L2RPN: Learning to Run a Power Network in a Sustainable World NeurIPS2020 challenge design

Antoine Marot Isabelle Guyon Benjamin Donnot
Gabriel Dulac-Arnold Patrick Panciatici Mariette Awad
Aidan O'Sullivan Adrian Kelly Zigfried Hampel-Arias

l2rpn@chalearn.org

June 24, 2020

Abstract

We present the design of the NeurIPS2020 Learning to run a Power Network challenge (L2RPN)[1]. power networks transport electricity across states, countries and even continents. They are the backbone of power distribution, playing a central economical and societal role by supplying reliable power to industry, services, and consumers. Their importance appears even more critical today as we transition towards a more sustainable world within a carbon-free economy, and concentrate energy distribution in the form of electricity. Problems that arise within the power network range from transient brownouts to complete electrical blackouts which can create significant economic and social perturbations, i.e. *de facto* freezing society. Grid operators are still responsible for ensuring that a reliable supply of electricity is provided everywhere, at all times. With the advent of renewable energy, electric mobility, and limitations placed on engaging in new grid infrastructure projects, the task of controlling existing grids is becoming increasingly difficult, forcing grid operators to do "more with less".

This challenge is the latest round in a series of power network control challenges whose objectives have been to progressively test the potential of reinforcement learning (RL) to address this important real-world problem: controlling electricity transport in power networks running closer to their operational limits while keeping people and equipment safe. While initial competitions tested the feasibility of developing realistic power network environments and the applicability of RL agents, this competition introduces a realistically-sized grid environment along with two fundamental real-life properties of power network systems to reconsider while shifting towards a sustainable world: robustness and adaptability.

[1] `https://l2rpn.chalearn.org/`

Keywords

Reinforcement Learning, Robustness, Adaptability, Industrial Control, Real-World RL, Safety, Smart Grids, Stochastic environment, Renewable Energy, Climate Change, Sustainable World

1 Introduction

1.1 Background

Electrification has played a prominent role in the development of modern societies in the past century. Some institutions such as the National Academy of Engineering acknowledge it as the number one engineering achievement of the 20th century [2]. Electricity has now become a commodity, and power networks transporting electricity across states, countries, and continents are essential components of modern societies. Such grids play a central economical and societal role by supplying reliable power to industry, services, and consumers. Electrical transmission is not a simple task, however, and requires 24/7 monitoring and control of the grid to avoid electricity blackouts, which can lead to significant losses and delay in public services and strategical industries. Grid operators are skilled engineers, in charge of ensuring that a reliable supply of electricity is provided everywhere and at all times. As surprising as it may be, their task is becoming increasingly difficult in the digital era because efforts made to automate operation are insufficient and they have to constantly examine massive amounts of data in real-time [13]. In the face of a resurgence of blackouts in modern but aging grids (California, New York, Australia, The UK in 2019), new developments are becoming urgent to keep operating **robust** grids [11]. **Electricity should be a reliable commodity, especially for those in less densely populated areas with more difficult access to public services**. Power systems are in some sense archaic, complex, "artificially intelligent"[3] systems currently in operation, and they are in great need of 21st century innovation to manage the increasingly complex task of satisfying electricity demand, all while using renewable energies and opening market exchanges. Renewable sources such as wind and solar are intermittent sources of energy due to their dependence on meteorological conditions. Also, while providing opportunities for exchange, electricity markets bring in their own variability and uncertainties in the system. As with most national power network operators, the French national grid operator, Réseau de Transport d'Électricité (RTE), is undergoing rapid and profound changes under a steep energy transition. This places new flexibility and reactivity requirements on the smart grids of the future. **Adaptability** is key for the power network to fully **reach its**

[2] Electrification among the 20^{th} century greatest achievements `http://www.greatachievements.org/`

[3] Although outside the scope of our this challenge design, it is important to point out that the current control algorithms for the grid are complex & distributed already, and leverage notions of optimization and prediction.

potential in mitigating climate change by allowing for total de-carbonization of our energy system, doing this under a sustainable approach with as little new infrastructure footprint as possible.

Rather than relying on additional hardware (e.g. more transportation lines), new software that could operate the grid with the latest artificial intelligence (AI) could be a game-changer towards optimizing the usage of existing assets [5]. In the community of power systems, optimal control methods [3] have fallen short because they are neither able to scale to multi-step time horizons, nor deal well with the large, non-linear combinatorial action space. Humans, perhaps surprisingly, are effectively still in charge of operating the grid in near real-time. With recent breakthroughs, reinforcement learning [14, 2](RL), is a promising avenue to develop artificial agents capable of operating acomplex power system in real-time.

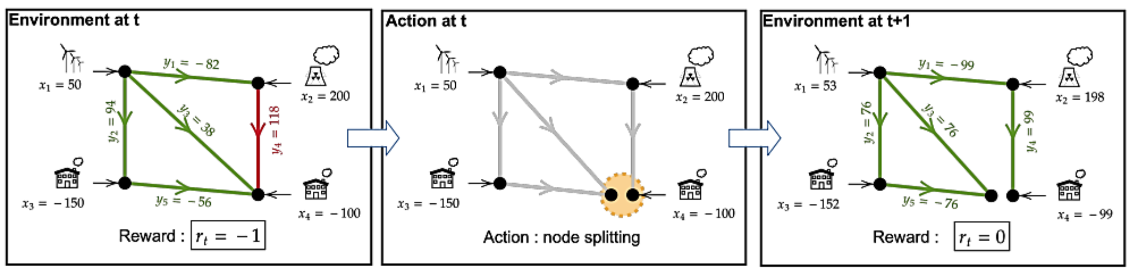


Figure 1: **L2RPN quick overview:** an agent first observe the power network state at time **t** with flows **y** in power lines given injections **x**, aka productions and consumptions. An overload (red line) occurs on the grid: the agent gets a negative reward from the environment, since the grid is at stake. The agent hence take a discrete node splitting remedial action at time t. 3 node splitting actions would have been possible at this 4 connectable element location: the number of possible node splitting actions increases exponentially with the number of connectable elements. While productions and consumptions change at **t+1**, the problem is solved and the agent gets a better reward.

1.2 Open Real-World Research Challenge

The Learning to Run a Power Network (L2RPN) challenge aims at enabling significant advances in solving an important, real-world problem of great societal importance. The safe and efficient transmission of electricity in the face of strong production and consumption shifts while limited in grid capacity expansion is one of the primary engineering hurdles for humanity's transition to zero-carbon energy. This has a positive societal impact both in highly developed nations and in those whose power networks are still being developed, enabling vital services ranging from household lighting to functional hospitals. **This new**

challenge is also of prime interest to the RL community to make advances in the field by addressing a real-world problem, generalizing RL methods to work on out-of-distribution events as well as non-stationary and complex environments. The challenge is centered around a simulator that represents the underlying complexities of real-world grid operations. This work is an important joint effort, bringing together collaborators in research (INRIA, UCL, Turing Institute, Google Research, Kassel University, Lab41 - US research lab) and a power system consortium (RTE, NREL - the American National Laboratory for Renewable Energies, EPRI - the Electric Power Research Institute, pandapower team - the open-source power network simulation library, TenneT - the national grid operator of the Netherlands and large parts of Germany). The collaborative group released a white paper as a companion for this design paper to explain in simple terms the different necessary concepts of a power network and power network operations, and to highlight upcoming challenges on which RL could help [10].

