RL for Mitigating Cascading Failures: Targeted Exploration via Sensitivity Factors

Anmol Dwivedi
Rensselaer Polytechnic Institute

Ali Tajer
Rensselaer Polytechnic Institute

Santiago Paternain
Rensselaer Polytechnic Institute

Nurali Virani
GE Vernova Advanced Research

Abstract

Electricity grid's resiliency and climate change strongly impact one another due to an array of technical and policy-related decisions that impact both. This paper introduces a physics-informed machine learning-based framework to enhance grid's resiliency. Specifically, when encountering disruptive events, this paper designs remedial control actions to prevent blackouts. The proposed **P**hysics-**G**uided **R**einforcement **L**earning (PG-RL) framework determines effective real-time remedial line-switching actions, considering their impact on power balance, system security, and grid reliability. To identify an effective blackout mitigation policy, PG-RL leverages power-flow sensitivity factors to guide the RL exploration during agent training. Comprehensive evaluations using the Grid2Op platform demonstrate that incorporating physical signals into RL significantly improves resource utilization within electric grids and achieves better blackout mitigation policies – both of which are critical in addressing climate change.

1 Introduction

Power grid resiliency and climate change are symbiotically interconnected. Climate change is increasing the frequency and intensity of extreme weather events, such as hurricanes, floods, wildfires, and heatwaves, requiring improved grid resiliency to maintain power and reduce economic and societal impacts. Mitigating climate change needs reduction in the energy system's carbon footprint, which critically hinges on integrating renewable resources at scale. However, grid resilience enhancement is needed to provide robustness against equipment failures and manage stability impact of variability from renewable generation. Thus, mitigating and adapting to climate change necessitates enhancing grid resilience. This paper provides a physics-informed machine learning (ML) approach to enhance grid resiliency, defined as the grid's ability to withstand, adapt, and recover from disruptions.

One major source of disruption impacting grid resiliency are transmission line and equipment failures, often caused due to aging infrastructure stressed by extreme weather and congestion due to growing electricity demand. These gradual stresses can lead to system anomalies that can escalate if left unaddressed [1]. To mitigate these risks, system operators implement real-time remedial actions like network topology changes [2, 3, 4, 5]. Selecting these remedial actions must balance two opposing impacts: greedy actions render *quick* impact to protect specific components but may have inadvertent consequences, while look-ahead strategies enhance network robustness but have delayed impact. Striking this balance is crucial for maintaining reliable operation and maximizing grid utilization.

There are two main approaches for the sequential design of real-time remedial decisions: model-based and data-driven. Model-based methods, like model predictive control (MPC), *approximate* the system model and use multi-horizon optimization to predict future states and make decisions [6, 7, 8, 9]. While these methods offer precise control by adhering to system constraints, they require an accurate analytical model, which can be difficult for T-grids. Moreover, coordinating discrete actions like line-switching over extended planning horizons is computationally intensive and time-consuming.

Tackling Climate Change with Machine Learning: workshop at NeurIPS 2024.

Conversely, data-driven approaches like deep reinforcement learning (RL) learn decision policies through sequential interactions with the system model. Deep RL has been successfully applied to various power system challenges [10, 11, 12, 13]. By shifting the computational burden to the offline training phase, these methods allow for rapid decision-making during real-time operations, making them promising for real-time network overload management [14, 15, 16].

Using off-the-shelf RL algorithms (method-driven algorithms [17]) for complex tasks like power-grid overload management presents computational challenges, primarily due to the systems' scale and complexity. Generic exploration policies often select actions that cause severe overloads and blackouts, preempting a comprehensive exploration of the Markov decision process (MDP) state space. This limitation hampers accurate decision utility predictions for the unexplored MDP states, rendering a highly sub-optimal remedial control policy. A solution to circumvent the computational complexity and tractability is leveraging the physics knowledge of the system and incorporating it into RL exploration design.

Contribution: We formalize a **P**hysics-**G**uided **R**einforcement **L**earning (PG-RL) framework for real-time decisions to alleviate transmission line overloads over long operation planning horizons. The framework's key feature is its efficient physics-guided exploration policy design that judiciously exploits the underlying structure of the MDP state and action spaces to facilitate the integration of auxiliary domain knowledge, such as power-flow sensitivity factors [18], for a physics-guided exploration during agent training. Extensive evaluations on Grid2Op [19] demonstrate the superior performance of our framework over counterpart black-box RL algorithms. The data and code required to reproduce our results is publicly available.

Related Work: The study in [20] uses guided exploration based on Q-values while [21] employs policy gradient methods, both on bus-split actions pre-selected via exhaustive search. To accommodate the exponentially many bus-split topological actions, the study in [22] employs graph neural networks combined with hierarchical RL [23] to structure agent training. Recent approaches, such as [24] and [25], focus on integrating domain knowledge via curriculum learning and combining it with Monte-Carlo tree search for improved action selection. However, existing RL approaches (i) focus exclusively on bus-splitting actions; (ii) lack the integration of physical power system signals for guided exploration; and (iii) overlook active line-switching, particularly line *removal* actions, due to concerns about reducing power transfer capabilities and increasing cascading failure risk.

2 Problem Formulation

Transmission grids are vulnerable to stress by adverse internal and external conditions, e.g., line thermal limit violations due to excessive heat and line loading. Without timely remedial actions, this stress can lead to cascading failures resulting in blackouts. To mitigate these risks, our objective is to maximize the system's survival time over a horizon T, denoted by $\mathrm{ST}(T)$, defined as the time until a blackout occurs [19]. In this paper, we focus on line-switching actions $\mathbf{W}_{\mathsf{line}}[n] \triangleq [W_1[n], \dots, W_L[n]]^\top$ to reduce system stress by controlling line flows, where the binary decision variable $W_\ell[n] \in \{0, 1\}$ indicates whether line ℓ is removed (0) or reconnected (1) at time $n \in [T]$. We also define c_ℓ^{line} as the cost of line-switching for line ℓ. Hence, the system-wide cost incurred due to line-switching over a horizon T is $C_{\mathsf{line}}(T) \triangleq \sum_{n=1}^{T} \sum_{\ell=1}^{L} c_\ell^{\mathsf{line}} \cdot W_\ell[n]$.

Operational Constraints: Line-switching decisions are constrained by operational requirements to maintain system security. Once a line is switched, it must remain offline for a mandated downtime period τ_{D} before being eligible for another switch. For naturally failed lines (e.g., due to prolonged overload), a longer downtime period τ_{F} is required before reconnection, where $\tau_{\mathsf{F}} \gg \tau_{\mathrm{D}}$.

