG1 feedwater & safety valve control	6	[64, 32]	FWV1, MSSV01	Discrete (9)
Steam Generator 2	SG2 feedwater & safety valve control	6	[64, 32]	FWV2, MSSV02	Discrete (9)
Feedwater Control System	HX bypass & feedwater pump control	12	[128, 64]	HXBypass2, FWP2	Discrete (4)
Steam to Turbine	Turbine control valve control	5	[64, 32]	MSSV09	Discrete (3)
Steam to Condenser	Condenser control valve control	5	[64, 32]	MSSV07	Discrete (3)

4.4. Training Convergence

The reward curves indicate learning progress over approximately 230,000 training steps. The L3_SG, L4_FWCS, and L4_STT agents showed stable reward improvement from early in training and reached a relatively stable reward region after approximately 100,000 steps. The L4_SG1 and L4_SG2 agents

initially incurred large negative rewards during exploration, but their rewards recovered sharply after approximately 150,000 steps. The L4_STC agent also showed an overall increasing trend, although its stabilization was slower than that of the other agents. These results are interpreted as preliminary evidence of trainability in the proposed AH-HMARL structure, while additional training and repeated runs are required to confirm convergence stability. Figure 4 shows the per-agent learning curves.

Figure 4: Per-agent learning curves over approximately 230,000 training steps



4.5. Autonomous Recovery Behavior

The trained agents produced autonomous control responses consistent with the intended recovery objectives in both scenarios. In the feedwater-pump degradation scenario, immediately after fault injection the responsible agent selected a pump start action that compensated for the feedwater-flow reduction, after which the feedwater and steam flows returned toward their normal ranges in the observed trajectory. In the intermediate heat-exchanger tube-rupture scenario, the agent responded by opening the heat-exchanger bypass valve, and the feedwater flow showed a stabilizing trend. Figures 5 and 6 present the variable trajectories and action histories for the two scenarios. These behaviors were achieved without encoding scenario-specific abnormal-operation procedures as explicit control rules, which supports the diagnosis-independent recovery concept of the proposed framework. Overall, this case study provides preliminary qualitative validation that the proposed AH-HMARL framework can be implemented on a parallel iPWR simulator platform and can learn recovery-oriented control behavior for selected steam-generation subsystem abnormalities without explicitly encoded scenario-specific procedures.

Figure 5: Recovery behavior in the feedwater-pump degradation scenario

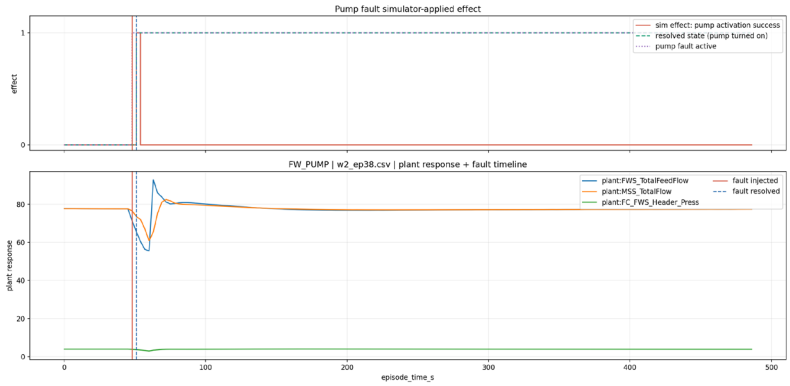
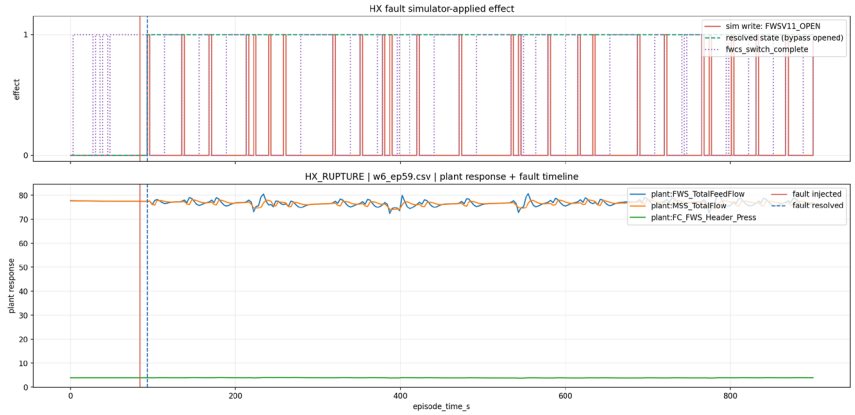