This challenge uses a game-like simulation environment emulating a real power network, in which participants build grid-operating agents with real-time control of the power network structure. These agents are responsible for safe and efficient grid operation, while respecting the physical constraint of balancing the supply and demand of energy and following temporal operational rules. The open-source simulation environment is based on the Grid2Operate platform, runs with pandapower power network simulator backend[15], and implements the OpenAI Gym API[1]. Grid2Operate models realistic concepts found in real-world operations used to test advanced control algorithms. We, the organizers, provide encouraging baseline results suggesting RL solutions may improve upon trivial benchmarks such as a static policy or expert systems.

Although many power network simulators exist, few framework are built to be used as a game-like simulator for training and evaluating RL agents, as illustrated in Figure 1. To our knowledge, this is the only open-source game-like simulator built for grid operation simulation and in direct collaboration with a large-scale grid operator. This ensures grounding of our test cases, modeling of real-time operations and the simulation of grid design choices. **In the long term, this testbed platform will enable the research community to address new questions** by creating new synthetic environments, and running new experiments and benchmarks to identify relevant new policies. Another project, funded by the United States Department of Energy, follows a similar avenue though with a focus on industrial implementation rather than for research purposes[4].

This competition is held by having participants train their agents on a series of predefined scenarios. They may then evaluate their agent on hidden validation scenarios having the same grid layout but under different weather, production, consumption conditions and events. The Codalab challenge platform assesses submissions and ranks them accordingly on a public leader board. The ultimate goal of the competition is to encourage

[4]HADREC Arpa-e project `https://arpa-e.energy.gov/?q=slick-sheet-project/high-performance-adaptive-deep-reinforcement-learning-based-real-time-emergency`

research in the next-generation grid control, with the hope of demonstrating the abilities of current methods to successfully solve harder tasks, as well as encouraging various national grid operators to adopt next-generation machine-learning-based control strategies. Hopefully, this contributes to creating a community around machine learning methods for power system control, so that these two historically rich but disjoint communities join forces. The challenge aims at attracting a community of researchers around the globe with diverse geographic demography so that nations of the global South can participate and be introduced to these control strategies and hopefully leap-frog when it comes to modernizing their infrastructure and resources. The challenge can be an eye-opener for communities to avoid investing in traditional power network expansion and instead increase reliance on advanced machine learning and AI-based control that would help the world reach a sustainable zero-carbon emission more efficiently.

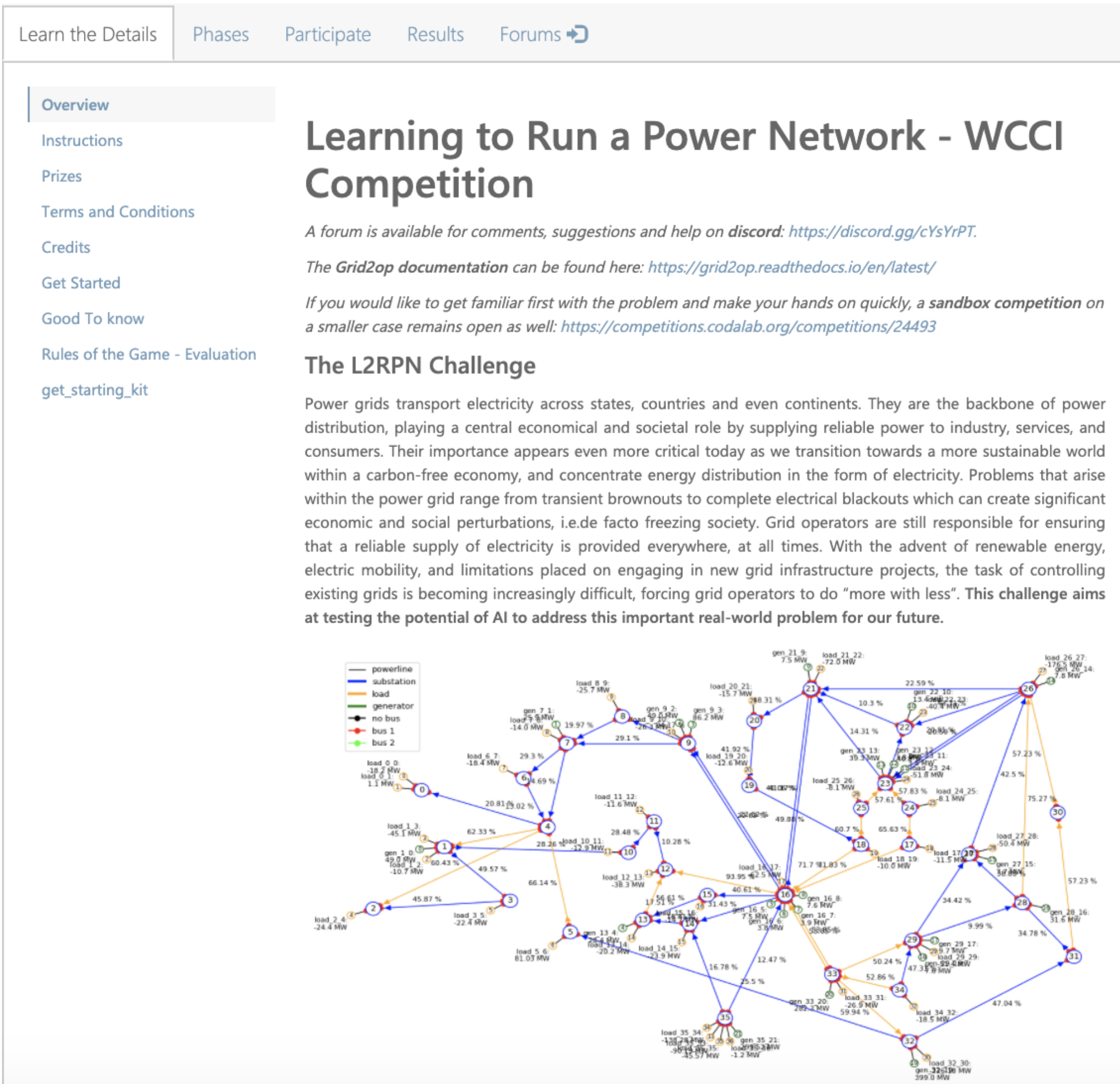
2 Novel challenge design

L2RPN (Learning to Run a Power Network) is a challenge series we started in 2019, which is, to our knowledge, the only one of its kind. Before the NeurIPS2020 competition, we organized two smaller scope challenges to iron out our competition protocol and get ready to scale up tasks for NeurIPS participants to a larger realistic power network (corresponding to the state of California, USA). Figure 2 shows an example of previous L2RPN competition page[5].