Maximizing Survival Time: Our objective is to constantly monitor the system and, upon detecting mounting stress (e.g., imminent overflows), initiate flow control decisions (line-switching) to maximize the system's $\mathrm{ST}(T)$. Such decisions are highly constrained with decision costs $C_{\mathsf{line}}(T)$ and operational constraints due to downtime periods τ_{N} and τ_{F}. To quantify $\mathrm{ST}(T)$, we use a proxy, the **risk margin** for each transmission line ℓ at time n, defined as $\rho_\ell[n] \triangleq \frac{A_\ell[n]}{A_\ell^{\mathsf{max}}}$, where $A_\ell[n]$ and A_ℓ^{max} denotes the present and maximum line current flows, respectively. Based on $\rho_\ell[n]$, a line ℓ is considered *overloaded*, if $\rho_\ell[n] \geq 1$. Minimizing these risk margins reduces the likelihood of overloads, thereby extending $\mathrm{ST}(T)$. We also use risk margins to identify *critical* states, which are states that necessitates remedial interventions, defined by the rule $\max_{i \in [L]} \rho_i[n] \geq \eta$. To maximize $\mathrm{ST}(T)$, our goal is to sequentially form the decisions $\bar{\mathbf{W}}_{\mathrm{line}} \triangleq \{\mathbf{W}_{\mathsf{line}}[n] : n \in \mathbb{N}\}$ all while adhering to operational constraints and controlled decision costs β_{line}, formulated as:

$$\mathcal{P}: \left\{ \begin{array}{cl} \min\limits_{\{\bar{\mathbf{W}}_{\text{line}}\}} & \sum\limits_{n=1}^{T} \sum\limits_{\ell=1}^{L} \rho_\ell[n] \\ \text{s.t.} & C_{\text{line}}(T) \le \beta_{\text{line}} \\ & \text{Operational Constraints} \end{array} \right. . \tag{1}$$

Cascading Failure Mitigation as an MDP: The complexity of identifying optimal line-switching (discrete) decisions grows exponentially with the number of lines L and the target horizon T, and is further compounded by the need to meet operational constraints. To address the challenges of solving $\mathcal{P}$ in (1), we design an agent-based approach. At any instance $n \in [T]$, the agent has access to the system's states $\{\mathbf{X}[m] : m \in [n]\}$ and uses this information to determine the line-switching actions. These actions lead to outcomes that are partly deterministic, reflecting the direct impact on the system state, and partly stochastic, representing the randomness of future electricity demands. To effectively model these stochastic interactions, we employ a Markov decision process (MDP) characterized by the tuple $(\mathcal{S}, \mathcal{A}_{\mathsf{line}}, \mathbb{P}, \mathcal{R}, \gamma)$. Detailed information about the MDP modeling techniques employed is provided in Appendix A.1. Finding an *optimal* decision policy π^* can be found by solving [26]

$$\mathcal{P}_2: \quad \pi^*(\mathbf{S}) \overset{\triangle}{=} \arg\max_{\pi} \; Q_\pi(\mathbf{S}, \pi(\mathbf{S})) , \tag{2}$$

where $Q_\pi(\mathbf{S}, a)$ characterizes the state-action value function.

3 Physics-Guided RL Framework

Motivation: Model-free off-policy RL algorithms [27, 28] with function approximation [29] are effective in finding good policies without requiring access to the transition probability kernel $\mathbb{P}$ for high-dimensional MDP state spaces $\mathcal{S}$. However, the successful design of these algorithms hinges on a comprehensive exploration of the state space to accurately learn the expected decision utilities, such as Q-value estimates. Common approaches entail dynamically updating a behavior policy π, informed by a separate exploratory policy like ϵ-greedy [28], illustrated in Algorithm 1. While Q-learning with *random* ϵ-greedy exploration is effective in many domains [29], it faces challenges in power-grid overload management. Random network topology exploration actions $a[n] \in \mathcal{A}_{\mathsf{line}}$ can quickly induce severe overloads and, thus, blackouts. This is because topological actions force an abrupt change in the system state $\mathbf{X}[n]$ by redistributing transmission line power-flows after a network topological change, compromising risk margins ρ_ℓ and exposing the system to potential cascading failures, preventing a comprehensive exploration of $\mathcal{S}$. This results in inaccurate Q-value predictions for the unexplored MDP states, rendering a highly sub-optimal remedial control policy.

Algorithm 1 Canonical ϵ-greedy Exploration

```
Input: $\epsilon_1, \mathcal{A}, Q(s,a)$,    Output: Action $a$
if $\mu \sim \mathcal{U}(0,1) < \epsilon_1$ then
    $a \sim$ Uniform($\mathcal{A}$)            ▷ Random-Explore
else                                         ▷ $Q$-guided Exploit
    Select $a$ based on $Q(s,a')$
end if
```

Algorithm 2 Physics-Guided ϵ-greedy Exploration

```
Input: $\epsilon_1, \epsilon_2, \mathcal{A}, Q(s,a)$        Output: Action $a$
if $\mu \sim \mathcal{U}(0,1) < \epsilon_1$ then
    if $\zeta \sim \mathcal{U}(0,1) < \epsilon_2$ then          ▷ Physics-Explore
        $a \sim$ Physics-Guided($\mathcal{A}$)                ▷ Algorithm 4
    else
        $a \sim$ Uniform($\mathcal{A}$)                       ▷ Random-Explore
    end if
else                                                       ▷ $Q$-guided Exploit
    Select $a$ based on $Q(s,a')$
end if
```

Sensitivity Factors: We leverage power-flow *sensitivity factors* to guide exploration decisions by augmenting ϵ-greedy during agent training, as illustrated in Algorithm 2. Sensitivity factors [18] help express the mapping between MDP states $\mathcal{S}$ and actions $\mathcal{A}$ by linearizing the system around the current operating point. This approach allows us to analytically approximate the impact of any action $a[n] \in \mathcal{A}$ on risk margins and, consequently, the MDP reward $r \in \mathcal{R}$. To address the challenges associated with implementing random topological actions during ϵ-greedy exploration, we use line outage distribution factors (LODF) to analyze the effects of line removals. Specifically, the sensitivity factor matrix $\mathsf{LODF} \in \mathbb{R}^{L \times L}$, represents the impact of removing line k on the flow in line ℓ by [18]

$$F_\ell[n+1] \approx F_\ell[n] + \mathsf{LODF}_{\ell,k}[n] \cdot F_k[n] , \tag{3}$$