Figure 6: Recovery behavior in the intermediate heat-exchanger tube-rupture scenario



5. CONCLUSION

This study proposed a resilient abnormal-operation automation framework for iPWR-based SMRs that integrates a quantitative Abstraction Hierarchy with hierarchical multi-agent reinforcement learning architecture. The framework is based on a diagnosis-independent self-recovery concept that prioritizes restoring degraded safety-related functions over explicit root-cause classification, thereby providing a functional basis for responding to abnormal conditions without scenario-specific event diagnosis. Using Work Domain Analysis, the iPWR was decomposed into Rasmussen's five abstraction levels, and AH nodes were mathematically formulated to serve as deviation-detection criteria and reward-design inputs for the RL agents. The H-MARL architecture was then constructed by instantiating AH nodes as agents in a manager-worker hierarchy, and APPO was used to support asynchronous sample collection and distributed learning across parallel simulator instances. A parallel simulation platform built from ten isolated iPWR instances made scalable training practical despite the real-time-only nature of the simulator.

An initial case study was conducted using the six active agents of the steam-generation subsystem. The reward curves showed learning progress over approximately 230,000 training steps, and the trained agents produced recovery-oriented autonomous control responses in two abnormal scenarios: feedwater-pump degradation and heat-exchanger tube rupture. These responses were obtained without encoding scenario-specific abnormal-operation procedures as explicit control rules. The results should therefore be interpreted as qualitative proof of concept for the proposed framework rather than as a complete performance demonstration, since the present scope is limited to a single subsystem and a small set of abnormal scenarios.

Future work will proceed along three directions. First, additional training and repeated runs will be performed to confirm convergence stability and to evaluate robustness across wider ranges of fault severity and ramp time. Second, scenario coverage will be extended to additional abnormal conditions, such as reactor-coolant-pump trip, to examine whether the proposed AH-HMARL structure remains effective across a broader spectrum of events. Third, the sixteen currently inactive agents in other subsystems will be incrementally activated to evaluate the scalability and control capability of the framework in more complex, multi-system abnormal situations. In parallel, execution-level safety mechanisms, such as action masking and constraint-based filters, will be investigated to reduce physically inadmissible actions during exploration and execution [21, 27, 28].