The common objective of the entire challenge series is to advance towards empowering existing 20^{th} century power infrastructure with artificial intelligence to become 21^{st} century long awaited "smart grids". AI can greatly enhance the capabilities of current power network controllers. On one hand, such novel controllers should **reduce operational cost** for daily operations and **improve grid robustness**. On the other hand, in mid-term to long-term computationally intensive studies, the integration of adaptive and intelligent controllers should **reduce the grid footprint when integrating new energy resources**. After two initial competitions that validated both the feasibility of our synthetic power network environments and the viability of reinforcement learning, this new competition will be a key milestone in AI research having real-world impact on existing power networks.

IJCNN 2019 & 2020 - L2RPN Feasibility Challenges: A significantly reduced version of the NeurIPS challenge was first run at IJCNN 2019 [12] on a standard academic power network case (IEEE 14 grid). This demonstrated that RL approaches can accommodate the high combinatorial space and adequately change grid connectivity patterns, i.e. the grid's topology, to avoid certain blackout and efficiently manage the power flows on scenarios. There exists no other automated methods beyond the simplest level of "branch switching" [7], even on small grids. A second follow-up competition has also run for IJCNN

[5]Codalab WCCi competition page: `https://competitions.codalab.org/competitions/24902`


Learn the Details | Phases | Participate | Results | Forums

- Overview
- Instructions
- Prizes
- Terms and Conditions
- Credits
- Get Started
- Good To know
- Rules of the Game - Evaluation
- get_starting_kit

Learning to Run a Power Network - WCCI Competition

*A forum is available for comments, suggestions and help on **discord**: https://discord.gg/cYsYrPT.*

*The **Grid2op documentation** can be found here: https://grid2op.readthedocs.io/en/latest/*

*If you would like to get familiar first with the problem and make your hands on quickly, a **sandbox competition** on a smaller case remains open as well: https://competitions.codalab.org/competitions/24493*

The L2RPN Challenge

Power grids transport electricity across states, countries and even continents. They are the backbone of power distribution, playing a central economical and societal role by supplying reliable power to industry, services, and consumers. Their importance appears even more critical today as we transition towards a more sustainable world within a carbon-free economy, and concentrate energy distribution in the form of electricity. Problems that arise within the power grid range from transient brownouts to complete electrical blackouts which can create significant economic and social perturbations, i.e.de facto freezing society. Grid operators are still responsible for ensuring that a reliable supply of electricity is provided everywhere, at all times. With the advent of renewable energy, electric mobility, and limitations placed on engaging in new grid infrastructure projects, the task of controlling existing grids is becoming increasingly difficult, forcing grid operators to do "more with less". **This challenge aims at testing the potential of AI to address this important real-world problem for our future.**

Figure 2: Illustration of Codalab challenge platform page for L2RPN WCCI Competition

2020 challenge[6] which tested the scalability of such methods on a larger grid (1 region out of 3 on the IEEE 118 grid) while dealing with new discrete events such as planned maintenance.

Neurips 2020 L2RPN in a Sustainable World Challenge: The NeurIPS2020 challenge evaluates artificial agents on environments addressing current real-world challenges related to robustness and adaptability. Successful agents may demonstrate to the power system community the practical applicability of these methods on real-world grid operations and investments. This is the main point of novelty of this challenge compared to our previous feasibility-related challenges. The NeurIPS2020 challenge significantly extends our previous smaller challenges in the following ways:

1. **Real-sized grid**: The IEEE 118 grid is approximately the size of some European countries' transmission grids and has also been used in research to represent the Californian grid for instance [9]. It is the most studied grid in power network literature.
2. **Realistic stochastic time series**: Collaboration with NREL enabled us to create time series with realistic dynamics and production intermittence for renewable energies, which will be the largest source of variability in coming years.
3. **High-dimensional continuous and discrete state and action spaces**: The NeurIPS2020 challenge allows agents to perfom continuous actions such as production re-dispatching, in addition to discrete combinatorial topological actions. Power production is now the result of both exogenous (programmed) and endogeneous (corrective action) decisions, which in combination with an increase in grid size in turn increases the space dimensionalities. This is further explained in subsection 1.3.1 and summarized in Figure 3.
4. **Out-of distribution adversarial attack**: The participants are tested in the robustness track on adversarial hazardous events properly selected by challenge organizers. This should favor RL approaches with notions of safety.
5. **Non-stationary environments**: The *adaptability track* has distribution shifts in production with increasing renewable energy and decreasing thermal production. This might favor meta learning or model-based approaches.

In addition to providing more realistic scenarios that aim to convince grid operators to seriously consider machine learning approaches, our challenge also provides a platform that instantiates general challenges to real-world RL as well. The uptake of RL in real-world systems is hindered by a series of challenges [6], which we manifest through multiple aspects of our environment: safety, high-dimensional state/action spaces, non-stationarity and unspecified reward functions. As these are all present in our challenge environment,

[6] 2020 IJCNN Competition: `https://wcci2020.org/competitions/`

contributions are likely to also produce general algorithmic contributions for real-world RL.

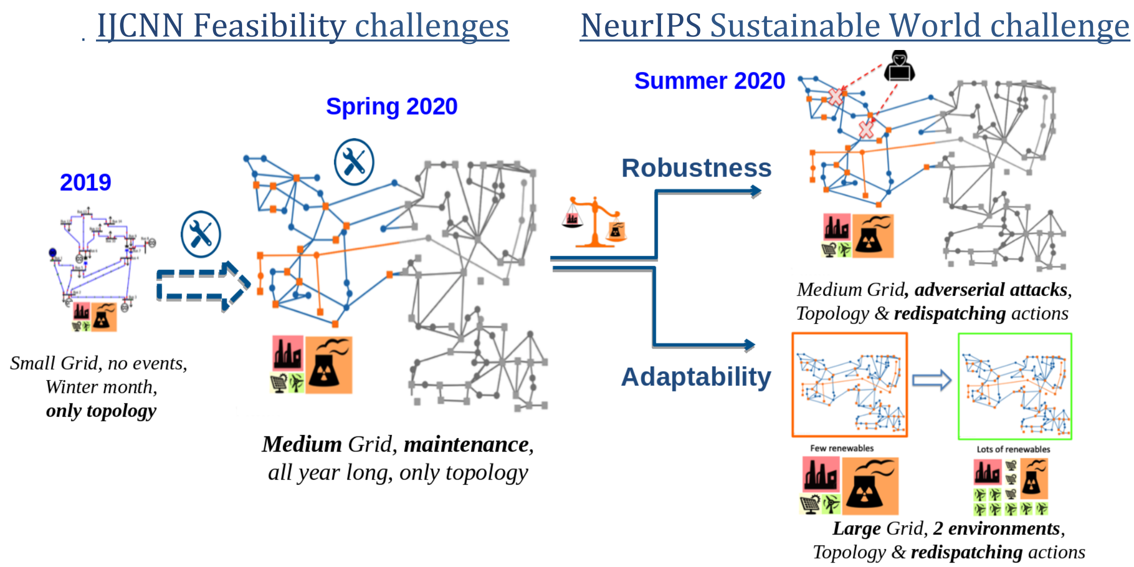


Figure 3: **Successive challenges in L2RPN series:** maintenance events and redispatching actions are successively introduced, while the grid size increases and new harder game objectives are defined such as robustness and adptability.