Action Space ($\|\mathcal{A}\|$)	Agent Type	Avg. ST	% Do-nothing	% Reconnect	% Removals	Avg. Action Diversity
–	Do-Nothing	4733.96	100	–	–	–
$\mathcal{A}_{\mathsf{line}}$ (60)	Re-Connection	4743.87	99.90	0.10	–	1.093 (1.821%)
$\mathcal{A}_{\mathsf{line}}$ (119)	`milp_agent`[31]	4062.62	12.05	1.70	86.24	6.093 (5.12%)
$\mathcal{A}_{\mathsf{line}}$ (119) $\mu_{\mathsf{line}} = 0$	$\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(0)$	5929.03	26.78	5.85	67.35	13.406 (11.265%)
	PG-RL [$\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(0)$]	**6657.09**	1.74	7.66	90.59	**17.062** (**14.337**%)
$\mathcal{A}_{\mathsf{line}}$ (119) $\mu_{\mathsf{line}} = 1$	$\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(1)$	5327.06	81.51	0.28	18.20	3.625 (3.046%)
	PG-RL [$\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(1)$]	**6603.56**	13.93	7.00	79.06	**17.156** (**14.416**%)
$\mathcal{A}_{\mathsf{line}}$ (119) $\mu_{\mathsf{line}} = 1.5$	$\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(1.5)$	4916.34	92.69	0.01	7.28	3.406 (2.862%)
	PG-RL [$\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(1.5)$]	**6761.34**	46.53	6.12	47.34	**15.718** (**13.208**%)

Table 1: Performance on the Grid2Op 36-bus system with $\eta = 0.95$.

where $F_k[n]$ is the pre-outage flow in line k, helping predict the anticipated impact of line removal action k. Likewise, the sensitivities of line flows to line reconnection actions are derived in [30].

Physics-Guided Exploration: We leverage sensitivity factors to guide agent exploration with the following key idea: Topological actions $a[n] \in \mathcal{A}_{\mathsf{line}}$ that **reduce** line flows below their limits A_ℓ^{max}, without causing overloads in **other healthy** lines, help transition to more favorable MDP states in the short term, that may otherwise be challenging to reach by taking a sequence of random exploratory actions. However, removing a line k can both reduce flow in some lines and increase flow in others. To address this, we focus on identifying remedial actions that **minimize** flow in the **maximally** loaded line. At time n, we define the maximally loaded line index $\ell_{\mathsf{max}} \overset{\triangle}{=} \arg\max_{\ell \in [L]} \rho_\ell[n]$. By leveraging the structure of the $\mathsf{LODF}[n]$ matrix, we first design Algorithm 3 to identify an effective set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ of potential remedial actions $a[n] \in \mathcal{A}_{\mathsf{line}}$ that greedily **reduce** risk margin $\rho_{\ell_{\mathsf{max}}}[n]$. Then, the agent selects an action $a[n] \in \mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$, guided by the dynamic effective set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$, as outlined in Algorithm 4, for action selection during agent training (as per the PG-RL design in Algorithm 2).

4 Experiments

To demonstrate our framework, we use the Grid2Op 36-bus and the IEEE 118-bus power networks from Grid2Op [19]. Detailed descriptions of the Grid2Op dataset, environment, and performance metrics are in Appendix A.3. We train RL agents with a dueling NN architecture [32] with prioritized experience replay [33] and ϵ-greedy exploration. Appendix A.4 provides a thorough description of the baselines. Table 1 compares the agent's survival time ST(T), averaged across all test episodes for $T = 8062$, showing increased agent sophistication as we move down the table. We denote the best policy from random ϵ-greedy (Algorithm 1) as $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(\mu_{\mathsf{line}})$ and from physics-guided ϵ-greedy (Algorithm 2) by $\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(\mu_{\mathsf{line}})$. For fair comparisons, DQN$_{\boldsymbol{\theta}}$ models for each μ_{line} (5) are trained independently using Algorithms 1 and 2 for 20 hours, using identical hyperparameters listed in Appendix A.5. We also adopt an exponential decay schedule for ϵ_1 while fix $\epsilon_2 = 1$ in Algorithm 2.

In Table 1, we observe that policy $\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(0)$ achieves an average ST of 6,657.09, a 12.2% improvement over $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(0)$ and a 25.2% increase over baselines. Notably, the physics-guided agent takes 25.05% more line-switch actions than its random counterpart, successfully identifying more effective line-removal actions due to the targeted design of $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ during agent training. To illustrate this effectiveness, Fig. 2 plots the number of agent-MDP interactions as a function of agent training time for $\mu_{\mathsf{line}} = 0$. We observe that the PG-RL design results in a greater number of agent-MDP interactions, indicating a more thorough exploration of the MDP state space for the same computational budget.

The ability of $\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}$ to identify more effective actions, in comparison to $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}$, is further substantiated by incrementally increasing μ_{line} and observing the performance changes. As μ_{line} increases, the reward $r[n]$ in (5) becomes *less* informative about potentially effective actions due to the increasing penalties on line-switch actions, thus amplifying the importance of physics-guided exploration design. This is observed in Table 1 where unlike the policy $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(\mu_{\mathsf{line}})$, the ST associated with $\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}(\mu_{\mathsf{line}})$ does not degrade as μ_{line} increases. It is noteworthy that despite the inherent linear approximations of sensitivity factors, confining the RL exploration to actions derived from the set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ enhances state space exploration. Overall, the agent's ability to identify impactful topological actions, leading to greater action diversity, contributes to the enhanced utilization of the electrical grid while also a significant increase in ST. Similar results for the IEEE 118-bus system are provided in Appendix A.6, confirming the trends observed in the Grid2Op 36-bus system.

