RL2Grid: Benchmarking Reinforcement Learning in Power Grid Operations

Anonymous Author(s)
Submission Id: «submission id»

ABSTRACT

Reinforcement learning (RL) has the potential to transform power grid operations by providing adaptive, scalable controllers essential for grid decarbonization and resilience. However, despite their promise, today's RL methods struggle to deal with complex dynamics, aleatoric uncertainty, long-horizon goals, and hard physical constraints, limiting their application in challenging real-world problems. This paper presents RL2Grid, a benchmark developed in collaboration with European power system operators to accelerate progress in grid control and foster the maturity of RL. Our work builds upon Grid2Op, a power grid simulation framework developed by RTE France, to provide standardized tasks, state and action spaces, and rewards within a common interface, presenting a common basis for monitoring and promoting progress. Additionally, we formalize heuristic-guided transitions and safety constraints derived from human operator knowledge and safe practices to reflect grid requirements. We evaluate and compare popular RL baselines across the increasingly complex settings represented within RL2Grid, establishing reference performance metrics and offering insights into the effectiveness of different approaches. Our findings indicate that power grids present substantial challenges for modern RL, underscoring the need for novel methods capable of dealing with complex real-world physical systems.

KEYWORDS

Reinforcement Learning, Power Grids, Benchmark

1 INTRODUCTION

Power grids play a key role in efforts to combat climate change, necessitating a rapid transition to low-carbon energy and improved robustness against climate-induced extremes. This requires power grids to operate under increasing speed, scale, and uncertainty, due in large part to evolving supply and demand profiles resulting from the integration of variable renewable energy sources and distributed devices [12]. This integration creates significant challenges for human operators of power grids [22], further exacerbated by the limitations of traditional power system solvers in addressing realistic systems [3]. Deep reinforcement learning (RL) offers a promising approach to reshaping power grid operations, having demonstrated impressive performance in simulated environments such as Atari and Starcraft [26, 28]. However, there remain many open challenges that impede the practical application of RL in real-world environments, such as dealing with complex dynamics and aleatoric uncertainty, learning long-horizon goals, large combinatorial action spaces, and satisfying hard physical constraints. We argue that *power grids encompass many of these challenges, which are also open research questions in RL*. For these reasons, investigating realistic power grid tasks from an RL perspective could yield substantial benefits for both society and the RL research community. However, progress in relevant RL methodologies is hindered by a lack of standardized benchmarks that can help promote and monitor progress, identify bottlenecks, and develop insights to address real-world challenges.

We fill this gap by introducing RL2Grid, an RL benchmark for realistic power grid operations designed in collaboration with major transmission system operators (TSOs). RL2Grid aims to accelerate progress in grid control and advance RL methods tailored for real-world problems by modeling a diverse, standardized set of increasingly complex power grid environments for RL research that involve dealing with the combinatorially large number of possible actions available in typical grid operations. These tasks build upon RTE France's Grid2Op [30], a realistic power grid simulation framework, and are presented within a standard Gymnasium-based interface, alongside common state and action spaces, and rewards, to provide a shared base for comparison. We additionally perform an in-depth analysis of RL2Grid's design choices by investigating the quality of the actions available in different task settings. To incorporate power grids' real operation practices and hard physical requirements (e.g., satisfying power flows, generator and line limits, as well as avoiding islanding) we also introduce a heuristic module incorporating common line reconnection and idle practices in the grid dynamics [30], as well as constrained task formalizations for safe RL [6]. Finally, by extending the well-known CleanRL library [10] to include flexible configurations for algorithm implementation details, we conduct a comprehensive empirical comparison of classic RL algorithms that are frequently used in the literature as baselines or building blocks for more complex approaches.

Through RL2Grid, we aim to provide a launching point to foster the maturity of RL methods within real-world environments such as power grids, notably by providing realistic tasks that encompass important open questions, and by providing a standardized basis for comparative evaluation and analysis of paths forward. Finally, we discuss important open problems in power grids and their relationship to open problems in RL, as well as highlighting directions for further improving the realism of the power grid simulators, which is a necessary next step to enable last-mile development and deployment of the more general methodological advances we hope RL2Grid will promote.

2 PRELIMINARIES

To clarify the setup, Figure 1 shows a toy example with four substations (dots) interconnected by transmission lines (edges), two generators, and loads connected to buses within each substation. Generators produce power that flows through transmission lines to meet demands (loads). Transmission leads to power losses due

Proc. of the Adaptive and Learning Agents Workshop (ALA 2026), Aydeniz, Delgrange, Mohammedalamen, Yang (eds.), May 25 – 26, 2026, Paphos, Cyprus, https://alaworkshop2026.github.io/. 2026.

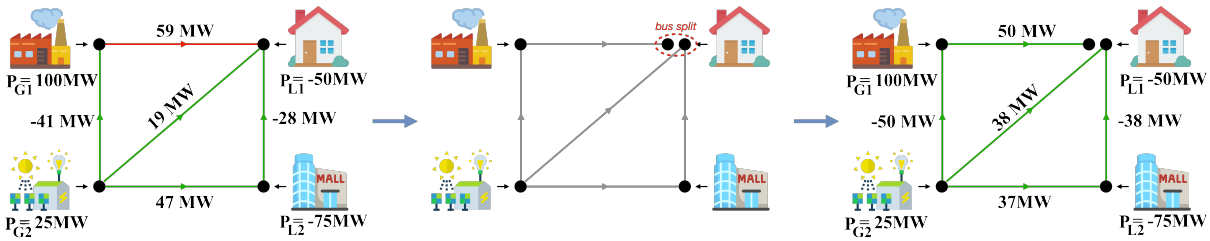

Figure 1: A high-level overview of a power grid action (bus split) to address an overloaded line (red).

to lines' resistive heat, and substations (which may contain multiple buses) can act as "switches" to direct power flows. All these components have physical constraints that must be satisfied (e.g., generators have ramping limits, and transmission lines have maximum capacities, with prolonged overloads causing disconnections and permanent damage).

RL Formalization. A power grid task can be modeled as a Markov decision process (MDP): a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \rho, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the finite sets of states and actions, respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition probability distribution, $\rho : \mathcal{S} \to [0, 1]$ is the initial state distribution, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, and $\gamma \in [0, 1)$ is the discount factor. In policy optimization, agents learn a parameterized policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, modeling the probability of taking an action $a_t \in \mathcal{A}$ in a state $s_t \in \mathcal{S}$ at a certain step t. Agents also typically learn state and/or action value functions V_π and Q_π (Equation 1), representing the expected discounted return when starting from a state s (and action a for Q_π) and then following π. The goal is to find a policy that maximizes the expected discounted return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)]$.

$$
\begin{aligned}
V_\pi(s) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s\right], \\
Q_\pi(s, a) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, a_0 = a\right].
\end{aligned} \tag{1}
$$

To promote safety, we also model grid tasks as constrained MDPs (CMDPs), the most common formalism used in safe RL today [6], by adding a set of constraints $C := \{C_i\}_{i=1,\dots,n}$ defined over unsafe state and action pairs identified by indicator cost functions (a positive value deems a pair as unsafe). These can be used to describe both instantaneous constraints which must be satisfied at every point in time, and cumulative constraints specifying limits on the accumulation of cost over a specified horizon. Although the instantaneous case better describes the safety specifications of common power systems (e.g., satisfying power flows and generator limits), cumulative constraints are often easier to accommodate within existing algorithms [35]. To address these constrained problems, RL approaches typically use an unconstrained Lagrangian optimization formalization $\mathcal{L}$ over the policy parameters using dual variables as $\mathcal{L}^\pi(\boldsymbol{\lambda}) = J_R + \mathcal{L}_C(\boldsymbol{\lambda})$, where J_R is the return objective to maximize, $\mathcal{L}_C(\boldsymbol{\lambda}) = -\sum_{i=1}^{n} \lambda_i (V_{C_i}^\pi - \tau_i)$, $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1,\dots,n}$ act as penalties on J_R for each constraint, $\boldsymbol{\tau} = \{\tau_i\}_{i=1,\dots,n}$ are the constraint thresholds, and $V_C^\pi = \{V_{C_i}^\pi\}_{i=1,\dots,n}$ are the expected cost returns.

3 RL2GRID BENCHMARK

RL2Grid considers the general setting of operating a power grid via topology optimization, as well as redispatch and curtailment actions, in order to keep the grid operational over a long horizon (i.e., a week to a month of operations divided into 5 minute steps):

(i) Topology optimization involves identifying substations where a bus-split action can mitigate the overload by adjusting the grid topology (i.e., how elements are interconnected in the grid). This approach is cost-effective for grid operators as it typically involves simple activation of a switch.[1] However, determining the "optimal" topology from the exponential number of possible configurations is typically infeasible using existing optimization-based solvers.

(ii) Redispatch or curtailment deals with adjusting the power flow by redispatching or curtailing the power output of fossil and renewable power generators (respectively). However, this method is often economically demanding, as it disrupts the normal operations of third parties controlling the generators and can lead to additional power costs.

RL2Grid tasks are designed on top of 7 "base grids" from Grid2Op. Each one has a double bus system, meaning that every electrical component (i.e., generator, load, and transmission line) has two possible connections within a substation. Table 1 summarizes these base grids, along with the features and components they include. These grids present two possible types of contingencies: (i) *Maintenance* (M): Scheduled events observed by the agent where a line is disconnected and cannot be reconnected until a fixed number of steps has passed (a *cooldown*); and (ii) *Opponent* (O): Unforeseen events (e.g., weather conditions) that introduce stochasticity by causing a random line to disconnect, entering its cooldown state. The agent does not know about these events in advance and must address the (potential) resulting contingencies in real time. Environments may also include storage units (*Batteries* (B)) that can act as both generators (discharge) and loads (charge).

Transition dynamics. Each scenario in RL2Grid is defined over time series of synthetic but operationally realistic load demand and generation profiles created with ChroniX2Grid [23]. We use the time series integrated in Grid2Op's base grids, which consists of months to years' worth of data simulating generation and demand profiles happening over extended periods of time. Due to confidentiality and privacy constraints, up-to-date real-world grid time series data is typically unavailable to use. At the beginning of each episode, a random time index is sampled from the full time series

[1] There is some uncertainty (and debate) regarding how frequently each component can be switched safely in practice, without degrading the underlying equipment.