3 Data & Environment

As this is a RL competition, instead of a single dataset, we provide a whole environment in which participants can train and evaluate their agents. The environment is very comprehensive, however, its underlying complexity is hidden to the participants. **The effective barrier of entry to novice participants is very low**, as can be appreciated by the sample code provided in Figure 7. This is important to engage new machine learning participants. We also provide a GUI interface to help participants understand the problem and get an intuition of the tasks and the effectiveness of various agents.

Our recent paper [12] describes what we considered as an environment and how we designed one for our first challenge. We summarize here the key elements and then highlight our new developments. Specifically, we developed a new framework, Grid2Operate, allowing for more modularity, robustness and complexity compared to our previous pypownet platform. It has been regularly used and tested by our collaborators throughout its development, with the goal of becoming the reference platform for real-time power network operation research.

	Challenge Serie			
Features	IJCNN 2019	IJCNN 2020	Neurips Track 1	Neurips Track 2
Grid size (number of nodes)	14	45	45	118
Discrete (Topological) actions	Yes	Yes	Yes	Yes
Continuous (Redispathing) Actions	No	Limited	Yes	Yes
Discrete Events	No	Yes	Yes	Yes
Adversarial Events	No	No	Yes	No
Non-Stationary DIstributions	No	No	No	Yes
Long Scenarios Testing	No	No	Yes	Yes
Action Space Size - Discrete - Continuous	205 - 205 - 0	4005 - 4000 - 5	4020 - 4000 - 20	12060 - 12000 - 60
Observation Space Size - Discrete - Continuous	450 - 360 - 90	1500 - 500 - 900	1500 - 500 - 900	4200 - 1 500 - 2700

Figure 4: **Differences in L2RPN series**: new features along challenges lead to more realistic and complex control problems (red highlights when the feature does not exist or is very simplistic, in green when its dimension is complex, in orange when in between).

3.1 Environment Description

A power network environment is the combination of the following components, most already described in [12] and illustrated on Figure 5:

- A power network (with substations and powerlines of different characteristics) and a grid topology $\boldsymbol{\tau}(t)$, represented as a **graph**. The IEEE 118 scenario is chosen for this challenge as previously discussed.
- Real-time power injections (productions and consumptions) $\mathbf{x(t)}$ and stochastic forecasts of the upcoming time-step $\mathbf{\hat{x}(t{+}1|t)}$.
- Events Ev(t) such as maintenance (deterministic) and line disconnection contingencies (stochastic). Events are a novelty this year.
- Line Capacities $y_{\mathbf{lim}}$ and additional operational rules such as reaction time, activation time or recovery time applicable to actions taken by the agent.
- Terminal conditions such as not being able to supply the necessary power after a cascading failure on the grid, or not finishing a scenario in the allocated time.

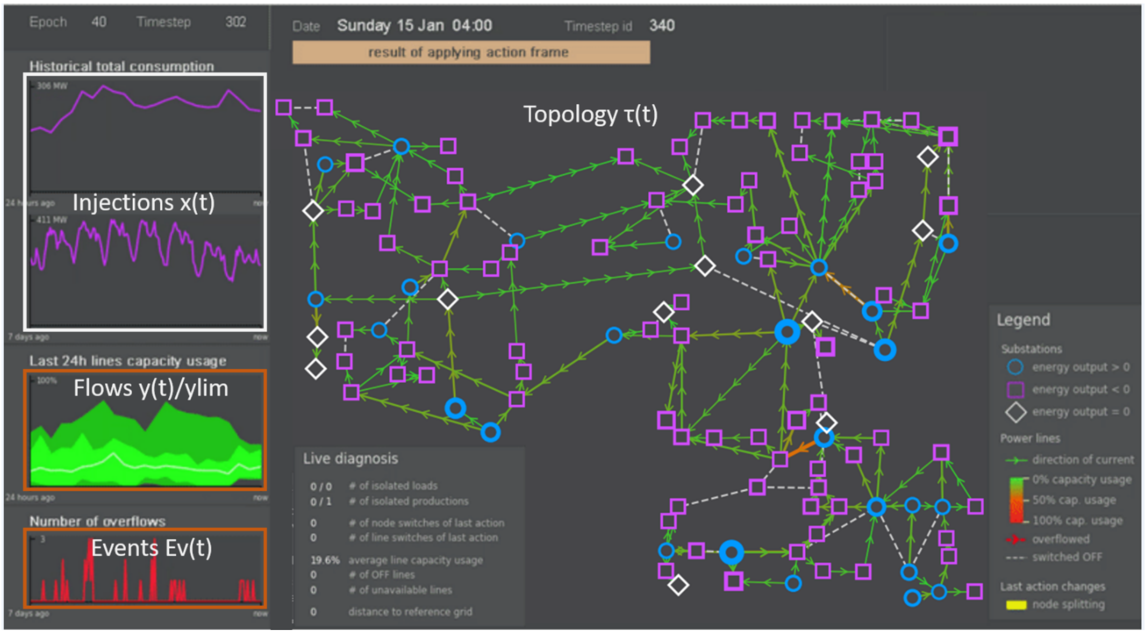


Figure 5: **Illustration of L2RPN Environment**: injections x(t), events Ev(t) and a grid topology τ(t) induces flows y(t) in every power lines that the agent needs to manage.

The line power flows $y(t)$ are computed by a power-flow simulator (pandapower). The Grid2Op framework then allows, as latter described, some interactions for an agent with this physical simulator through actions $a(t)$. Participants have the option to simulate the effect of their action before choosing the next time step forecast, though by using additional computation time budget. **More explanations can also be found in our white paper**[10].

Agent Actions: There exists two families of actions: one permits topology changes (discrete and combinatorial actions) while the other allows production changes (continuous actions) for safety reasons, while previously set by the market.

The topology can be changed by switching on and off power lines (about 100 possible unitary actions on IEEE 14 with 20 lines - this number scales linearly with the power network size) or by node splitting actions as in Figure 1 (about an additional 100 unitary actions on IEEE 14 with 14 substations - this number grows exponentially with the size of grid).

The productions can also be changed with respect to physical properties such as maximal and minimal possible production per power plant as well as on the maximum possible increase or decrease for each power plant between consecutive time-steps. An agent can choose to increase or decrease the power of a production unit leading to a continuous action space having a dimension equal to the number of generators on the grid.

3.2 Environment Formalisation

We can formalise the environment as a Markov Decision Process (MDP). A MDP can be defined as a tuple $(\mathcal{S}, \mathcal{A}, p, r)$, where an agent observes a state $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$ at timestep t. From state s_t and taking an action a_t, an agent arrives in a new state s_{t+1} with probability $p(s_{t+1}|s_t, a_t)$, and receive a reward $r(s_t, a_t, s_{t+1})$. Our environments are episodic, which is to say that they last a finite number of timesteps, $1 \leq t \leq T$.