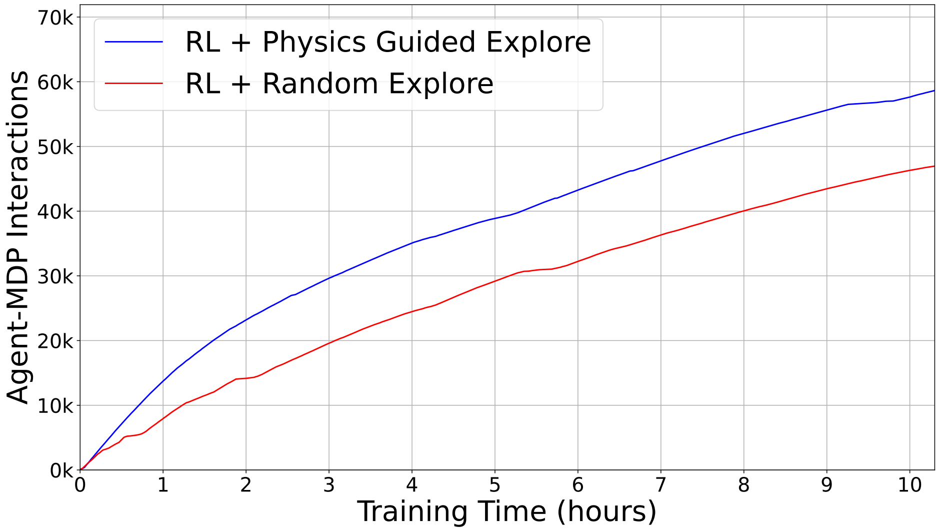

Figure 1: Agent-MDP interactions for the Grid2Op 36-bus system with $\eta = 0.95$ and $\mu_{\text{line}} = 0$.

5 Conclusion and Future Work

We introduced a physics-guided RL framework for determining effective sequences of real-time remedial control actions to mitigate cascading failures. The approach, focused on transmission line-switches, utilizes linear sensitivity factors to enhance RL exploration during agent training. By improving sample efficiency and yielding superior remedial control policies within a constrained computational budget, our framework ensures better utilization of grid resources, which is critical in the context of climate change adaptation and mitigation. Comparative analyses on the Grid2Op 36-bus and the IEEE 118-bus networks highlight the superior performance of our framework against relevant baselines. Future work will involve using bus-split sensitivity factors [34] to computationally efficiently prune and identify effective bus-split actions for remedial control policy design. Another direction is to leverage the linearity of sensitivity factors to implement simultaneous remedial actions, expediting line flow control along desired trajectories.

References

[1] August 14, 2003 blackout: NERC actions to prevent and mitigate the impacts of future cascading blackouts. `https://www.nerc.com/docs/docs/blackout/NERC_Final_Blackout_Report_07_13_04.pdf`, February 2014.

[2] Emily B. Fisher, Richard P. O'Neill, and Michael C. Ferris. Optimal transmission switching. *IEEE Transactions on Power Systems*, 23(3):1346–1355, 2008.

[3] Amin Khodaei and Mohammad Shahidehpour. Transmission switching in security-constrained unit commitment. *IEEE Transactions on Power Systems*, 25(4):1937–1945, 2010.

[4] J. David Fuller, Raynier Ramasra, and Amanda Cha. Fast heuristics for transmission-line switching. *IEEE Transactions on Power Systems*, 27(3):1377–1386, 2012.

[5] Payman Dehghanian, Yaping Wang, Gurunath Gurrala, Erick Moreno-Centeno, and Mladen Kezunovic. Flexible implementation of power system corrective topology control. *Electric Power Systems Research*, 128:79–89, 2015. ISSN 0378-7796.

[6] Mats Larsson, David J. Hill, and Gustaf Olsson. Emergency voltage control using search and predictive control. *International Journal of Electrical Power & Energy Systems*, 24(2):121–130, 2002.

[7] Juliano S. A. Carneiro and Luca Ferrarini. Preventing thermal overloads in transmission circuits via model predictive control. *IEEE Transactions on Control Systems Technology*, 18(6): 1406–1412, 2010.

[8] Mads R Almassalkhi and Ian A Hiskens. Model-predictive cascade mitigation in electric power systems with storage and renewables—Part I: Theory and implementation. *IEEE Transactions on Power Systems*, 30(1):67–77, 2014.

[9] Mads R Almassalkhi and Ian A Hiskens. Model-predictive cascade mitigation in electric power systems with storage and renewables—Part II: Case-Study. *IEEE Transactions on Power Systems*, 30(1):78–87, 2014.

[10] D. Ernst, M. Glavic, and L. Wehenkel. Power systems stability control: reinforcement learning framework. *IEEE Transactions on Power Systems*, 19(1):427–435, 2004.

[11] Jun Yan, Haibo He, Xiangnan Zhong, and Yufei Tang. Q-learning-based vulnerability analysis of smart grid against sequential topology attacks. *IEEE Transactions on Information Forensics and Security*, 12(1):200–210, 2017.

[12] Jiajun Duan, Di Shi, et al. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Transactions on Power Systems*, 35(1):814–817, 2020.

[13] Anmol Dwivedi and Ali Tajer. GRNN-based real-time fault chain prediction. *IEEE Transactions on Power Systems*, 39(1):934–946, 2024.

[14] Adrian Kelly, Aidan O'Sullivan, Patrick de Mars, and Antoine Marot. Reinforcement learning for electricity network operation. *arXiv:2003.07339*, 2020.

[15] Antoine Marot, Benjamin Donnot, Camilo Romero, Balthazar Donon, Marvin Lerousseau, Luca Veyrin-Forrer, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. *Electric Power Systems Research*, 189:106635, 2020.

[16] Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. Learning to run a power network challenge: A retrospective analysis. In *Proc. NeurIPS Competition and Demonstration Track*, December 2021.

[17] David Rolnick, Alan Aspuru-Guzik, Sara Beery, Bistra Dilkina, Priya L. Donti, Marzyeh Ghassemi, Hannah Kerner, Claire Monteleoni, Esther Rolf, Milind Tambe, and Adam White. Application-driven innovation in machine learning. *arXiv:2403.17381*, 2024.

[18] Allen J Wood, Bruce F Wollenberg, and Gerald B Sheblé. *Power Generation, Operation, and Control*. John Wiley & Sons, 2013.

[19] Benjamin Donnot. Grid2Op - A Testbed Platform to Model Sequential Decision Making in Power Systems, 2020. URL `https://github.com/rte-france/grid2op`.

[20] Tu Lan, Jiajun Duan, Bei Zhang, Di Shi, Zhiwei Wang, Ruisheng Diao, and Xiaohu Zhang. AI-based autonomous line flow control via topology adjustment for maximizing time-series ATCs. In *Proc. IEEE Power and Energy Society General Meeting*, QC, Canada, August 2020.

[21] Anandsingh Chauhan, Mayank Baranwal, and Ansuma Basumatary. PowRL: A reinforcement learning framework for robust management of power networks. In *Proc. AAAI Conference on Artificial Intelligence*, Washington, DC, June 2023.

[22] Deunsol Yoon, Sunghoon Hong, Byung-Jun Lee, and Kee-Eung Kim. Winning the L2RPN challenge: Power grid management via semi-Markov afterstate actor-critic. In *Proc. International Conference on Learning Representations*, May 2021.

[23] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2): 181–211, 1999.