Table 1: List of base grid environments and contingencies currently supported by RL2Grid.

ID	*Maintenance*	*Opponent*	*Battery*	*# Subs.*	*# Lines*	*# Gens.*	*# Loads*
bus14	✓	×	×	14	20	6	11
bus36-M	✓	×	×	36	59	22	37
bus36-MO-{v0, v1}	✓	✓	×	36	59	22	37
bus118-M	✓	×	×	118	186	62	99
bus118-MOB-{v0, v1}	✓	✓	✓	118	186	62	91-99

to initialize the environment, ensuring that agents are trained under a wide variety of grid conditions and do not overfit to specific temporal patterns. From there, at each step, the environment transitions from s_t to s_{t+1} through a multi-stage process that reflects realistic grid operations. First, stochastic (unforeseen) opponent events (e.g., faults caused by weather events) are applied, following distributions specific to each base grid. Agent actions (e.g., topology or redispatch changes) are then executed, and invalid actions (e.g., those violating cooldowns) do not have any effect. The environment updates the cooldown timers and applies any scheduled maintenance events. The underlying Grid2Op power simulation framework then simulates physical power flows using an AC power flow solver; if the system is infeasible due to islanding or demand shortfall, the episode ends. Otherwise, line overloads are identified, and persistent overloads lasting more than three steps trigger automated disconnections. More details on this are discussed by RTE France [30]. The next state is constructed by updating grid variables (e.g., topology, power flows, forecasts), capturing the nonlinear, nonconvex, and stochastic dynamics of real power systems.

State space. Agents observe the state of the power grid at each time step. The state includes common features such as production at each generator, load demands, status, capacity, and cooldown of transmission lines. Additional features are provided based on the environment's characteristics (e.g., maintenance, opponent events, and/or batteries as detailed in Table 1) and the action space. In the topological case, the state includes the topological vector (an integer vector indicating where each device is connected), the connection status of lines, overflow status, and substation cooldowns. In the continuous case, the state consists of target and actual dispatches, curtailment, and generator ramping limits. These are the main features used by the AC power flow solver to transition the grid to the next state. An exhaustive list of the features that comprise the state is discussed in the following.

At a certain time-step t an agent gets the following set of features: $[t, \text{Gen}_P, \text{Gen}_\theta, \text{Load}_P, \text{Load}_\theta, \rho, \text{Cooldown}_{\text{lines}}]$. Additionally, based on the nature of the task, the agent can observe additional features as follows:

- Topological actions: when an agent operates using (discrete) topological actions, it observes $[\text{Topo}_{\text{vect}}, \text{Line}_{\text{status}}, \text{T}_{\text{overflow}}, \text{T}_{\text{sub-cooldown}}]$.
- Redispatching actions: when an agent operates using (continuous) redispatching actions, it observes $[\text{Tg}_{\text{dispatch}}, \text{Curr}_{\text{dispatch}}, \text{Gen}_{\text{margin-up}}, \text{Gen}_{\text{margin-down}}]$.
- Curtailment actions: when an agent operates using (continuous) curtailment actions, it observes $[\text{Gen}_{\text{P}_{\text{curt}}}, \text{Curtail}, \text{Curtail}_{\text{limit}}]$.
- Maintenance: when the task has maintenance contingencies (see Table 1), the agent gets $[\text{Time}_{\text{next-maint}}, \text{Duration}_{\text{next-maint}}]$.
- Storage: when the task has batteries (see Table 1), the agent gets $[\text{Storage}_{\text{charge}}, \text{Storage}_{\text{power}_{\text{tg}}}, \text{Storage}_{\text{power}}, \text{Storage}_\theta]$.

Such a distinction is useful to reduce the size of the space the agent can observe when there are features that are not relevant to a specific task. For example, if an agent uses only discrete actions (topology), then everything related to target dispatch, actual dispatch, and storage is irrelevant as they will not change. Likewise, if an agent uses only continuous actions, it is not necessary to include features related to "topology" as they will not be modified. Additionally, all the features related to voltage (e.g., voltage for generators, loads, . . .) and reactive values (e.g., reactive power for generator, loads, . . .) can be neglected.

For the interested RL practitioner, we refer to the original Grid2Op documentation for exhaustive descriptions of these features [30].

Action spaces. Each grid has two types of tasks, depending on the action space. Table 2 summarizes the total number of actions for each task configuration described in the following.

(i) *Topology space*: Agents take discrete actions that modify the substations topology (i.e., disconnecting or reconnecting a line, or changing the bus to which a component is connected). Notably, human operators currently modify the grid topology manually based on historical behaviors; there is no tractable approach to obtain optimal topology optimization solutions (at scale) as of yet.

Line switching introduces one discrete action per line, whereas bus reassignments (or "bus-splitting") yield an exponentially large number of valid actions depending on the number of elements connected to the substations. Specifically, the topological action space for a double bus substation composed of N_{lines} lines, N_g generators, N_l loads, has size $N = 2^{N_{\text{lines}}+N_g+N_l-1} - 1$ [3]. For instance, substation #5 in Figure 2 has 7 elements, resulting in 63 possible actions, while in the larger *bus36* and *bus118* grids, a single substation can have over 65,000 possible configurations.

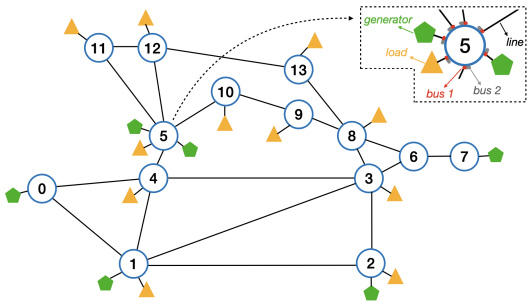

Figure 2: IEEE 14-bus sample grid.

Table 2: Action space sizes for the considered environments. Left: Difficulty for environments with a (discrete) topology-based action space. Right: (continuous) redispatching and curtailment tasks.

	# Actions per difficulty level					
	Topology (T)					**Redispatching and curtailment (R)**
	0	***1***	***2***	***3***	***4***	***0***
bus14	50	209	-	-	-	6
bus36-M	50	302	1829	11071	66978	22
bus36-MO-v0	50	302	1829	11071	66978	22
bus36-MO-v1	50	302	1829	11071	66978	22
bus118-M	50	308	1903	11744	72461	69
bus118-MOB-v0	50	309	1914	11849	73328	69
bus118-MOB-v1	50	309	1915	11852	73357	69

Considering the size of the space, we propose "difficulty levels" in which the action space has an increasing number of topology actions. We selected these action spaces through extensive simulations (72 hours on the computer cluster detailed in Section 4) by ranking the full action space based on the **survival rate** for the grid. This rate represents the number of steps that each action maintains the grid in normal conditions over an episode, and is defined as *the normalized number of steps for which an action does not cause a grid collapse* (because the total demand is not satisfied or parts of the grid become disconnected). In detail, we uniformly sampled topological actions, and after ordering them by highest survival rate, we took the first N_{actions} from the ordered space, where N_{actions} increases with each difficulty level. For example, Figure 3 shows the ranking for the *bus14* grid, highlighting how suitable all the topological actions are to address grid contingencies (and how easier levels contain actions that are more likely to address a contingency). To further motivate our method, we visually analyze the impact of the resultant action spaces in the supplementary material. With these difficulty levels, RL2Grid has a total of 32 topology-based tasks.

(ii) *Redispatching and curtailment space*: Agents take continuous actions changing power generation schedule. Unlike topology actions, which only involve remotely activating a switch, these actions are not free. Economic costs arise from altering the planned schedule of power plants, increased fuel costs, and financial compensation for renewable energy producers. Redispatching actions apply to fossil fuel-based generators, while curtailment actions apply to renewable energy-based generators. Batteries, if present, are also considered generators and come with continuous actions for charging/discharging operations. This action space is relatively tractable for RL algorithms since it involves one continuous action per generator (i.e., $N = N_g$). Thus, we present a total of 7 continuous action-based training environments (one per base grid).

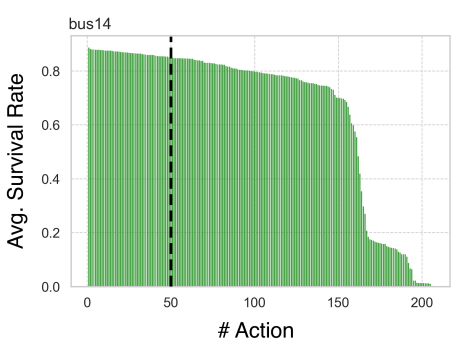

Figure 3: Action ranking for the *bus14* task. The dashed line separates the difficulty levels (i.e., with 50 and 209 actions).

Reward. The reward design is informed by TSOs' mandate to satisfy real-world grid operation requirements: promote long-term safety and efficiency by rewarding grid survival and penalizing unsafe or costly actions (in terms of economic costs). At each step t, the reward an agent gets is $R_t = \alpha R_{\text{survive},t} + \beta R_{\text{overload},t} + \eta R_{\text{cost},t}$, where the weights are evaluated in the supplementary material. Specifically, the agent gets a cumulative positive constant $R_{\text{survive},t}$ for each step, normalized by the total length of a training episode ($\in [0, 1]$). The overload and cost rewards are defined as:

(i) Overload: Penalizes line overloads and disconnections, and rewards available line capacity based on the difference between line flows and capacity limits. In unconstrained settings, disconnected lines incur a fixed penalty. This is more formally defined as:

$$\sum_{\ell \in \mathcal{L}} \left[\max\left(0, \frac{P_{F,\ell,t} - P^{\max}_{F,\ell}}{P^{\max}_{F,\ell} + \epsilon}\right) - \mathbb{1}(\ell \text{ is disc.}) \right], \tag{2}$$

where $P_{F,\ell,t}$ is the power flow on line ℓ at time t, $P^{\max}_{F,\ell}$ is its capacity limit, ϵ is a small constant to avoid divisions by 0, and the indicator function returns 1 if the line is disconnected. This term is then normalized to lie within $[-1, 1]$.