MDPs generally respect the Markov property, which states that all the information necessary to generate s_{t+1} can be found in s_t and a. In our case, however, the state is *partially observed*, which means that underlying dynamics such as demand, weather, and other events at s_{t+1} can not be necessarily predicted given the information present in t_1. We a certain abuse of notation, we consider the observation and state at time to be equivalent in the eyes of the agent, denoting it s_t. Default representations of s_t, a_t and of the reward function are provided, but participants are free to adapt them to their needs.

Participants can access the information they find necessary by querying the simulator at each timestep, and constructing the state representation as they see fit using the Grid2Op "Observation" object that can be easily cast into a numeric vector. The same holds for actions they send that can be easily transformed to / from more numeric vectors. The functionning of these conversion is extensively explained in some tutorials already available[7]. Additionally, although there is a score function defined in Sec. 5, participants are free to define their own reward function, as balancing the various objectives is not necessarily best done by simply joining them into a global reward function. Interesting properties of our environments (real-sized, stochastic, out-of-distibution adverserial attacks, high-dimensional action spaces non-stationnary) have been described previously in Sec. 2.

3.3 Datasets for the competition

Multiple hundreds of yearly scenarios at 5 minutes resolution are available as a training set, generated with ChroniX2Grid package [8], and can be easily downloaded with grid2op (if the environment is new it is downloaded automatically, otherwise the previously downloaded version is used) .

```python
import grid2op
env = grid2op.make("l2rpn_case14_sandbox")
# the variable "env" implements the complete open ai gym interface
```

Figure 6: How to import a new environment using the grid2op framework

A few dozens of weekly scenarios carefully selected under specific criteria are kept secret,

[7]In the getting started section of the Grid2Op framework `https://github.com/rte-france/Grid2Op/tree/master/getting_started` and especially the notebook "4_TrainingAnAgent"

[8]ChroniX2Grid github: `https://github.com/mjothy/ChroniX2Grid/tree/master/chronix2grid`

using half for evaluation on Codalab during the competition and the other half for testing at the end of the competition, as further described in “Metrics” section.

3.4 Grid2Operate open-source framework

Even though the addressed problem of controlling power systems is novel for the AI community, a lot of care have been taken to ensure that participants can interact with it in a familiar way. The Grid2Operate framework (detailed in Figures 9 and 10) allows easy manipulation of the power network using the reinforcement learning framework “OpenAI Gym”. This framework is widely used by the RL community and has been successfully used last year for the MineRL competition [8].

The Grid2Operate framework comes with multiple environments already available for testing and training at the time of writing. These environments vary in size (from a 5 substations system as a tutorial, to systems with 118 substations on which they will be tested) and difficulty. For example, Figure 7 shows how to simply interact with the Grid2Op environment[9] once an agent is defined. Another capability of grid2op is to render the power network represent an observation for the “case5_example” environment as in Figure 8 which helps demonstrate how to use the platform and quickly analyze the environment.

In addition to all this material, some baselines were also made available to the participants (see subsection 6.3 for more details).

The baseline code is open-sourced [10], easily importable and usable by participants. Some explanations on the nature of these baselines were also be made available.

We now describe the high-level Grid2Operate framework architecture.

Environment class Fully compatible with the RL framework Open AI gym, this environment implements the widely used “*env.action_space*”, “*env.observation_space*”, “*env.step()*” or “*env.render()*”. A special care has been taken to facilitate the use of this package without requiring understanding the full details of the physics behind power systems. For the Robustness Track (see subsection 4 for more information) the environment also bears an ”Adversarial actor“ which takes actions given a specific budget in response to the action of the Agent.

Action class This class represents the way the “Agent” interacts with the Environment. It can be represented either as an object (default representation), or a vector[11] when this makes it easier to use. For the target environment of the second track of this challenge we

[9]Github does not allow natively to run a notebook, to try it yourself, you may want to clone the github repository or use the MyBinder tool at `https://mybinder.org/v2/gh/rte-france/Grid2Op/master`

[10]L2RPN Baseline github: `https://github.com/rte-france/l2rpn-baselines`

[11]Utilities are provided to make the conversion from one type to another as easy as possible. This process is also illustrated in a dedicated notebook for improving clarity.

```python
import grid2op
from grid2op.Agent import RandomAgent
env = grid2op.make("l2rpn_case14_sandbox")
my_agent = RandomAgent(env.action_space)
done = False
reward = env.reward_range[0]
obs = env.reset()
while not done:
    # optional
    # env.render()
    action = my_agent.act(obs, reward, done)
    obs, reward, done, info = en.step(act)
```

Figure 7: **Example of code running a baseline agent**. In this example code, we show how to create the example environment "case5_example", how to initialize a baseline agent, in this case the "DoNothing" Agent (see subsection 6.3 for more details) and how to assess its performance on this environment. A lot of programming effort has been put to offer an interface fully compatible with OpenAI gym. More baselines, example codes and material to get started with grid2op can be found on the official github repository at `https://github.com/rte-france/Grid2Op`.

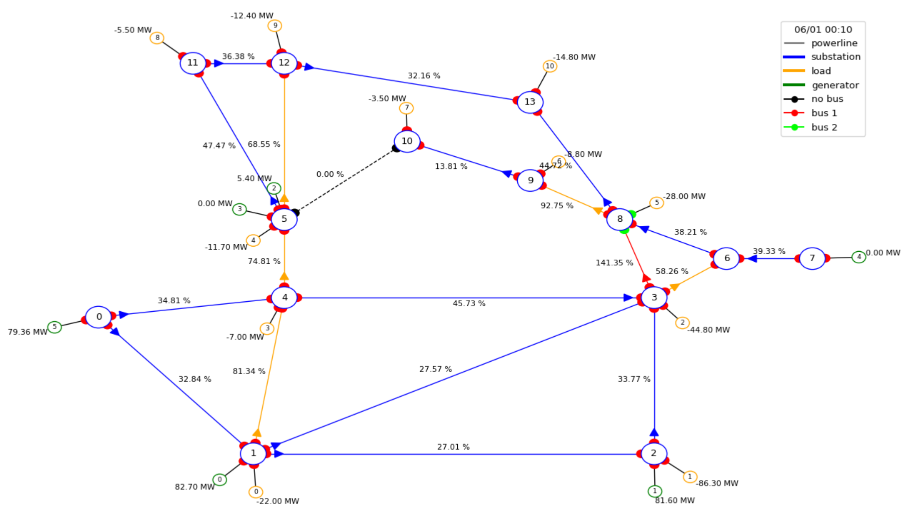


Figure 8: **Visual representation of a grid2op Observation**. The Grid2Operate framework allows people to easily inspect an observation. One of the method is to represent it as a graph. Participant can zoom in / out, focus on a specific part of the grid. The interactive version of this plot can be found in the jupyter notebook `https://github.com/rte-france/Grid2Op/blob/master/getting_started/Example_5bus.ipynb`.