[24] Amarsagar Reddy Ramapuram Matavalam, Kishan Prudhvi Guddanti, Yang Weng, and Venkataramana Ajjarapu. Curriculum based reinforcement learning of grid topology controllers to prevent thermal cascading. *IEEE Transactions on Power Systems*, 38(5):4206–4220, 2023.

[25] Geert Jan Meppelink. A hybrid reinforcement learning and tree search approach for network topology control. Master's thesis, NTNU, 2023.

[26] Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[27] J.N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

[28] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[30] P.W. Sauer, K.E. Reinhard, and T.J. Overbye. Extended factors for linear contingency analysis. In *Proc. Hawaii International Conference on System Sciences*, Maui, Hawaii, January 2001.

[31] François Quentin. MILP-agent, 2022. URL `https://github.com/rte-france/grid2op-milp-agent`.

[32] Ziyu Wang, , Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *Proc. International Conference on Machine Learning*, New York, NY, June 2016.

[33] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *Proc. International Conference on Learning Representations*, San Juan, Puerto Rico, May 2016.

[34] Joost van Dijk, Jan Viebahn, Bastiaan Cijsouw, and Jasper van Casteren. Bus split distribution factors. *IEEE Transactions on Power Systems*, 39(3):5115–5125, 2024.

[35] Anmol Dwivedi, Santiago Paternain, and Ali Tajer. Blackout mitigation via physics-guided RL. *arXiv:2401.09640*, 2024.

A Appendix

A.1 MDP Modeling

State Space $\mathcal{S}$: Based on the system's state $\mathbf{X}[n]$, which captures the line and bus features, we denote the MDP state at time n by $\mathbf{S}[n]$, defined as a moving window of the states of length κ, i.e.,

$$\mathbf{S}[n] \ \triangleq [\mathbf{X}[n-(\kappa-1)], \dots, \mathbf{X}[n]]^\top \ , \tag{4}$$

where the state space is $\mathcal{S} = \mathbb{R}^{\kappa \cdot (L \cdot N + F \cdot H)}$. Leveraging the temporal correlation of demands, decisions based on the MDP state $\mathbf{S}[n]$ help predict future load demands.

Action Space $\mathcal{A}$: We denote the action space by $\mathcal{A} \triangleq \mathcal{A}_{\sf line}$, where $\mathcal{A}_{\sf line}$ is the space of line-switching. Action space $\mathcal{A}_{\sf line}$ includes two actions for each line $\ell \in [L]$ associated with reconnecting and removing it. Besides these $2L$ actions, we also include a *do-nothing* action to accommodate the instances at which (i) the mandated downtime period $\tau_{\rm D}$ makes all line-switch actions operationally infeasible; or (ii) the system's risk $\max_{i \in [L]} \rho_i[n]$ is sufficiently low. This action allows the agent to determine the MDP state at time $n+1$ solely based on the system dynamics driven by changes in load demand $\mathbf{D}[n+1]$.

Stochastic Transition Kernel $\mathbb{P}$: After an action $a[n] \in \mathcal{A}$ is taken at time n, the MDP state $\mathbf{S}[n]$ transitions to the next state $\mathbf{S}[n+1]$ according to an unknown transition probability kernel $\mathbb{P}$ $\mathbf{S}[n+1] \sim \mathbb{P}(\mathbf{S} \mid \mathbf{S}[n], a[n])$ where $\mathbb{P}$ captures the system dynamics influenced by both the random future load demand and the implemented action $a[n] \in \mathcal{A}$.

Reward Dynamics $\mathcal{R}$: To capture the immediate effectiveness of taking an action $a[n] \in \mathcal{A}$ in any given MDP state $\mathbf{S}[n]$, we define an instant reward function

$$r[n] \triangleq \sum_{\ell=1}^{L} \left(1 - \rho_\ell^2[n]\right) - \mu_{\mathsf{line}} \left(\sum_{\ell=1}^{L} c_\ell^{\mathsf{line}} \cdot W_\ell[n] \right) , \tag{5}$$

which is the decision reward associated with transitioning from MDP state $\mathbf{S}[n]$ to $\mathbf{S}[n+1]$, where the constant μ_{line} is associated with the cost constraint β_{line} introduced in (1), respectively. The inclusion of parameter μ_{line} allows us to flexibly model different cost constraints, reflecting diverse economic considerations in power systems. Greater values for the parameter μ_{line} in (5) promote solutions that satisfy stricter cost requirements.

A.2 Algorithmic Details

Algorithm 3 Construct Set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ from Action Space $\mathcal{A}_{\mathsf{line}}$

```
procedure EFFECTIVE SET $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}(\mathcal{A}_{\mathsf{line}})$
    Observe system state $\mathbf{X}[n]$ and construct $\mathcal{L}[n]$
    Initialize $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \leftarrow \emptyset$
    Construct $\mathcal{A}_{\mathsf{line}}^{\mathsf{rem}}[n] \leftarrow \{\ell \in \mathcal{L}[n] : \tau_{\mathrm{D}} = 0 \ \& \ \tau_{\mathrm{F}} = 0\}$        ▷ legal removals
    Construct $\mathsf{LODF}[n] \in \mathbb{R}^{L \times L}$ matrix from $\mathbf{X}[n]$
    Find $\ell_{\mathsf{max}} = \triangleq \arg\max_{\ell \in \mathcal{L}[n]} \ \rho_\ell[n]$
    for line $k$ in $\mathcal{A}_{\mathsf{line}}^{\mathsf{rem}}[n] \backslash \{\ell_{\mathsf{max}}\}$ do        ▷ legal line removals that decrease flow
        Compute $F_{\ell_{\mathsf{max}}}[n+1] \leftarrow F_{\ell_{\mathsf{max}}}[n] + \mathsf{LODF}_{\ell_{\mathsf{max}},k} \cdot F_k[n]$
        if $|F_{\ell_{\mathsf{max}}}[n+1]| \leq F_{\ell_{\mathsf{max}}}^{\mathsf{max}}$ then
            $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \leftarrow \mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \bigcup \{k\}$
        end if
    end for
    for line $k$ in $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ do        ▷ no additional overloads
        for line $\ell$ in $\mathcal{L}[n] \backslash \{\ell_{\mathsf{max}}\}$ do
            Compute $F_\ell[n+1] \leftarrow F_\ell[n] + \mathsf{LODF}_{\ell,k} \cdot F_k[n]$
            if $|F_\ell[n+1]| > F_\ell^{\mathsf{max}}$ then
                $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \leftarrow \mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \backslash \{k\}$
                Break
            end if
        end for
    end for
    Construct $\mathcal{A}_{\mathsf{line}}^{\mathsf{reco}}[n] \leftarrow \{\ell \in \neg\mathcal{L}[n] : \tau_{\mathrm{D}} = 0 \ \& \ \tau_{\mathrm{F}} = 0\}$        ▷ legal reconnect
    $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \leftarrow \mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n] \bigcup \mathcal{A}_{\mathsf{line}}^{\mathsf{reco}}[n]$
    return $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$
end procedure
```

Algorithm 3 has three main steps.