(ii) Cost: Penalizes redispatching or curtailment actions based on deviations from planned dispatch schedules and energy losses. This is defined as: $-\left[\left(P_{G,t} - P_{D,t}\right) + |c_{\text{redisp},t}| + |P_{\text{storage},t}|\right] c_{\text{marginal},t}$, where $P_{G,t}$ and $P_{D,t}$ denote the total power generated and total demand consumed at time t, respectively, with their difference representing transmission losses, $c_{\text{redisp},t}$ corresponds to the redispatched power (i.e., the absolute deviation from scheduled generator setpoints), and $P_{\text{storage},t}$ represents the power exchanged with storage units. All cost components are scaled by the marginal generation cost $c_{\text{marginal},t}$, defined as the cost per MWh of the most expensive generator currently producing power. This value is also normalized to lie in the range $[-1, 0]$.

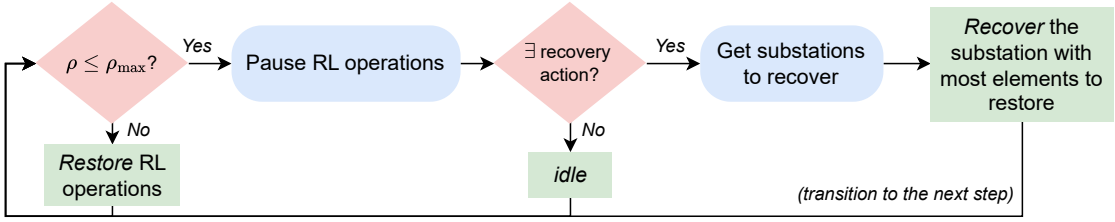

Figure 4: Recovery heuristic combining L2RPN strategies and operator expertise. The agent acts under risk (e.g., line overloads); otherwise, the heuristic incrementally restores the original topology.

3.1 Heuristic-based Transitions

To reduce the problem horizon, RL2Grid incorporates two expert-informed heuristics that modify the transition dynamics described above. These heuristics emulate human operator behavior: they suppress agent actions when the grid is stable and allow the agent to try to recover normal operations when contingencies occur.

The *idle heuristic (I)* is triggered when all line loadings are below a safety threshold set to 95% of their capacity. In this case, the agent's control is suspended and replaced by no-op actions. The *recovery heuristic (R)*, shown in Figure 4, activates when the grid is safe but the topology differs from its original configuration. It incrementally restores the original topology, modifying at most one substation per step, and defaults to idle once recovery is complete.

These heuristics modify the environment so that each agent action initiates a sequence of n-step heuristic-guided transitions during which rewards accumulate. While our heuristics encode operational priors from winning L2RPN competitors [20, 21], our methods have the benefit of being embedded directly in the environment for compatibility with standard RL libraries and are systematically benchmarked. We believe this design balances expert-in-the-loop guidance with the learning process, enabling more sample-efficient training and improved policy performance in realistic, safety-critical scenarios.

3.2 Fostering Safe Operations via Constraints

Prior works such as L2RPN have not incorporated constrained formulations to foster safety [20–22]. RL2Grid introduces CMDP-based (constrained) tasks that reflect two key classes of safety violations faced by system operators as constraints.

In the *load shedding and islanding (LSI)* case, unsatisfied demand and islanding trigger a positive cost. Let us denote the total demand and generation at time t as $P_{D,t}$ and $P_{G,t}$, respectively, and define $L_t = \mathbb{1}(P_{G,t} < P_{D,t}), \quad I_t = 1(N_{I,t} > 0)$, where $N_{I,t}$ is the number of disconnected components. The LSI cost is $C_{\text{LSI}}(t) = L_t + I_t$, with a zero cumulative threshold $\sum_t C_{\text{LSI}}(t) = 0$ to model a hard safety constraint. For *transmission line overload (TLO)*, a positive cost occurs upon thermal overloads and line disconnections. Let us denote the power flow on line ℓ as $P_{F,\ell,t}$, its capacity as $P^{\max}_{F,\ell}$, and define $O_{\ell,t} = \mathbb{1}(P_{F,\ell,t} > P^{\max}_{F,\ell}), \quad D_{\ell,t} = \mathbb{1}(\ell \text{ is disconnected})$, where $D_{\ell,t}$ excludes scheduled maintenance or opponent-driven disconnections. The TLO cost is $C_{\text{TLO}}(t) = \sum_{\ell \in \mathcal{L}}(O_{\ell,t} + D_{\ell,t})$, with a cumulative threshold $\sum_{t=0}^{T} C_{\text{TLO}}(t) < \tau$.

These constraints are applied across all 32 topology-based environments, resulting in 64 additional constrained variants.[2] By modeling these critical safety violations, RL2Grid also enables the development and benchmarking of safe RL methods.

4 EXPERIMENTS

We evaluate the performance of baseline RL algorithms that typically serve as building blocks for more complex algorithms in a subset of RL2Grid tasks that are representative to demonstrate the complexity of power grid operations. In particular, we test: (i) (double) DQN [37], PPO [33], and SAC [7] and their heuristic versions on the discrete topological action space for the *bus14, bus36-MO-v0, bus118-M, bus118-MOB-v0* tasks over most levels of difficulty; (ii) PPO, SAC, and TD3 [4] in the continuous redispatching action space of these environments; and (iii) the Lagrangian version of PPO, LagrPPO [34], in the two constrained versions (LSI and TLO) of the topological *bus14* task.

Experimental Setup. Experiments are performed on Xeon E5-2650 CPU nodes with 256GB of RAM, using CleanRL-based implementations [10] and the hyperparameters detailed in Appendix E, selected via grid search. If not specified, results show the average survival smoothed over 500 episodes of 10 runs per method, with shaded regions representing 95% confidence intervals. As described above, the survival denotes the normalized number of time steps the grid remains operational over an episode. The experiments in this work (excluding the grid search) required >180,000 CPU hours to execute, and Appendix D addresses the associated environmental impact and our efforts to offset estimated CO_2 emissions.

4.1 Results

We indicate with V the "vanilla" baselines, and with I and R the experiments with the heuristics described in Section 3.1. Overall, the RL baselines struggle to deal with the real-world complexities of power grid operations. As expected, we also notice that the heuristic-guided transitions reduce the problem complexity and typically achieve higher performance, despite being not nearly sufficient to operate complex grid setups for long periods of time.

Figure 5 compares the training performance of the baseline algorithms, the heuristic versions, and the constrained variants on the topological *bus14* grid at difficulty level 0. Among the unconstrained baselines, only *PPO successfully learns an effective policy*. However,

[2] Constraints are compatible with redispatching tasks but are primarily evaluated on topology due to their operational relevance and the complexity of associated actions.

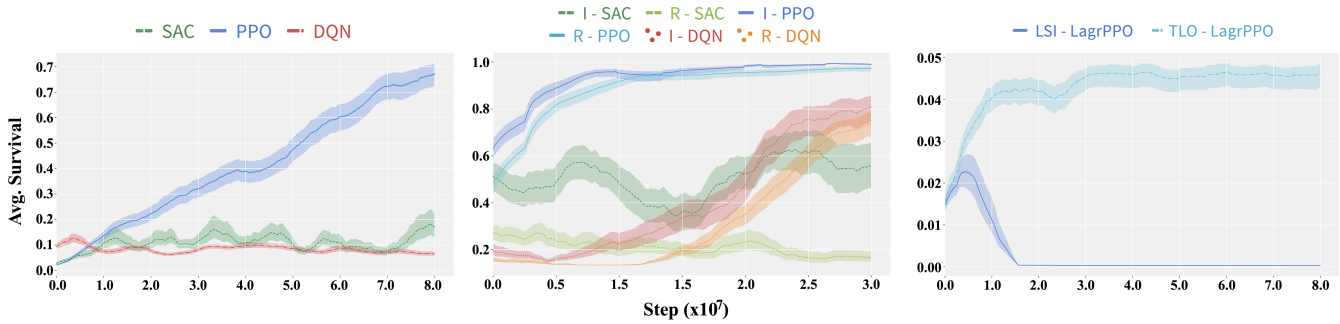

Figure 5: Average survival of the baselines (left), heuristic versions (center), and constrained variants (right) on the bus14 task with topological actions at difficulty 0 (higher values are better).

Table 3: Average survival rate (higher is better) in a subset of difficulty 0 tasks with topological actions for the baselines: idle (*I*), MILP optimization (*MILP*); and RL algorithms: vanilla RL (*V*), and the heuristic versions (*I, R*) for DQN, PPO, and SAC.

Env.	Baseline		DQN			PPO			SAC		
	I	*MILP*	*V*	*I*	*R*	*V*	*I*	*R*	*V*	*I*	*R*
bus14	0.18	0.06	0.07	0.86	0.74	0.74	**0.99**	0.97	0.17	0.56	0.16
bus36-MO-v0	0.03	0.06	0.04	0.14	0.19	0.06	0.17	**0.29**	0.01	0.10	0.13
bus118-MOB-v0	0.10	0.05	0.07	0.19	0.27	0.04	0.18	**0.28**	0.01	0.15	0.19

Table 4: Average survival rate (higher is better) of the grid in a subset of tasks of difficulty 0 with continuous redispatching actions obtained by an idle baseline (*I*) and RL-based vanilla baselines (*V*) for PPO, SAC, and TD3.

Env.	*I*	PPO (*V*)	SAC (*V*)	TD3 (*V*)
bus14	0.00	**0.17**	0.01	0.06
bus36-MO-v0	**0.08**	**0.08**	0.02	0.01
bus118-MOB-v0	0.11	**0.25**	0.08	0.07

incorporating human-informed heuristic operations leads to notable improvements in both performance and sample efficiency across all methods. With heuristic augmentation, PPO achieves good long-term control, while DQN and SAC also exhibit strong performance. Interestingly, our results indicate that *performing idle operations in smaller grid domains (rather than reverting to the original topology) can improve RL performance.* In contrast, introducing constraints significantly increases task difficulty: agents are penalized for violations, and LagrPPO struggles even in this basic setting, failing to learn effective control and frequently exceeding constraint thresholds (see Appendix F for details).