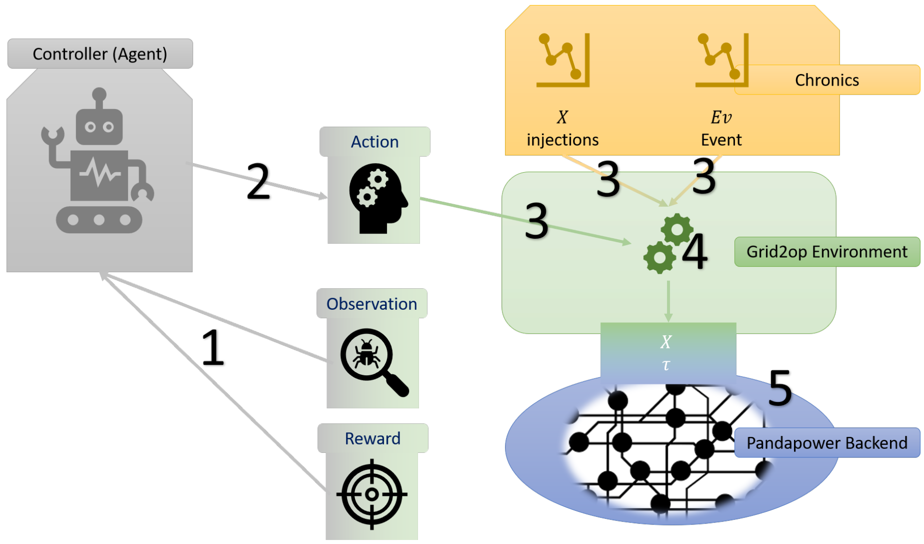


Figure 9: At time t, the agent receive a reward [scalar] and an observation [object convertible to vector] from the environment (1). The Agent then produces an action (2). This action is sent int turn to the environment (3). The environment is further updated with new chronic values (4). The pandapower backend start its computation (5).

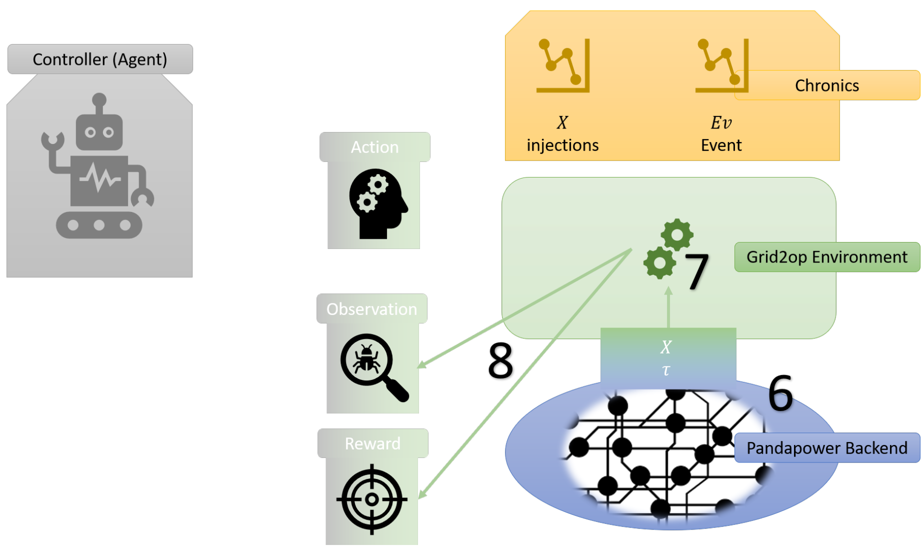


Figure 10: The powerflow solver (pandapower) is run (6). It allows the environment to retrieve grid state information through an API (7). Is also checked whether an action is valid or not, hence considered or ignored. New Observations and reward signal from the environment are sent to the agent at time t+1 (8). In case of a game over, it terminates.

use a power network with 118 substations. A dense representation of this vector counts 1 500 components as summarized in Figure 4. However, if we use a more standard representation of the action space, in the form of a one-hot sparse vector, this takes 22 000 independent unitary discrete independent actions. To reduce the size of the action space, the organizers might select among these actions the most relevant ones to help the participants[12]. In addition to this discrete action space, 60 continuous actions on the thermal generators, which production can be adapted in real time, are available.

Observation class The observation space is composed of all the data that can be extracted from the power network. It is made of continuous components (loads, productions, flows, voltages, etc.) and some discrete components (line status, can a powerline be reconnected?, topology vector, etc.). For this competition it has a size of 4 200. Unlike more classical reinforcement learning frameworks, a function called "simulate" is also available, to mimic the operational process (in which operators can use simulators to compute the possible outcome of some actions) and make the problem more tractable. This function allows the participants to simulate the effect of their action on a forecasted snapshot of the powergrid state. This can serve two major purposes: (1) to validate that an action is effective and (2) to explore new actions when training.

4 Tasks and application scenarios

Inspired by some of the high-level difficulties present in controlling real-world power networks, we have decided to put forward *robustness* and *adaptability* as the high-level tasks we want to challenge participants with. For both of these tasks, the constraints related to safety, production contracts and occasional planned & un-planned interruptions are also be present. As these track environments are built upon a previous IJCNN 2020 challenge, which will remain open as a benchmark, participants will be able to gradually develop and test their agents towards succeeding at these more advanced tracks.

We now describe more precisely the two tracks that were illustrated on Figure 3.

Robustness Track: For the last decades, developed power networks have tried to remain robust to unexpected events following a simple guiding "N-1" redundancy principle: the system should keep operating in nominal conditions despite the loss of any single asset such as lines or power plants at any time. Operators hence anticipated through forecast and simulations the effect of having any line disconnected in the hours to come. This very fundamental property has been the root of power system success: distribute electricity reliably, throughout the grid, at all times. However, since we are pushing these aging systems towards their limits, this redundancy constraint is no longer easy to satisfy. In

[12]This pre-selection aims at helping the participants. If some participants wants to use the full size action space, they are be allowed to.

addition, climate change can potentially lead to more frequent and extreme weather events which have to also be accounted for[13]. Operators now need to consider more generic worst-case scenarios [4], pushing redundancies to N-2 (there are 40 000 N-2 contingencies to consider compared to 200 N-1 contingencies on the 118 IEEE grid). In addition to this, the temporal resolution is moving to 15 minutes sampling and refresh rates instead of 3 hours used previously. Because of computation time budget constraint in real-time operations, not every contingency can be studied on this forecasting horizon. **There is therefore a need for new methods to be robust to worst-case events in a sample-efficient and real-time manner**.