1. The agent constructs a legal action set $\mathcal{A}_{\mathsf{line}}^{\mathsf{rem}}[n] \subset \mathcal{A}_{\mathsf{line}}$ from $\mathbf{X}[n]$, comprising of permissible line removal candidates. Specifically, lines $\ell \in \mathcal{L}[n]$ with legality conditions $\tau_{\mathrm{D}} = 0$ and $\tau_{\mathrm{F}} = 0$ can only be removed rendering other control actions in $\mathcal{A}_{\mathsf{line}}$ irrelevant at time n.
2. A dynamic set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ is constructed by initially identifying lines $k \in \mathcal{A}_{\mathsf{line}}^{\mathsf{rem}}[n] \backslash \{\ell_{\mathsf{max}}\}$ whose removal *decrease* flow in line ℓ_{max} below its rated limit $F_{\ell_{\mathsf{max}}}^{\mathsf{max}}$.
3. Finally, the agent eliminates lines from $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ the removal of which creates additional overloads in the network. Note that we include *all* currently disconnected lines $\ell \in \neg\mathcal{L}[n]$ as potential candidates for reconnection in the set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$, provided they adhere to legality conditions ($\tau_{\mathrm{D}} = 0$ and $\tau_{\mathrm{F}} = 0$). It is noteworthy that the set $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$ is *time-varying*. Hence,

Algorithm 4 Physics-Guided Exploration

```
procedure PHYSICS-GUIDED EXPLORE($\mathcal{A}_{\text{line}}$)
    Construct $\mathcal{R}^{\text{eff}}_{\text{line}}[n]$ from $\mathcal{A}_{\text{line}}$ using Algorithm 3
    Initialize maxReward $\leftarrow -\infty$
    Initialize maxAction $\leftarrow$ None
    for each action $a$ in $\mathcal{R}^{\text{eff}}_{\text{line}}[n]$ do                ▷ get reward estimate
        Obtain reward estimate $\tilde{r}[n]$ for action $a[n]$ via sensitivity factors
        if $\tilde{r}[n] >$ maxReward then
            maxReward $\leftarrow \tilde{r}$
            maxAction $\leftarrow a$
        end if
    end for
    return maxAction
end procedure
```

depending on the current system state $\mathbf{X}[n]$, $\mathcal{R}^{\text{eff}}_{\text{line}}[n]$ may either contain a few elements or be empty.

Algorithm 5 Q-Guided *Exploitation* with Probability $1 - \epsilon_n$

```
procedure $Q$-GUIDED EXPLOIT($\mathcal{A}_{\text{line}}, \boldsymbol{\theta}_n$)
    Infer MDP state $\mathbf{S}[n]$ from $\mathbf{X}[n]$
    Construct $\mathcal{A}^{\text{legal}}_{\text{line}}[n] \leftarrow \{\ell \in [L] : \tau_{\text{D}} = 0 \ \& \ \tau_{\text{F}} = 0\}$      ▷ legal line-switch
    $\mathcal{A}^{\text{legal}} \leftarrow \mathcal{A}^{\text{legal}}_{\text{line}}[n] \bigcup \mathcal{A}_{\text{gen}}$
    Initialize $\mathbf{Q}[n] \leftarrow \text{DQN}_{\boldsymbol{\theta}_n}(\mathbf{S}[n])$
    $\mathbf{Q}_{\mathcal{A}^{\text{legal}}}[n] \leftarrow \text{Filter}(\mathbf{Q}[n], \mathcal{A}^{\text{legal}})$                ▷ filter *legal* $Q$-values
    topFiveActions $\leftarrow \text{TopFive}(\mathbf{Q}_{\mathcal{A}^{\text{legal}}}[n])$             ▷ find top-5 legal $Q$-values
    Initialize maxReward $\leftarrow -\infty$
    Initialize maxAction $\leftarrow$ None
    for each action $a$ in topFiveActions do                          ▷ get reward estimate
        Obtain reward estimate $\tilde{r}[n]$ for action $a$ via flow model (3)
        if $\tilde{r}[n] >$ maxReward then
            maxReward $\leftarrow \tilde{r}[n]$
            maxAction $\leftarrow a$
        end if
    end for
    return maxAction
end procedure
```

Q-Guided Exploitation Policy (Algorithm 5) The agent refines its action choices over time by leveraging the feature representation $\boldsymbol{\theta}_n$, learned through the minimization of the temporal difference error via stochastic gradient descent. Specifically, the agent employs the current $\text{DQN}_{\boldsymbol{\theta}_n}$ to select an action $a \in \mathcal{A}$ with probability $1 - \epsilon_n$. The process begins with the agent inferring the MDP state $\mathbf{S}[n]$ in (4) from $\mathbf{X}[n]$. Next, the agent predicts a $\mathbf{Q}[n] \in \mathbb{R}^{|\mathcal{A}|}$ vector using the network model $\text{DQN}_{\boldsymbol{\theta}_n}(\mathbf{S}[n])$ through a forward pass, where each element represents Q-value predictions associated with each remedial control actions $a[n] \in \mathcal{A}$. Rather than choosing the action with the highest Q-value, the agent first identifies legal action subset $\mathcal{A}^{\text{legal}} \triangleq \mathcal{A}^{\text{legal}}_{\text{line}}[n]$ from $\mathbf{X}[n]$. Next, the agent identifies actions $a[n] \in \mathcal{A}^{\text{legal}}$ associated with the top-5 Q-values within this legal action subset $\mathcal{A}^{\text{legal}}$ and chooses one optimizing for the reward estimate $\tilde{r}[n]$. This policy accelerates learning without the need to design a sophisticated reward function $\mathcal{R}$ that penalizes illegal actions.

A.3 Grid2Op Environment Details

Grid2Op Environment Grid2Op is an open-source gym-like platform for simulating power transmission networks with real-world operational constraints. Grid2Op offers diverse episodes throughout the year with distinct monthly load profiles. Each episode encompasses generation $\mathbf{G}[n]$ and load demand $\mathbf{D}[n]$ set-points for all time steps $n \in [T]$ across every month throughout the year. Each episode represents approximately 28 days with a 5-minute time resolution, based on which we have horizon $T = 8062$. December consistently shows high aggregate demand, pushing transmission lines closer to their maximum flow limits while May experiences relatively lower demand.