Finally, Tables 3 and 4 report the average survival at convergence for the unconstrained baselines in both topological and redispatching tasks.[3] For topology control, we evaluate two traditional baselines: an "idle" policy (I) and the MILP-based agent from Grid2Op [31], which minimizes line overloads via topological actions under DC power flow approximations.[4] For redispatching, only the idle baseline is considered, as no built-in agent is provided in Grid2Op. Notably, even model-free RL-based agents consistently outperform these traditional methods across both settings.

These performance results motivate the need for further advancements in RL algorithms that can contend with the complex dynamics and aleatoric uncertainty, long-horizon goals, and hard physical constraints of real-world tasks such as power grid operations.

4.2 Performance analysis.

Figure 6 analyzes how the baseline algorithms are learning to control the grid.[5] In particular, we quantify three distinct operational metrics, margin, overloads, and topology, each defined and shown (with 95% confidence intervals) as a scalar-valued "score" signal derived from the agent's interaction with the environment:

- *Margin* (first column): Cumulative available margin across all transmission lines; disconnections are penalized and lower power usage is rewarded. It is defined per-step t as: $\sum_{\ell \in \mathcal{L}} m_{\ell,t}$, where $m_{\ell,t} = \max\left(0, P^{\max}_{F,\ell} - |P_{F,\ell,t}|\right)$, if line ℓ is connected, $m_{\ell,t} = -1$ if not. Higher values indicate that the agent maintains greater flexibility to handle contingencies. We find that successful agents tend to maximize line margins.
- *Overloads* (second column): Shows the overload component $R_{\text{overload},t}$ (see Section 3). Lower values imply that the agent maintains power flows within operational limits; a key characteristic of effective policies. Unsurprisingly, unconstrained agents violate the overload constraints. Perhaps surprisingly,

[3]Due to space constraints, we report average performance without confidence intervals. Full results and training curves, including the *bus118-M* task and higher difficulty levels, are available in the Appendix F.

[4]The approximation is necessary due to the intractability of AC formulations [21].

[5]A similar analysis for the heuristic and constrained experiments is available in Appendix G.

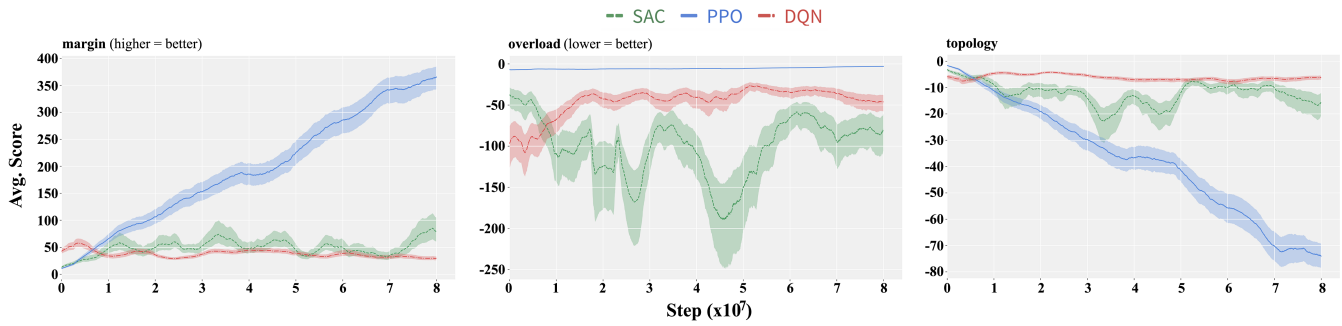

Figure 6: Average score for key operational components across baselines. The first column represents line margins (higher values indicate better contingency management). The second column tracks overload penalties (lower values reflecting improved grid stability). The third column captures topology modifications, showing the extent to which agents reconfigure the grid.

the constrained versions exhibit even more severe violations of these constraints, which we hypothesize is due to their difficulty in optimizing performance in the constrained environment.

- *Topology* (third column): Quantifies deviations from the initial grid configuration (where all elements are connected to the first bus). It is defined per-step t as: $-d(G_t, G_0)$, where G_t denotes the grid topology at time t, and $d(G_t, G_0)$ is the Hamming distance of the topology at time t from the initial topology G_0. This metric quantifies the extent of grid reconfiguration. Negative values indicate significant structural interventions, which often correlate with better policy performance. A value close to 0 suggests minimal changes, while higher values indicate significant topological modifications. We find that successful agents tend to actively reconfigure the grid to optimize operations.

5 RELATED WORK

Sequential decision-making benchmarks for power systems often focus on small-scale problems and/or simplified setups, such as Henri et al. [8] for microgrids, CityLearn for demand response and urban energy management [38], and gym-ANM for small electricity distribution networks [9]. RL environments for electric vehicle (EV) charging and electricity markets have also been introduced [41]. Recently, SustainGym spanned diverse tasks ranging from EV charging to carbon-aware data center job scheduling [40]. The ARPA-E GO Competition provides a larger-scale benchmark, but is geared towards offline optimization [1]. Recent works also consider cascading failure mitigation, demand response optimization, and real-time grid control [11, 25, 36]; however, these works are more geared towards methodological advancements rather than proposing a benchmark. We refer the reader to recent reviews for more RL applications in power grid operations [35].

Relationship with Grid2Op. Grid2Op is an open-source power simulation framework designed by RTE France [30]. It simulates grid operations, requiring that grids work for long horizons in a way that is robust to contingency events, as well as adhering to physical and operational constraints. For the latter, Grid2Op models: (i) cooldown periods to prevent immediate reconnection of disconnected lines, and limits on the frequency of actions on the same line to avoid asset degradation; (ii) limited thermal capacity of transmission lines; (iii) ramp rates on generators that restrict how much power generation can change between time periods; and (iv) adherence to AC power flow constraints. Grid2Op has been primarily used as the base for the L2RPN competitions [19]. While RL methods have been used for L2RPN, they fail to provide a common ground to foster advancements in the field. Each method uses custom input features and action spaces of (very) limited size, often without providing sufficient evidence on how and why these spaces were considered. To date, there is no standardized solution that allows RL researchers to easily get started in this field and compare over established baselines.[6]

6 TACKLING THE CHALLENGES OF POWER GRIDS WITH RL

Applying RL in power grids presents many open problems, each offering significant opportunities for advancing both grid operations and RL methodologies. While we address a subset of these challenges via our work, there remains ample room for future work.

6.1 Relevant RL Methodologies

RL has the potential to be beneficial in addressing open grid problems. There are also potential risks (e.g., with respect to safety, reliability, and robustness) that are important to address. Here, we summarize interesting avenues for future research.

Safe RL. Safety is paramount in power grid operations. Safe RL methods aim to ensure that learning and control policies adhere to strict safety constraints, preventing actions that could lead to blackouts or equipment damage [30]. Ensuring safety while optimizing performance is a critical area of research [5, 24]. In particular, incorporating novel algorithms based on our CMDP representations can be particularly beneficial for ensuring that solutions adhere to physical and operational limits [14, 18].

Human-in-the-loop. Effective grid management requires human expertise and intervention. Incorporating human supervision, interaction, and feedback into RL systems allows for a synergistic approach where human operators and AI work together to optimize grid operations [22]. This collaboration can enhance decision-making and build trust in AI-driven solutions.

[6]Appendix A further discusses the relationship between L2RPN and RL2Grid.

Hierarchical control and multi-agent RL. Power grids operate across multiple hierarchical levels, from individual substations to entire regions. Effective coordination within and across these levels is crucial for maintaining efficient and reliable operations. Hierarchical RL methods can be developed to manage multi-level control tasks, in a way that addresses the scale and complexity of grid operations [2, 29]. Another promising direction is the use of multi-agent representations. Given the vast and distributed nature of power grids, scalability can be enhanced by dividing the grid into distinct areas or agents, each responsible for its own operations. Multi-agent RL (MARL) frameworks can enable these agents to learn and coordinate actions [17, 28], to improve overall grid performance while managing local contingencies more effectively.

Robust RL. The integration of renewable energy sources introduces significant variability and uncertainty into power grids, leading to non-stationary environments. RL algorithms need to adapt to these evolving dynamics to ensure stable and efficient grid operations despite fluctuating supply and demand profiles. Handling non-stationarity is thus a critical research direction [16, 27].

Model-based RL. Model-based RL methods leverage models of the grid dynamics to improve learning efficiency and policy performance. These methods can provide more accurate predictions and better generalize across different scenarios, leading to faster and more robust solutions [15]. Additionally, the AlphaZero algorithm, which combines tree search with deep learning, has shown remarkable success in games like chess and Go and could offer new strategies for handling complex, sequential decision-making tasks with high-dimensional spaces [13].

Better representations. Improving model representations for RL in power grids can also lead to more efficient learning and better policy performance. Leveraging graph neural networks (GNNs) offers a potential avenue for advancement. Power grids can be naturally represented as graphs, with nodes representing buses and edges representing transmission lines. GNNs can effectively model these structures, capturing the spatial and topological dependencies inherent in power grids. Integrating GNNs with RL algorithms can enhance the representation and learning of grid dynamics.

Non-RL approaches. While RL holds great promise, it is also essential to consider non-RL approaches such as optimization solvers, which are relevant particularly for problems with well-defined optimization objectives and constraints. In addition, exploring hybrid methods that combine RL with traditional optimization techniques can yield powerful tools for complex grid management tasks.

6.2 Improving Realism of Environments

It is important to acknowledge that RL2Grid is only a first step. Notably, developing "last-mile" deployable solutions will require further improvements in the realism of power grid environments, which we now discuss.

Scalability. Realistic power systems akin to those managed by RTE France and other transmission system operators may capture hundreds to thousands of buses. To ensure that RL solutions are applicable to real-world scenarios, improving the size and scale of grid environments is essential.

Real data. Grid2Op (and thus, RL2Grid) relies on realistic but synthetic data, which already provide significant challenges for RL. After scaling up RL to deal with the challenges provided by RL2Grid, future environments should (in a way that is cognizant of privacy issues) publicly release real or more realistic synthetic grid data to design to bridge the gap with real power grid operations.