To encourage work in this higher-risk setting, we define an adversarial environment where adversarial actors can dynamically attack the grid. Extending the environment of a preceding IJCNN 2020 competition, an agent designed by the organizing team will choose adversarial attacks under some budget, mainly in terms of line disconnection frequency and duration constraints, and apply it to this former environment. The attacks of the adversarial actor will be designed to consistently disrupt previously winning agents from the IJCNN challenge. We might limit the possible attacks to a specific set of selected line disconnections to keep it manageable for the participants. New submissions in this adversarial environment will have to be robust to those worst-case events on new test scenarios. Simple adversarial actors will be made available to the participants for training, while not revealing the full adversarial actor used during evaluation. As a real-life use case, this approach is inspired by the way in which power network operators such as RTE test identified potential cases of attacks (cyber, climate) on vendor controller solutions to assess their robustness. AI agents will therefore also need to respond well to these scenarios if they want to be considered for use in real systems.

Adaptability Track: With the current strong trend of de-carbonization of power production, the energy mix is moving to a significantly greater share of renewable production, which comes with production irregularities that are harder to predict. At the same time, national infrastructure projects are less frequent, and the national grid can no longer grow as quickly as in the past. As power line construction is limited, the current grid must be flexible enough to adapt its operations to this new energy mix that it was not originally designed to handle.

Given these constraints, grid operators would like to understand **how adaptable AI-based grid controllers can be to changing supply and demand patterns, given a static grid capacity.** In the end, they would like to assess how much more flexibility AI controllers can offer to avoid new costly investments in the future in line with [9]. In the spirit of this question, our second track asks participants to create agents that can deal both with today and tomorrow's energy mix while relying on the same underlying grid. This non-stationary scenario allows us to test agents' abilities in adapting to evolving

[13]The storm of 1999 in France had a formidable impact on grid infrastructure: `https://www.rte-france.com/l-heritage-de-la-tempete/`

changes in grid use. We provide training scenarios with both types of supply & demand patterns for training. Participants will then be tested on new unseen scenarios that can come from either energy mix without identifying the type of scenario. We imagine more complex methods such as model-based or transfer learning methods could help succeed in this track. As a real-life use case, this approach is inspired by the way in which RTE studies the capability of integrating new renewable resources in the future to make relevant recommendations to the government and civil society to enter a sustainable world.

One of the hopes of using learned agents for grid control is that these agents will discover novel strategies that could be used effectively in live scenarios. In the spirit of AlphaGo demonstrating totally novel ways in which the game of Go can be played, we believe it will be very interesting to discover strategies found by artificial agents in the face of difficult control scenarios. For instance, **it remains largely unknown today how much flexibility an existing grid topology can eventually offer when controlled widely at a high frequency** because of its non-linear combinatorial complexity. Identifying new unknown but applicable strategies would already be a great step towards smarter grids.

For each track, baselines are available to the participants at the beginning of the competition. New stronger baselines could be released in the second half of the challenge to stimulate competition.

5 Metrics

While participants can design their own reward when training their agent, the principle of a challenge requires there to be an objective score at evaluation time. As electric grids are more and more modeled as live market exchanges, almost all of the grid's operational characteristics can be converted into a monetary cost. Therefore, the score reflects the realistic operational costs of a power network, which grounds the algorithmic performance of proposed agents in a very real-world quantity: money.

To begin with, we recall that transporting electricity always generates some energy losses[14] $\boldsymbol{\mathcal{E}_{loss}(t)}$ due to the Joule effect in resistive power lines at any time t:

$$\mathcal{E}_{loss}(t) = \sum_{l=1}^{n_l} r_l * y_l(t)^2 \tag{1}$$

At any time t, the operator of the grid is responsible for compensating those energy losses by purchasing on the energy market the corresponding amount of production at the marginal price $\boldsymbol{p(t)}$. We can therefore define the following energy loss cost $\boldsymbol{c_{loss}(t)}$:

$$c_{loss}(t) = \mathcal{E}_{loss}(t) * p(t) \tag{2}$$

[14]This energy loss corresponds to 2.2% of the total energy consumption on high voltage power networks: `https://bilan-electrique-2018.rte-france.com/loss-rate/?lang=en`

Topological action can increase or decrease $\mathcal{E}_{Loss}(t)$. This already leads to a continuous optimization problem to solve. Energy losses cost RTE 500 million €/year, so **a gain of 20% would already save 100 million €/year**.

Then we should consider that operator decisions when taking an action can induce costs, especially when requiring market actors to perform specific actions, as they should be paid in return. Topological actions are mostly free, as the grid belongs to the power network operator, and no energy cost is involved. However, redispatching actions involve producers which should get paid. As the grid operators ask to redispatch energy $\boldsymbol{\mathcal{E}_{redispatch}(t)}$, some power plants increase their production by $\mathcal{E}_{redispatch}(t)$ while others compensate by decreasing their production by the same amount to keep the power network balanced. Hence, the grid operator pays both producers for this redispatched energy at a cost $\boldsymbol{c_{redispatching(t)}}$ higher than the marginal price $p(t)$ by some factor α:

$$c_{redispatching}(t) = 2 * \mathcal{E}_{redispatch} * \alpha p(t), \ \alpha \geqslant 1 \tag{3}$$

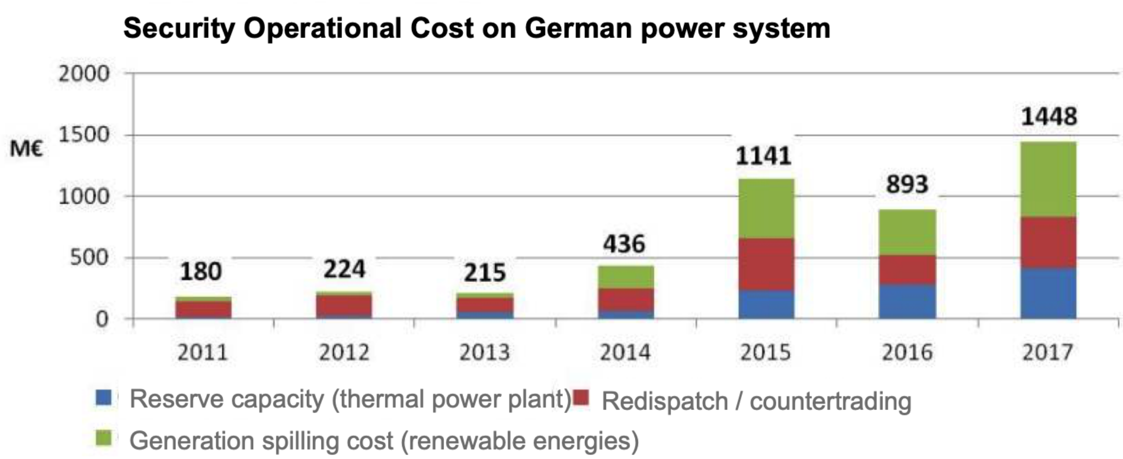


Figure 11: **Undesirable 1 billion/year operational cost sharp increase in Germany** in recent years after quick installations of renewables without developing new flexibilities

If no flexibility is identified or integrated on the grid, operational costs related to redispatching can dramatically increase due to renewable energy sources as was the case recently in Germany with **an avoidable 1 billion €/year increase**[15] illustrated on Figure 11.