Datasets For both systems, we have performed a *random* split of Grid2Op episodes. For the test sets, we selected 32 scenarios for the Grid2Op 36-bus system and 34 scenarios for the IEEE 118-bus system, while assigning 450 scenarios to the training sets and a subset for validation to determine the hyperparameters. To ensure proper representation of various demand profiles, the test set includes at least two episodes from each month.

Performance Metrics: A key performance metric is the agent's survival time $\text{ST}(T)$, averaged across all test set episodes for $T = 8062$. We explore factors influencing ST through analyzing action diversity and track *unique* control actions per episode. Furthermore, we quantify the fraction of times each of the following three possible actions are taken: "do-nothing," and "line-switch $\mathcal{A}_{\text{line}}$,". Since the agent takes remedial actions only under *critical* states associated with critical time instances n, we report action decision fractions that exclusively stem from these critical states, corresponding to instances when $\rho_{\ell_{\text{max}}}[n] \geq \eta$. We also note that monthly load demand variations $\mathbf{D}[n]$ influence how frequently different MDP states $\mathbf{S}[n]$ are visited. This results in varying control actions per episode. To form an overall insight, we report the *average* percentage of actions chosen across all test episodes.

System-State Feature $\mathbf{X}[n]$	Size	Type	Notation
prod_p	G	float	$\mathbf{G}[n]$
load_p	D	float	$\mathbf{D}[n]$
p_or, p_ex	L	float	$F_\ell[n]$
a_or, a_ex	L	float	$A_\ell[n]$
rho	L	float	$\rho_\ell[n]$
line_status	L	bool	$\mathcal{L}[n]$
timestep_overflow	L	int	overload time
time_before_cooldown_line	L	int	line downtime
time_before_cooldown_sub	N	int	bus downtime

Table 2: Heterogeneous input system state features $\mathbf{X}[n]$.

System Parameters and MDP State Space The Grid2Op 36-bus system consists of $N = 36$ buses, $L = 59$ transmission lines (including transformers), $G = 10$ *dispatchable* generators, and $D = 37$ loads. We employ $F = 8$ line and $H = 3$ bus features (Table 2), totaling $O = 567$ *heterogeneous* input system state $\mathbf{X}[n]$ features. Each MDP state $\mathbf{S}[n]$ considers the past $\kappa = 6$ system states for decision-making. Without loss of generality, we set $\eta = 0.95$ specified in Section 2 as the threshold for determining whether the system is critical.

The IEEE 118-bus system consists of $N = 118$ buses, $L = 186$ transmission lines, (including transformers), $G = 32$ *dispatchable* generators, and $D = 99$ loads. While in principle we can choose all the 11 features in Table 2, to improve the computational complexity associated with agent training, we choose a subset of line-related features, specifically, $F = 5$ line features (p_or, a_or, rho, line_status and timestep_overflow). This results in a total of $O = 930$ *heterogeneous* input system state features and consider the past $\kappa = 5$ system states for decision-making. Without loss of generality, we set $\eta = 1.0$.

After performing a line-switch action $a[n] \in \mathcal{A}_{\text{line}}$ on any line ℓ, we impose a mandatory downtime of $\tau_{\text{D}} = 3$ time steps (15-minute interval) for each line $\ell \in [L]$. In the event of natural failure caused due to an overload cascade, we extend the downtime to $\tau_{\text{F}} = 12$ (60-minute interval).

MDP Action Space - Line-Switch Action Space Design $\mathcal{A}_{\text{line}}$: Following the MDP modeling discussed in Section A.1, for the Grid2Op 36-bus system we have $|\mathcal{A}_{\text{line}}| = 119$ $(2L + 1)$ and for the IEEE 118-bus system we have $|\mathcal{A}_{\text{line}}| = 373$ $(2L + 1)$.

A.4 Baseline Agents

For the chosen performance metrics, we consider four alternative baselines: (i) Do-Nothing agent consistently opts for the "do-nothing" action across all scenarios, independent of the system-state $\mathbf{X}[n]$; (ii) Re-Connection agent decides to "re-connect" a disconnected line that greedily *maximizes* the reward estimate $\tilde{r}[n]$ (5) at the current time step n. In cases where reconnection is infeasible due to line downtime constraints or when no lines are available for reconnection, the Re-Connection agent defaults to the "do-nothing" action for that step; (iii) `milp_agent`[31] agent strategically minimizes over-thermal line margins using line switching actions $\mathcal{A}_{\mathrm{line}}$ by formulating the problem as a mixed-integer linear program (MILP); and (iv) RL + Random Explore baseline agent: we employ a DQN$_{\boldsymbol{\theta}}$ network with a *tailored* random ϵ_n-greedy exploration policy during agent training. Specifically, similar to Algorithm 4, the agent first constructs a legal action set $\mathcal{A}_{\mathsf{line}}^{\mathsf{legal}}[n] \triangleq \{\ell \in [L] : \tau_{\mathrm{D}} = 0, \tau_{\mathrm{F}} = 0\}$ from $\mathbf{X}[n]$ at critical times. In contrast to Algorithm 4, however, this agent chooses a *random* legal action in the set $a[n] \in \mathcal{A}_{\mathsf{line}}^{\mathsf{legal}}[n]$ (instead of using $\mathcal{R}_{\mathsf{line}}^{\mathsf{eff}}[n]$). In the Grid2Op 36-bus system, using this random exploration policy, we train the DQN$_{\boldsymbol{\theta}}$ for 20 hours of repeated interactions with the Grid2Op simulator for each $\mu_{\mathsf{line}} \in \{0, 0.5, 1, 1.5\}$. We report results associated with the *best* model $\boldsymbol{\theta}$ and refer to the best policy obtained following this random ϵ_n-greedy exploration by $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}(\mu_{\mathsf{line}})$. Similarly, in the IEEE 118-bus system, we train the DQN$_{\boldsymbol{\theta}}$ model for 15 hours of repeated interactions.