N-1 security. Grid operators must ensure the system can withstand failure of any single component. Rather than modeling failures via random opponents, environments should handle this exhaustively and/or through adversarial agents tailored specifically to the method being tested.

Topology vs. redispatch. Different grid operators handle the relationship between redispatch and topology optimization differently (e.g., some co-optimize these processes, whereas others prefer to handle them separately). Future benchmarks should reflect this heterogeneity in how different power grids are managed. Moreover, Grid2Op's current approach of disconnecting lines after unaddressed overloads does not fully capture real-world practices, where operators attempt to prevent overheating at all costs. Incorporating more realistic consequences for unaddressed overloads, such as system costs, can improve the fidelity of benchmarks. Additionally, grid operators cannot switch every element to every busbar, and there are limits on the number of connected components per substation. Reflecting these constraints can lead to more practical and applicable RL solutions. Storage assets also play an increasingly important role in grid operations. Future benchmarks should accurately model storage and clarify the extent of control grid operators have over these assets.

Phase-shift transformers. These are currently modeled as integer variables in the action space, but should be represented more accurately to reflect their operational impact. Maintenance activities also vary significantly, with Type A involving physical presence at the site and Type B allowing remote interventions. Differentiating these types of maintenance activities in benchmarks can provide a more accurate representation of real-world constraints.

7 CONCLUSIONS

Power grids are essential in combating climate change, requiring a transition to low-carbon energy and enhanced resilience against climate-induced extremes. The integration of VRE sources introduces complexities and uncertainties in grid operations, posing significant challenges for human operators and traditional solvers. Our work aims to foster progress towards these challenges by introducing RL2Grid, a benchmark designed to bridge the gap between current grid management practices and RL research. RL2Grid provides a standardized interface for power grid environments, featuring common rewards, state spaces, action spaces, and safety constraints across a pre-designed set of diverse and complex grid tasks in order to provide a common ground for monitoring and promoting progress. We perform a comprehensive evaluation of the performance of popular baselines on RL2Grid tasks, including versions augmented with domain-informed heuristics aimed at improving performance and sample efficiency, and find that there is still significant room for improvement in the performance of these methods. RL2Grid aims to accelerate algorithmic innovation towards improving power grid operations amidst the evolving challenges posed by climate change.

REFERENCES

[1] ARPA-E. 2023. Grid Optimization (GO) Competition. https://gocompetition.energy.gov/.

[2] Ayhan Alp Aydeniz, Enrico Marchesini, Robert Loftin, and Kagan Tumer. 2023. Entropy Maximization in High Dimensional Multiagent State Spaces. In *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. 92–99.

[3] Anandsingh Chauhan, Mayank Baranwal, and Ansuma Basumatary. 2023. PowRL: A Reinforcement Learning Framework for Robust Management of Power Networks. In *AAAI*.

[4] Scott Fujimoto, Herke van Hoof, and David Meger. 2018. Addressing Function Approximation Error in Actor-Critic Methods. In *International Conference on Machine Learning (ICML)*.

[5] Javier Garcıa and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. In *Journal of Machine Learning Research (JMLR)*.

[6] Shangding Gu, Long Yang, Yali Du, Guang Chen, Florian Walter, Jun Wang, and Alois Knoll. 2024. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[7] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International Conference on Machine Learning (ICML)*.

[8] Gonzague Henri, Avishai Halev Tanguy Levent, Reda Alami, and Philippe Cordier. 2020. pymgrid: An Open-Source Python Microgrid Simulator for Applied Artificial Intelligence Research. arXiv.

[9] Robin Henry and Damien Ernst. 2021. Gym-ANM: Open-source software to leverage reinforcement learning for power system management in research and education. *Software Impacts* 9 (2021).

[10] Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. 2022. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research* 23, 274 (2022), 1–18. http://jmlr.org/papers/v23/21-1342.html

[11] Malte Lehna, Christoph Scholz Jan Viebahn, Antoine Marot, and Sven Tomforde. 2023. Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents. *Energy and AI* (2023).

[12] Yuanzheng Li, Chaofan Yu, Mohammad Shahidehpour, Tao Yang, Zhigang Zeng, and Tianyou Chai. 2023. Deep Reinforcement Learning for Smart Grid Operations: Algorithms, Applications, and Prospects. *Proc. IEEE* 111, 9 (2023), 1055–1096.

[13] Shaohuai Liu, Jinbo Liu, Weirui Ye, Nan Yang, Guanglun Zhang, Haiwang Zhong, Chongqing Kang, Qirong Jiang, Xuri Song, Fangchun Di, et al. 2023. Real-time scheduling of renewable power systems through planning-based reinforcement learning. *arXiv preprint arXiv:2303.05205* (2023).

[14] Yongshuai Liu, Avishai Halev, and Xin Liu. 2021. Policy Learning with Constraints in Model-free Reinforcement Learning: A Survey.

[15] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. 2022. A Survey on Model-based Reinforcement Learning. arXiv:2206.09328 [cs.LG]

[16] Enrico Marchesini and Christopher Amato. 2023. Improving Deep Policy Gradients with Value Function Search. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=6qZC7pfenQm

[17] Enrico Marchesini, Andrea Baisero, Rupali Bhati, and Christopher Amato. 2024. On Stateful Value Factorization in Multi-Agent Reinforcement Learning. arXiv:2408.15381 [cs.AI] https://arxiv.org/abs/2408.15381

[18] Enrico Marchesini, Luca Marzari, Alessandro Farinelli, and Christopher Amato. 2023. Safe Deep Reinforcement Learning by Verifying Task-Level Properties. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

[19] Antoine Marot, Benjamin Donnot, Karim Chaouache, Adrian Kelly, Qiuhua Huang, Ramij-Raja Hossain, and Jochen L Cremer. 2022. Learning to run a power network with trust. *Electric Power Systems Research* 212 (2022), 108487.

[20] Antoine Marot, Benjamin Donnot, Gabriel Dulac-Arnold, Adrian Kelly, Aidan O'Sullivan, Jan Viebahn, Mariette Awad, Isabelle Guyon, Patrick Panciatici, and Camilo Romero. 2021. Learning to run a power network challenge: a retrospective analysis. In *NeurIPS 2020 Competition and Demonstration Track*. PMLR, 112–132.

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

[22] Antoine Marot, Adrian Kelly, Matija Naglic, Vincent Barbesant, Jochen Cremer, Alexandru Stefanov, and Jan Viebahn. 2022. Perspectives on future power system control centers for energy transition. *Journal of Modern Power Systems and Clean Energy* 10, 2 (2022), 328–344.

[23] A. Marot, N. Megel, V. Renault, and M. Jothy. 2020. ChroniX2Grid - The Extensive PowerGrid Time-serie Generator. https://github.com/BDonnot/ChroniX2Grid.

[24] Luca Marzari, Changliu Liu, Priya Donti, and Enrico Marchesini. 2025. Improving Policy Optimization via ϵ-Retrain. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*.

[25] A. R. R. Matavalam, K. P. Guddanti, Y. Weng, and V. Ajjarapu. 2022. Curriculum Based Reinforcement Learning of Grid Topology Controllers to Prevent Thermal Cascading. *IEEE Transactions on Power Systems* (2022).

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. In *Conference on Neural Information Processing Systems (NeurIPS)*.

[27] Janosch Moos, Kay Hansel, Hany Abdulsamad, Svenja Stark, Debora Clever, and Jan Peters. 2022. Robust Reinforcement Learning: A Review of Foundations and Recent Advances. *Machine Learning and Knowledge Extraction* 4, 1 (2022), 276–315.

[28] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Comparative Evaluation of Multi-Agent Deep Reinforcement Learning Algorithms. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

[29] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. 2021. Hierarchical Reinforcement Learning: A Comprehensive Survey. *ACM Comput. Surv.* 54, 5 (jun 2021), 35. https://doi.org/10.1145/3453160

[30] RTE France. 2020. Grid2op: A testbed platform to model sequential decision making in power systems. https://GitHub.com/rte-france/grid2op.

[31] RTE France. 2022. Grid2op MILP agent. https://github.com/Grid2op/grid2op-milp-agent.

[32] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust Region Policy Optimization. In *International Conference on Machine Learning (ICML)*.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. In *arXiv*.

[34] Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. Responsive Safety in Reinforcement Learning by PID Lagrangian Methods. In *International Conference on Machine Learning (ICML)*.

[35] Tong Su, Tong Wu, Junbo Zhao, Anna Scaglione, and Le Xie. 2024. A review of safe reinforcement learning methods for modern power systems. *arXiv preprint arXiv:2407.00304* (2024).

[36] Erica van der Sar, Alessandro Zocca, and Sandjai Bhulai. 2024. Multi-Agent Reinforcement Learning for Power Grid Topology Optimization.

[37] Hado van Hasselt, Arthur Guez, and David Silver. 2016. Deep Reinforcement Learning with Double Q-learning. In *AAAI Conference on Artificial Intelligence*.

[38] Jose R Vazquez-Canteli, Sourav Dey, Gregor Henze, and Zoltan Nagy. 2020. CityLearn: Standardizing Research in Multi-Agent Reinforcement Learning for Demand Response and Urban Energy Management. arXiv.

[39] Wandb. 2025. *Experiment Tracking with Weights and Biases*. https://wandb.ai/site Accessed: 2025-05-15.

[40] Christopher Yeh, Victor Li, Rajeev Datta, Julio Arroyo, Chi Zhang, Yize Chen, Mehdi Hosseini, Azarang Golmohammadi, Yuanyuan Shi, Yisong Yue, and Adam Wierman. 2023. SustainGym: Reinforcement Learning Environments for Sustainable Energy Systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[41] Zidong Zhang, Dongxia Zhang, and Robert C. Qiu. 2020. Deep reinforcement learning for power system applications: An overview. *CSEE Journal of Power and Energy Systems* 6, 1 (2020), 213–225.

A RELATIONSHIP OF RL2GRID TO L2RPN TASKS AND SOLUTIONS

In this section, we clarify the relationship of the tasks presented within RL2Grid, as well as the baseline methods evaluated, to the tasks and solutions presented within the L2RPN competition series.