Acknowledgements

This research was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIT) (No. GTL24031-000). And 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] J. I. Lee. "Review of Small Modular Reactors: Challenges in Safety and Economy to Success," Korean Journal of Chemical Engineering, 41(10), pp. 2761-2780, (2024).
[2] B. Mignacca and G. Locatelli. "Economics and finance of Small Modular Reactors: A systematic review and research agenda," Renewable and Sustainable Energy Reviews, 118, p. 109519, (2020).
[3] J. Hartmann, J. Hyvärinen, and V. Rintala. "The operator and the seven small modular reactors — An estimate of the number of reactors that a single reactor operator can safely operate," Nuclear Engineering and Design, 418, p. 112929, (2024).
[4] J. Kim, S. Lee, and P. H. Seong. "Autonomous Nuclear Power Plants with Artificial Intelligence," Lecture Notes in Energy, vol. 94, Springer, (2023).
[5] A. D. Swain and H. E. Guttmann. "Handbook of human-reliability analysis with emphasis on nuclear power plant applications," NUREG/CR-1278, (1983).
[6] J. M. Kim, G. Lee, C. Lee, and S. J. Lee. "Abnormality diagnosis model for nuclear power plants using two-stage gated recurrent units," Nuclear Engineering and Technology, 52(9), pp. 2009-2016, (2020).
[7] H.-J. Lee, D. Lee, and J. Kim. "Event diagnosis method for a nuclear power plant using meta-learning," Nuclear Engineering and Technology, 56(6), pp. 1989-2001, (2024).
[8] H. J. Lee, D. Lee, and J. Kim. "Anomaly Recovery Algorithm Based on Robust AI Concept for Nuclear Power Plants," in Proc. 13th NPIC&HMIT, pp. 1346-1355, (2023).
[9] A. Gong, Y. Chen, J. Zhang, and X. Li. "Possibilities of reinforcement learning for nuclear power plants: Evidence on current applications and beyond," Nuclear Engineering and Technology, 56(6), pp. 1959-1974, (2024).
[10] D. Lee, A. M. Arigi, and J. Kim. "Algorithm for Autonomous Power-Increase Operation Using Deep Reinforcement Learning and a Rule-Based System," IEEE Access, 8, pp. 196727-196746, (2020).
[11] J. Bae, J. M. Kim, and S. J. Lee. "Deep reinforcement learning for a multi-objective operation in a nuclear power plant," Nuclear Engineering and Technology, 55(9), pp. 3277-3290, (2023).
[12] L. Tunkle, K. Abdulraheem, L. Lin, and M. I. Radaideh. "Nuclear microreactor transient and load-following control with deep reinforcement learning," Energy Conversion and Management: X, 27, p. 101090, (2025).
[13] A. Gong, M. Yan, S. Sun, K. Kong, J. Lyu, and X. Li. "One policy to rule them all: Handling multiple emergent accidents in nuclear power plants with ensemble-based behavior cloning," Nuclear Engineering and Technology, p. 103932, (2025).
[14] Md. Kamruzzaman, J. Duan, D. Shi, and M. Benidris. "A Deep Reinforcement Learning-Based Multi-Agent Framework to Enhance Power System Resilience Using Shunt Resources," IEEE Transactions on Power Systems, 36(6), pp. 5525-5536, (2021).
[15] J. Rasmussen. "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models," IEEE Transactions on Systems, Man, and Cybernetics, SMC-13(3), pp. 257-266, (1983).
[16] M. Lind. "Making sense of the abstraction hierarchy in the power plant domain," Cognition, Technology & Work, 5(2), pp. 67-81, (2003).
[17] C. K. Allison and N. A. Stanton. "Constraining Design: Applying the Insights of Cognitive Work Analysis to the Design of Novel In-Car Interfaces to Support Eco-Driving," Automotive Innovation, 3(1), pp. 30-41, (2020).
[18] M. Ghavamzadeh, S. Mahadevan, and R. Makar. "Hierarchical multi-agent reinforcement learning," Autonomous Agents and Multi-Agent Systems, 13(2), pp. 197-229, (2006).
[19] S. Pateria, B. Subagdja, A. Tan, and C. Quek. "Hierarchical Reinforcement Learning: A Comprehensive Survey," ACM Computing Surveys, 54(5), pp. 109:1-109:35, (2021).
[20] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation," in Advances in Neural Information Processing Systems, vol. 29, (2016).

[21] H. M. S. Ahmad, E. Sabouni, A. Wasilkoff, P. Budhraja, Z. Guo, S. Zhang, C. Fan, C. Cassandras, and W. Li. “Hierarchical Multi-Agent Reinforcement Learning with Control Barrier Functions for Safety-Critical Autonomous Systems,” arXiv:2507.14850, (2025).
[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. “Proximal Policy Optimization Algorithms,” arXiv:1707.06347, (2017).
[23] M. Luo, J. Yao, R. Liaw, E. Liang, and I. Stoica. “IMPACT: Importance Weighted Asynchronous Architectures with Clipped Target Networks,” International Conference on Learning Representations (ICLR), (2020).
[24] D. Horgan, et al. “Distributed Prioritized Experience Replay,” arXiv:1803.00933, (2018).
[25] C. Kang, W. Chang, and J. Choi. “Balanced Domain Randomization for Safe Reinforcement Learning,” Applied Sciences, 14(21), p. 9710, (2024).
[26] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. “Sim-to-Real Transfer of Robotic Control with Dynamics Randomization,” in Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 3803-3810, (2018).
[27] S. Huang and S. Ontañón. “A Closer Look at Invalid Action Masking in Policy Gradient Algorithms,” The International FLAIRS Conference Proceedings, 35, (2022).
[28] S. Han, M. Dastani, and S. Wang. “Neuro-symbolic Action Masking for Deep Reinforcement Learning,” arXiv:2602.10598, (2026)