A.5 DQN Architecture and Training

Our DQN architecture features a feed-forward NN with two hidden layers, each having O units and adopting `tanh` nonlinearities. The input layer, with a shape of $|\mathbf{S}[n]| = O \cdot \kappa$, feeds into the first hidden layer of O units, followed by another hidden layer of O units. The network then splits into two streams: an advantage-stream $\mathbf{A}_{\boldsymbol{\theta}}(\mathbf{S}[n], \cdot) \in \mathbb{R}^{|\mathcal{A}|}$ with a layer of $|\mathcal{A}|$ action-size units and `tanh` non-linearity, and a value-stream $V_{\boldsymbol{\theta}}(\mathbf{S}[n]) \in \mathbb{R}$ predicting the value function for the current MDP state $\mathbf{S}[n]$. $\mathbf{Q}_{\boldsymbol{\theta}}(\mathbf{S}[n], \cdot)$ are obtained by adding the value and advantage streams. We penalize the reward function $r[n]$ in (5) in the event of failures attributed to overloading cascades and premature scenario termination ($n < T$). Additionally, we normalize the reward constraining its values to the interval $[-1, 1]$. For the Grid2Op 36-bus system, we use a learning rate $\alpha_n = 5 \cdot 10^{-4}$ decayed every 2^{10} training iterations, a mini-batch size of $B = 64$, an initial $\epsilon = 0.99$ exponentially decayed to $\epsilon = 0.05$ over $26 \cdot 10^3$ agent-MDP training interaction steps and choose $\gamma = 0.99$. Likewise, for the IEEE 118-bus system we use similar parameters with a mini-batch size of $B = 32$. Likewise, for the IEEE 118-bus system we set $\alpha_n = 9 \cdot 10^{-4}$ with a mini-batch size of $B = 32$ and $21 \cdot 10^3$ agent MDP training interaction steps.

A.6 Results for the IEEE 118-bus System

Action Space ($|\mathcal{A}|$)	Agent Type	Avg. ST	% Do-nothing	% Reconnect	% Removals	Avg. Action Diversity
–	Do-Nothing	4371.91	100	–	–	–
$\mathcal{A}_{\mathsf{line}}$ (187)	Re-Connection	2813.64	98.73	1.26	–	1.235 (0.66%)
$\mathcal{A}_{\mathsf{line}}$ (373)	`milp_agent`[31]	4003.85	15.64	0.88	83.46	5.617 (1.505%)
$\mathcal{A}_{\mathsf{line}}$ (373)	RL + Random Explore	4812.88	3.58	20.30	76.08	8.323 (2.231%)
	RL + Physics Guided Explore	**5767.14**	1.86	25.34	72.77	**16.235** (**4.352**%)

Table 3: Performance on the IEEE 118-bus system with $\eta = 1.0$ and $\mu_{\mathsf{line}} = 0$.

All the results for the IEEE 118-bus system are tabulated in Table 3. Starting from the baselines, we observe that the Do-Nothing agent achieves a significantly higher average ST of 4,371 steps, compared to the Re-Connection agent's 2813.64 steps. This observation highlights the importance of strategically selecting *look-ahead* decisions, particularly in more complex and larger networks. Contrary to common assumptions, the Re-Connection agent's greedy approach of reconnecting lines can instead reduce ST, demonstrating that Do-Nothing can be more effective.

Focusing on the line switch action space $\mathcal{A}_{\mathsf{line}}$, we observe that the agent with policy $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}$ survives 4812.88 steps, a 10.1% increase over baselines, by allocating 76.08% to remedial control actions for line removals. More importantly, our physics-guided policy $\pi_{\boldsymbol{\theta}}^{\mathsf{physics}}$ achieves an average ST of 5767 steps, a 31.9% increase over baselines and a 19.2% improvement compared to $\pi_{\boldsymbol{\theta}}^{\mathsf{rand}}$ with greater action diversity. Fig. 2 illustrates the number of agent-MDP interactions as a function of training time, showcasing that the physics-guided exploration is more thorough for a given computational budget.

While this paper focuses on the improvements achieved through effective exploration using action space $\mathcal{A}_{\mathsf{line}}$, further enhancements of the physics-guided design can be realized by extending the action space to generator adjustments, i.e., $\mathcal{A}_{\mathsf{line}} \cup \mathcal{A}_{\mathsf{gen}}$. As presented in the study [35], this extension allows for a richer exploration of the state space. It enables reaching additional states by taking actions $a[n] \in \mathcal{A}_{\mathsf{gen}}$ from states that were originally accessible only via actions $a[n] \in \mathcal{A}_{\mathsf{line}}$, thereby improving downstream performance.

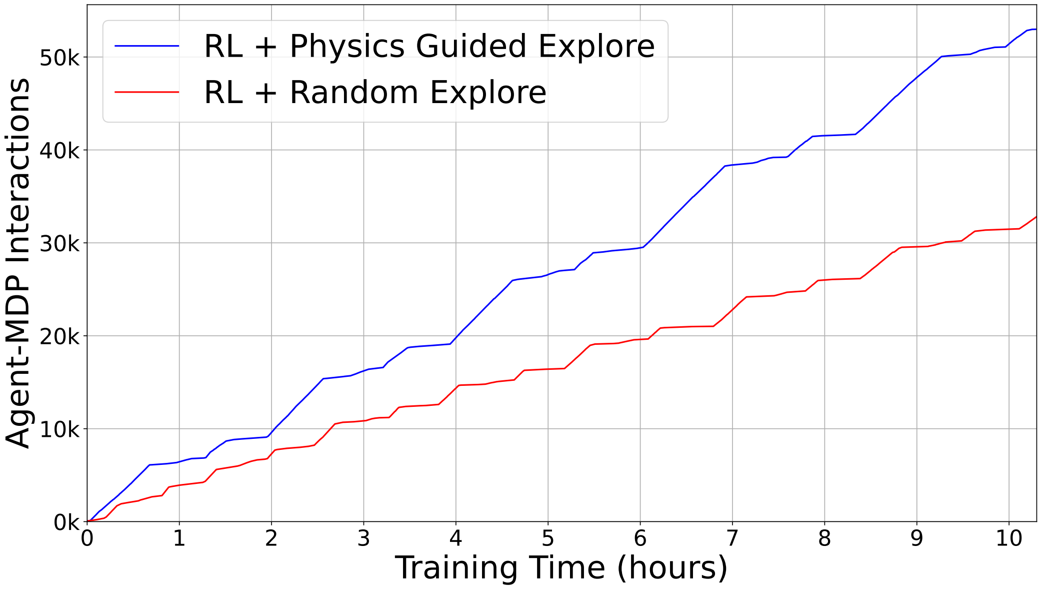

Figure 2: Agent−MDP interactions for the IEEE 118-bus system with $\eta = 1.0$ and $\mu_{\mathsf{line}} = 0$.