We remind that our work builds on Grid2Op to provide a benchmark with standardized tasks, state and action spaces, rewards, and a safe (constrained) formalization, as well as comprehensive evaluation of common baselines inspired by L2RPN. These are critical to provide a common basis for assessing advances in RL methods [28] as well as to improve accessibility to RL practitioners who may have limited prior knowledge of power systems.

Tasks. RL2Grid employs all the main Grid2Op "base environments" (which are likewise employed in L2RPN). However, the solutions developed for L2RPN relied on different customized components. Every competition relied on different time series, making effective comparisons far from trivial. For these reasons, on top of the standardization proposed in our work (see Section 3), we have made some underlying changes to the base environments to better reflect the current and future challenges of RL research. Examples include (i) episodes with longer horizons (i.e., an RL2Grid episode models a month of grid operations, ~8000 steps, compared to weekly episodes of most prior work); (ii) making the tasks as uniform as possible (i.e., by integrating curtailment operations in all Grid2Op tasks); (iii) enabling simulation steps inside the Gymnasium interface (a feature added in our code revision, which is not currently available in Grid2Op). These decisions were driven by our goal of ensuring that our benchmark is accessible, standardized, and provides a clear starting point for researchers who may not be familiar with the nuances of these competitions and power grids.

Baselines. Due to the different choices of input features and action spaces considered by different methods submitted to the L2RPN challenges, it was not possible to directly benchmark these specific methods on the RL2Grid tasks. However, the baselines chosen are representative of the methods submitted to past L2RPN competitions, in addition to representing commonly-used methods within the RL community as a whole. In particular, within the L2RPN submissions, a common approach was to incorporate heuristics. These heuristics varied significantly between methods and pushed us to design one that mimicked human operations in real grid operations. We developed this heuristic in collaboration with power system operators who have contributed to our work, incorporating fundamental insights from previous solutions while keeping the focus on standardization and benchmarking.

B RL BASELINES

In this section, we briefly introduce the baseline RL algorithms employed in our evaluation, referring to the original papers for exhaustive details about these methods [4, 7, 26, 33].

DQN [26]. A DQN agent uses a neural network to approximate the action value function Q by taking as input the state of the environment and outputting Q-values for every possible action. During training, the agent uses an ϵ-greedy policy to select random actions or follow the greedy policy on these Q-values, according to a linearly decaying probability ϵ. The Q network is thus updated to minimize the difference between predicted Q-values and a target derived from actual rewards and future Q-values. To deal with overestimation, we use Double-DQN [37] and decouple action selection from action evaluation using a target Q network. Due to its value-based nature, a DQN agent can only consider discrete (topological) actions.

PPO [33] and **Lagrangian PPO** (LagrPPO) [34]. A PPO agent uses its neural network to directly approximate a policy. The agent learns the policy parameters by simplifying the TRPO [32] algorithm, using a computationally tractable clipped objective. This clipping mechanism prevents large changes to the policy that could destabilize the training. At a high level, such a surrogate objective balances policy improvement and limits the divergence between policy updates. To drive the policy training, PPO also learns an advantage function to determine how much better (or worse) taking an action is compared to the expected value. By employing different probability distributions as a policy, a PPO agent can deal with both continuous (redispatching) and discrete (topological) actions. The Lagrangian version applies the same intuitions while learning additional value functions for each constraint. It then changes to policy training by considering the Lagrangian discussed in Section 2. In more detail, Lagrangian algorithms take gradient ascent steps in π and descent steps in $\boldsymbol{\lambda}$ to trade off safety and task performance. These methods focus on satisfying the constraints using penalties $\boldsymbol{\lambda}$ that grows unbounded when constraints are violated. When constraints are satisfied, $\boldsymbol{\lambda}$ scale down (to zero), allowing the algorithm to maximize the task objective.

SAC [7]. Similarly to PPO, a SAC agent learns different networks to maintain a policy and two value functions that mitigate positive bias in value estimates. Overall, the agent maximizes both the expected return and the entropy of the policy. The entropy term encourages exploration by promoting stochastic policies, which helps prevent premature convergence to suboptimal policies. In terms of actions, the SAC agent can deal with the same action types as PPO.

TD3 [4]. A TD3 agent learns multiple networks similarly to SAC. However, unlike the stochastic policies learned by PPO and SAC, TD3 learns a deterministic policy and can only deal with continuous actions. To encourage exploration, the agent does not maximize the entropy of the policy but adds noise to the output of the policy network.

C ENVIRONMENTS

C.1 Action Spaces Analysis

In this section, we visually analyze the action spaces of one representative environment for each power grid size (i.e., bus14, bus36-MO-v0, bus118-M).

For each difficulty level, Figures C.1, C.2, C.3 show the percentage of actions considered for each substation within the action space. The x-axis lists the substation IDs in descending order based on the number of available actions. The y-axis represents the ratio of actions used in the action space to the total number of available actions for each substation. Consequently, the highest difficulty level indicates that the action space includes all possible actions for all substations. Overall, this analysis suggests that the substation with the most electric components (i.e., the most possible topologies) is best suited to handle contingencies.

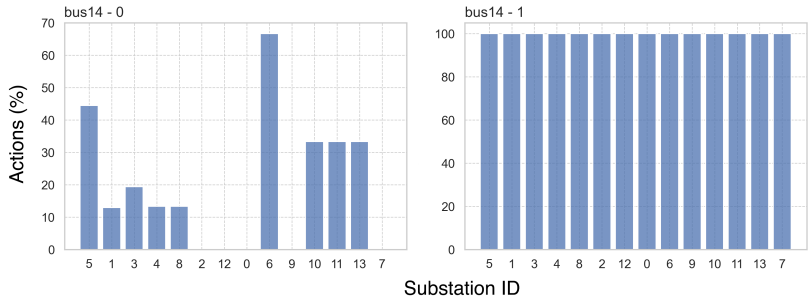

Figure C.1: Percentage of actions considered for each substation within the action space for bus14 (discrete) topological tasks (difficulty level is indicated with the number on the top left).

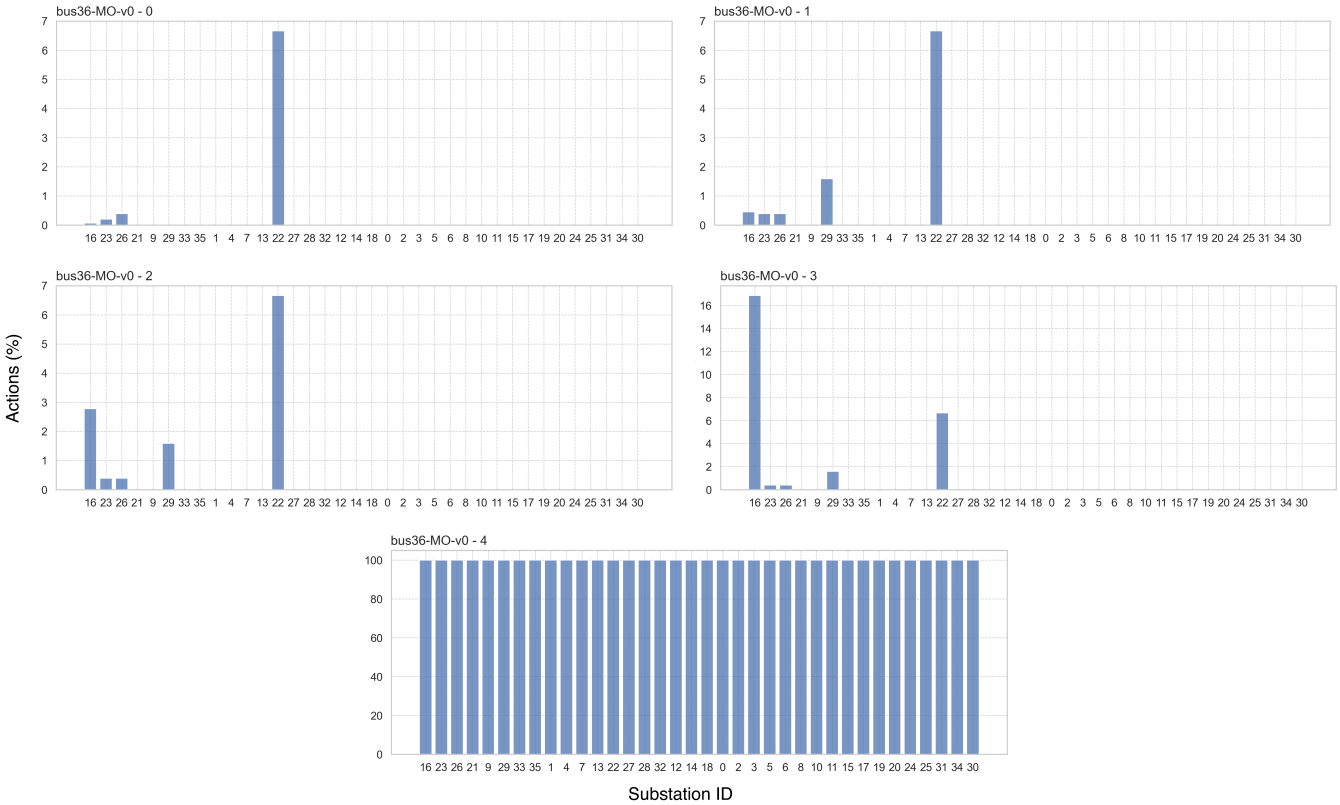

Figure C.2: Percentage of actions considered for each substation within the action space for bus36-MO-v0 (discrete) topological tasks (difficulty level is indicated with the number on the top left).

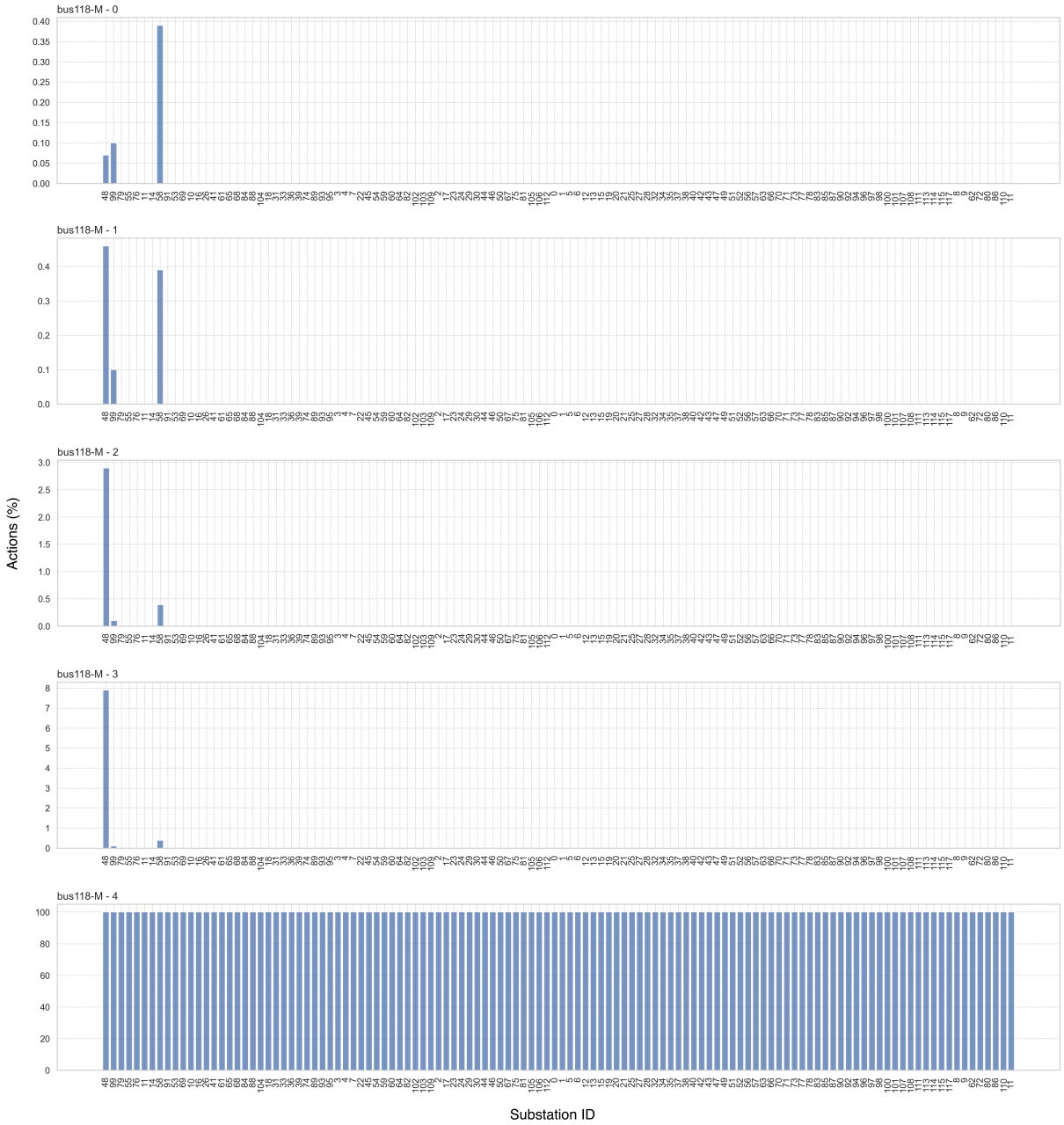

Figure C.3: Percentage of actions considered for each substation within the action space for bus118-M (discrete) topological tasks (difficulty level is indicated with the number on the top left).

Figures C.4, C.5 then present the data collected during the action ranking mechanism described in Section 3.

As a sanity check, Figure C.4 shows an example of the uniform sampling strategy used to select which action to simulate at each simulation step. The x-axis shows the total number of actions for the bus14 (discrete) topological task; the y-axis indicates the number of times each action was sampled during the ranking process.

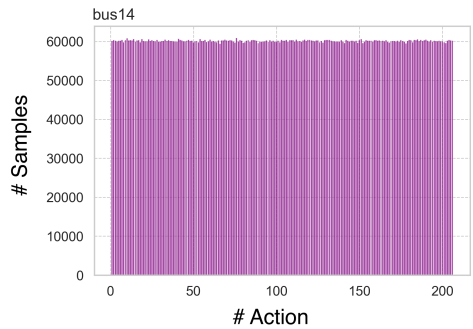

Figure C.4: Number of times each action is sampled over the ranking process time.

Figure C.5 shows the final ranking of the actions for the three representative environments. The x-axis shows the total number of actions for each task; the y-axis indicates the average survival rate of each action during the ranking process. Crucially, most of the actions are relevant (i.e., with a high survival rate) in the tasks, motivating the increasing levels of difficulty we proposed for the (discrete) topological environments.

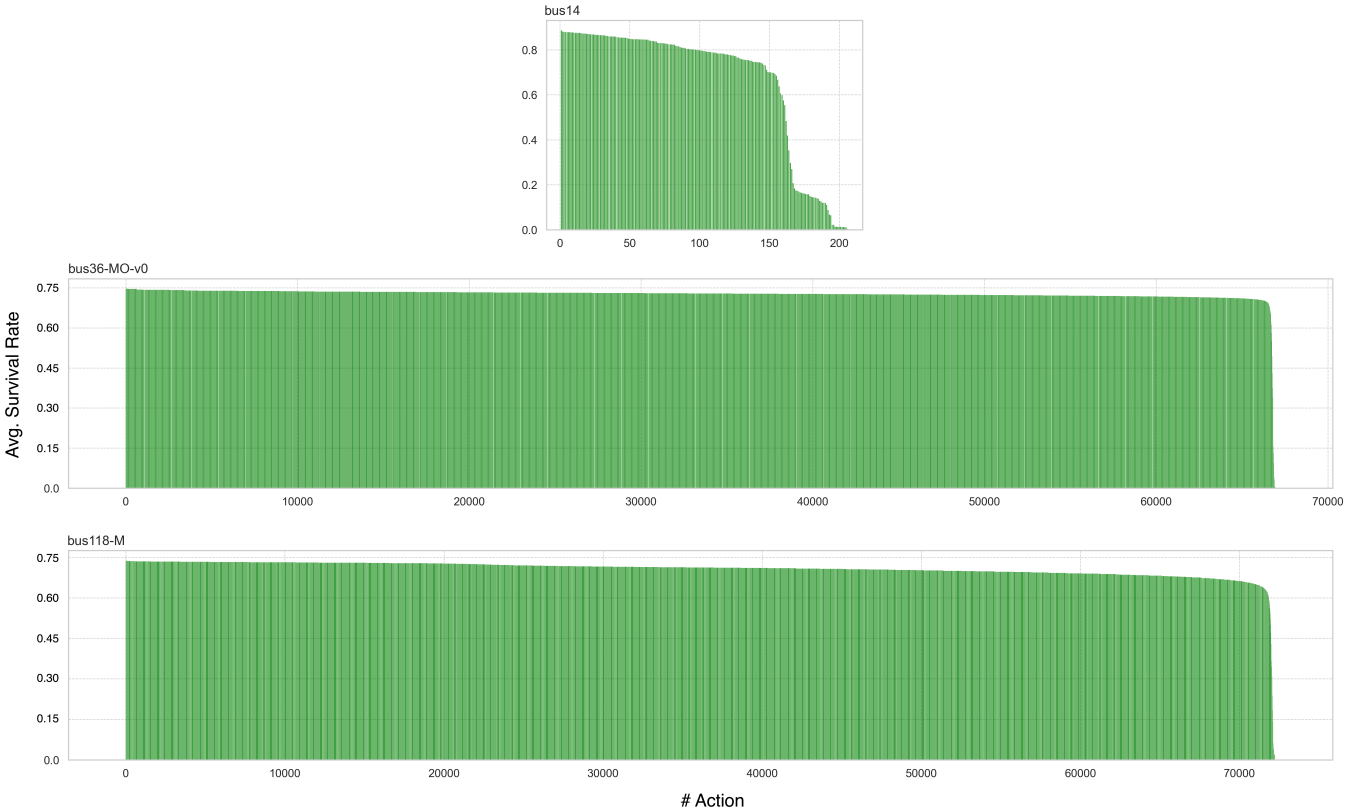

Figure C.5: Average survival rate of the action spaces after the ranking process time.

D ENVIRONMENTAL IMPACT

Despite each training run being "relatively" computationally inexpensive due to the use of CPUs, the experiments of our evaluation led to cumulative environmental impacts due to computations that run on computer clusters for an extended time. Our experiments were conducted using a private infrastructure with a carbon efficiency of $\approx 0.275 \frac{\text{kgCO}_2\text{eq}}{\text{kWh}}$. Total emissions are estimated to be $\approx 216.56\text{kgCO}_2\text{eq}$ using the Machine Learning Impact calculator, and we purchased offsets for this amount through Treedom. We do not directly estimate or offset other categories of environmental impacts (such as water usage or embodied hardware impacts), though recognizing that these are additionally important to consider.

E HYPERPARAMETERS

Table E.1 lists the hyperparameters considered during our initial grid search and the final (best-performing) parameters used for our experiments.

Table E.1: Details of the grid search used to find the best-performing hyperparameters for each algorithm in the topological (T) and redispatching (R) cases.

Algorithm	Parameter	Grid search	Chosen value (T - R)
Shared	*N° parallel envs*	10, 20, 50	50
	Learning starts	20000	20000
	Max gradient norm	10, 20, 50	10
	Discount γ	0.9, 0.95, 0.99	0.9
	α	0.1, 0.5, 1.0	1.0
	β	0.1, 0.5, 1.0	0.5
	η	0.1, 0.25, 0.5	0.5
	λ	0, 50	0 (TLO), 50 (LSI)
DQN	*Train frequency*	50, 100, 1000	50
	Target network update	1000, 5000, 10000	5000
	Buffer size	100000, 250000, 500000, 1000000	1000000
	Batch size	64, 128, 256	128
	Learning rate	0.003, 0.0003, 0.00003	0.0003
	ϵ*-decay fraction*	0.3, 0.5 0.7	0.5
PPO	*N° steps (total)*	10000, 20000, 50000	20000
	N° minibatches	4, 8, 12	4
	N° update epochs	20, 40, 80	40
	Actor learning rate	0.003, 0.0003, 0.00003	0.0003 - 0.00003
	Critic learning rate	0.003, 0.0003, 0.00003	0.0003 - 0.00003
	ϵ*-clip*	0.1, 0.2, 0.3	0.2
SAC	*Train frequency*	50, 100, 1000	50
	Actor delayed update	2, 4	2
	Noise clip	0.5	0.5
	Buffer size	100000, 250000, 500000, 1000000	500000
	Batch size	64, 128, 256	128
	Actor learning rate	0.003, 0.0003, 0.00003	0.0003 - 0.00003
	Critic learning rate	0.003, 0.0003, 0.0003	0.0003 - 0.00003
	Entropy regularization	0.02, 0.2, 0.4	0.2
TD3	*Actor delayed update*	2, 4	2
	Buffer size	100000, 250000, 500000, 1000000	1000000
	Batch size	64, 128, 256	128
	Actor learning rate	0.003, 0.0003, 0.00003	0.0003 - 0.00003
	Critic learning rate	0.003, 0.0003, 0.00003	0.0003 - 0.00003
	τ	0.005, 0.0005	0.005
	Policy noise	0.2	0.2
	Exploration noise	0.1	0.1

F OMITTED FIGURES IN SECTION 5

The following figures report plots collected on Wandb [39]. Figure F.1 shows the training curves for the remaining (discrete) topological action spaces. Due to the time limit imposed on the computation nodes (72 hours) and the different computational requirements of the algorithms, not all the baselines perform the same number of steps in the time limit and the experiments with 36 and 118 bus consider 5 runs. The demands and limited performance of the topological baselines led us to exclude the results with the complete action space (i.e., difficulty set to 4). Additionally, despite the grid search of Table E.1, some baselines achieved lower performance than expected (e.g., SAC and DQN in the *bus14* scenarios).

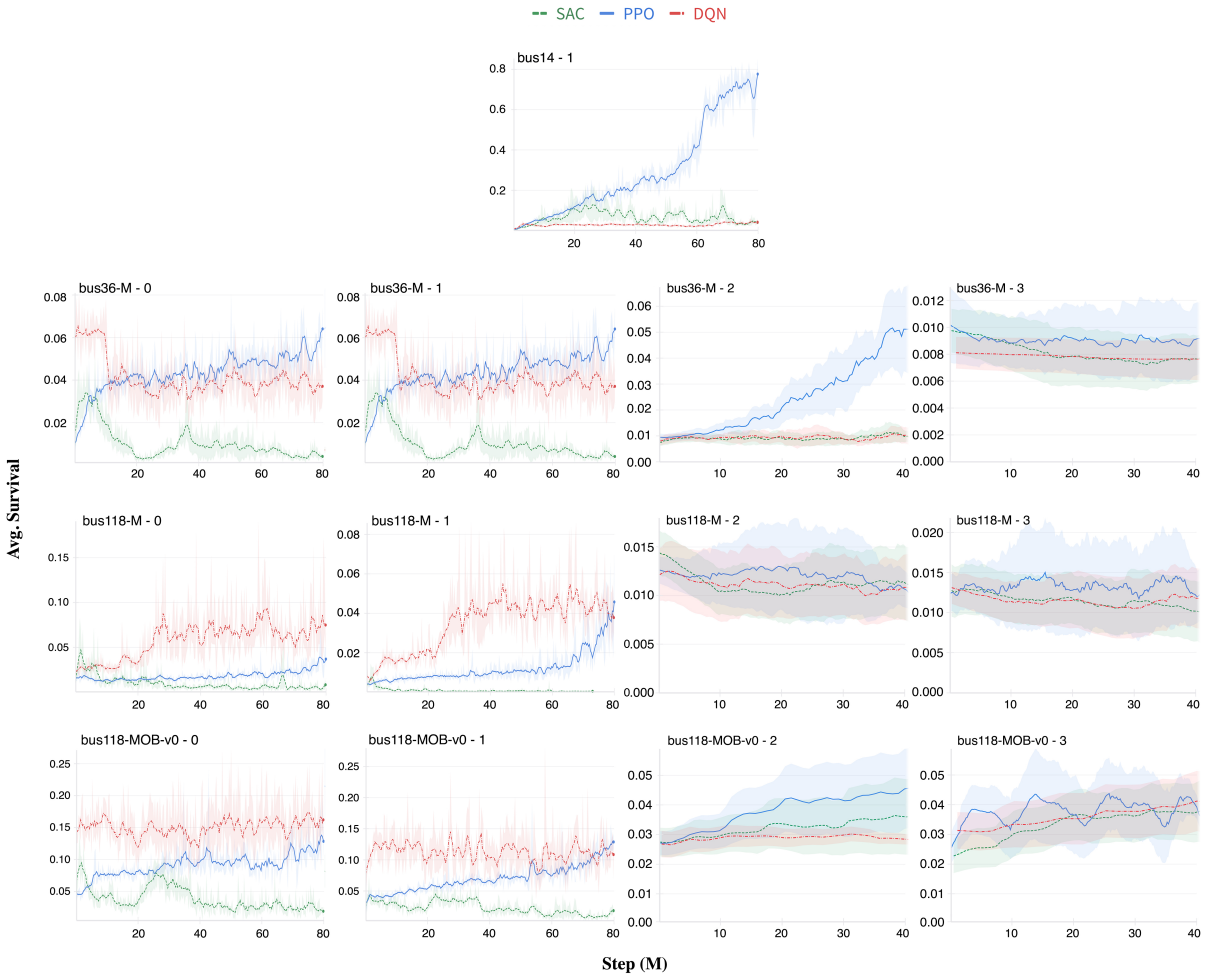

Figure F.1: Average survival rate for the discrete topological case in *bus14, bus36-M, bus118-M, bus118-MOB-v0* using the SAC, PPO, and DQN baselines. We indicate the difficulty level (ranging from 0 to 3) next to the environment identifier.

Figure F.2 shows the training curves for the (continuous) redispatching action spaces.

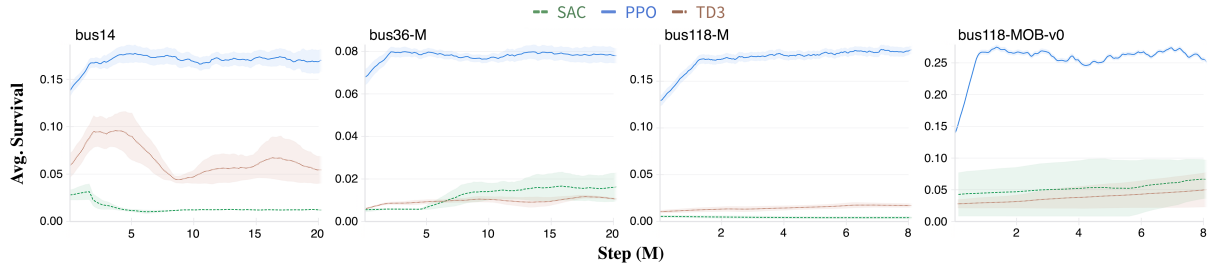

Figure F.2: Average survival rate for the continuous redispatching case in *bus14, bus36-M, bus118-M, bus118-MOB-v0* using the SAC, PPO, and TD3 baselines.

Figure F.3 shows the cost obtained over the training for the constrained experiments.

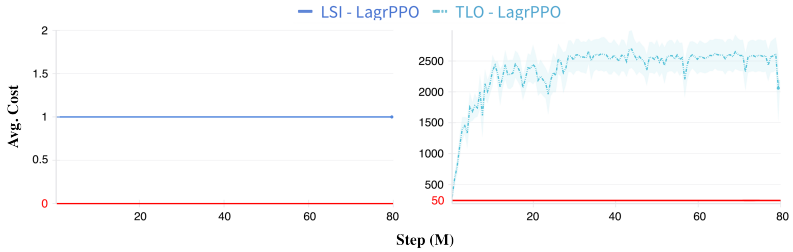

Figure F.3: Average cost rate for constrained case in the representative *bus14* using the LagrPPO baseline. The constrained threshold (red line) is set to 0 and 50 for the TLO and LSI cases, respectively.

G OMITTED PERFORMANCE ANALYSIS IN SECTION 5

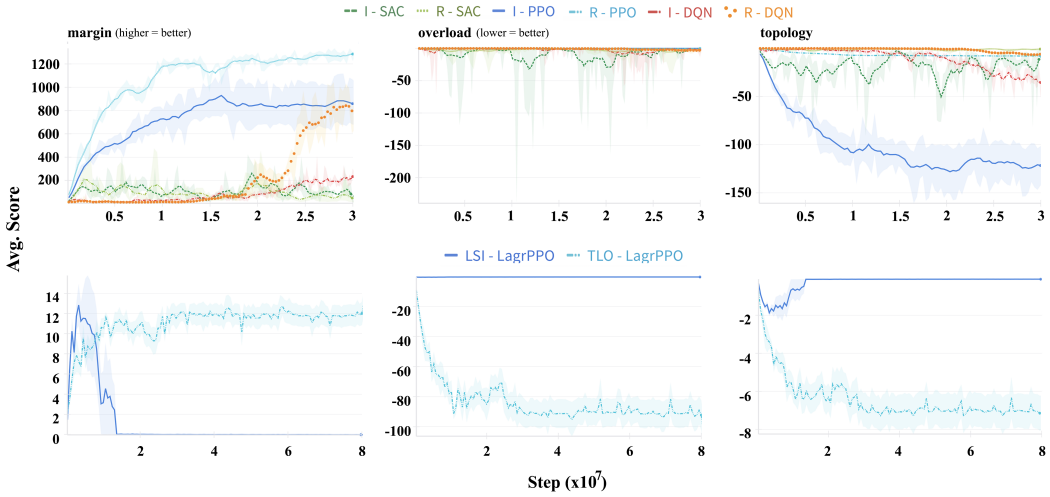

Figure G.1: Average score for key operational components across heuristic and constraints-based baselines (reported on separate rows). The first column represents line margins, where higher values indicate better contingency management. The second column tracks overload penalties, with lower values reflecting improved grid stability. The third column captures topology modifications, showing the extent to which agents reconfigure the grid.