[15] German power system operational cost `https://allemagne-energies.com/2018/06/19/allemagne-14-milliards-deuros-pour-stabiliser-le-reseau-electrique-en-2017/`

We can hence define our overall operational cost $\boldsymbol{c}_{\text{operations}}\boldsymbol{(t)}$:

$$c_{\text{operations}}(t) = c_{\text{loss}}(t) + c_{\text{redispatching}}(t) \tag{4}$$

Formally, we can define an "episode" $\boldsymbol{e}$ successfully managed by an agent up until time $\boldsymbol{t}_{\text{end}}$ (over a scenario of maximum length T_e) by:

$$\boldsymbol{e} = (o_1, a_1, o_2, a_2, \ldots, a_{t_{\text{end}}-1}, o_{t_{\text{end}}}) \tag{5}$$

where o_t represents the observation at time t and a_t the actions the agent took at time t. In particular, o_1 is the first observation and $o_{t_{\text{end}}}$ is the last one: either there is a game over at time t_{end} or the agent reached the end of the scenario. An agent can either manage to operate the grid for the entire scenario ($t_{\text{end}} = T_e$) or fail after some time t_{end} because of a blackout. In case of a blackout, the cost $\boldsymbol{c}_{\text{blackout}}\boldsymbol{(t)}$ at a given time t would be proportional to the amount of consumption not supplied $\text{Load}(t)$, at a price higher than the marginal price $p(t)$ by some factor β:

$$c_{\text{blackout}}(t) = \text{Load}(t) * \beta * p(t), \ \beta \geqslant 1 \tag{6}$$

Notice that $\text{Load}(t) >> \mathcal{E}_{\text{redispatch}}(t), \mathcal{E}_{\text{loss}}(t)$ which means that the cost of a blackout is a lot higher than the cost of operating the grid as expected. It is even higher if we further consider the secondary effects on the economy[16]. Furthermore, a blackout does not last forever and power networks restart at some point. But for the sake of simplicity while preserving most of the realism, all these additional complexities are not considered here.

Now we can define our cost $\boldsymbol{c}$ for an episode:

$$c(\boldsymbol{e}) = \sum_{t=1}^{t_{\text{end}}} c_{\text{operations}}(t) + \sum_{t=t_{\text{end}}}^{T_e} c_{\text{blackout}}(t) \tag{7}$$

We still encourage the participants to operate the grid as long as possible, but penalize them for the remaining time after the game is over, as this is a critical system and safety is paramount.

Finally, participants will be tested on $\boldsymbol{N}$ hidden scenarios of different lengths, varying from one day to one week, and on various difficult situations according to our baselines. This will test agent behavior in various representative conditions. Under those episodes, our final score to minimize will be:

$$Score = \sum_{i=1}^{N} c(e_i) \tag{8}$$

[16]More information can be found on this blackout cost simulator: `https://www.blackout-simulator.com/`

This score applies to both tracks, and highlights both the necessary cost of robustness in the first track, and the cost of adaptability in the second track. Participants who finish all the episodes, while optimizing for energy losses and minimizing the cost of their actions will in the end get the best rankings. **Failing at even one episode will significantly hinder the chances of one participant to win the competition**. Keeping hidden the test episodes prevents participants from overfitting on the episodes they have used during training.

Scaling of the score To improve the properties, readability and representativity of our score, based on our experience from the first competitions, we decided to rescale it. First, the raw score defined previously has to be minimized. But it feels more natural to rank participants according to a higher score. Second, the raw score for a given scenario lives in the range of millions to billions. This is not easily readable for any human, and it makes it hard to represent what would be the best reachable score. Finally, the raw score is mainly driven by the operational cost of blackouts and very little by optimizing energy losses (at least 2 order of magnitude difference in cost). So we wanted to more strongly encourage participants to optimize the continuous operational cost induced by energy losses , while still being robust to blackouts.

To address all these issues we decided to scaled the score in the following way, which is also illustrated in the figure 12:

1. to scale the score on a fixed range, from -100 (corresponding to a complete blackout during all the scenario) to +100 (corresponding to a large decrease in the system losses while managing the scenario completely - this is a "cheating agent" as all the safety considerations of the grid were disable, this represents an upper bound on what is possible).

2. to assign the best agents a score close to 100 and to the least performing agent a score closer to -100 (higher score is better)

3. finally we decided to scale the scores in a piecewise linear function such that the simplest baseline (do nothing) is assigned a score of 0, while the score of an agent that finishes a scenario by solely ensuring the safety (without considering the reduction of the system losses) is of 80. This means that 20 more points could be won by optimizing the system energy losses to reach a 100.

6 Get started: references and material available

6.1 References: White paper & website

The organizers wrote a white paper [10] for the purpose of this competition to give everyone the necessary background concepts to understand power network physics and operations

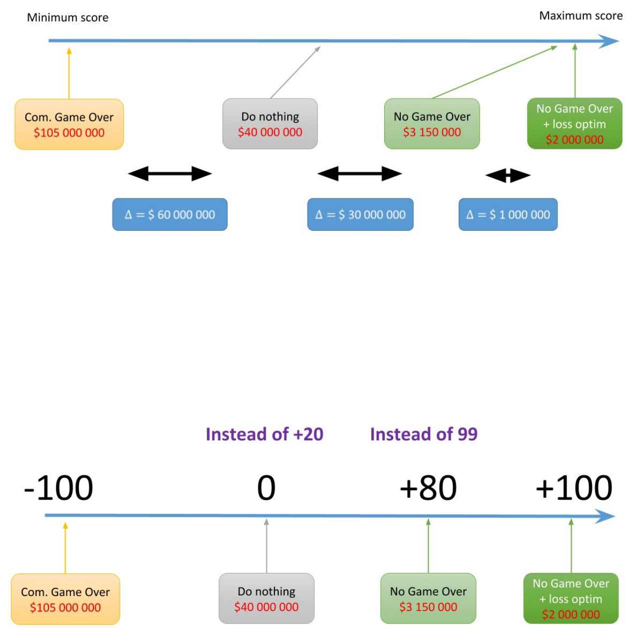


Figure 12: Scaling of the score explained: on top is represented, from left to right the operational cost the operational cost in four different cases (for four different "agents") and there associated value. On the bottom is represented the same four "agents" but emphasizing how our scaling affect their performances.

on the one side, as well as reinforcement learning framework on the other side. It further highlights the potential of applying Reinforcement Learning to power networks.

A challenge website[17]is also available, displaying more interactive materials to get familiar with this background knowledge. It present videos of the approaches used by the winner of the first 2019 L2RPN challenge. It finally list research works that could be of interest to participants for the competition.

6.2 Grid2Op platform & documentation

Grid2op users can get familiar with the framework with getting started notebooks (that can be run interactively thanks to MyBinder `https://mybinder.org/v2/gh/rte-france/`

[17]`https://l2rpn.chalearn.org`

Grid2Op/master) or downloaded from the official grid2op github https://github.com/rte-france/grid2op. A more extensive documentation is also publicly available and updated frequently at https://grid2op.readthedocs.io/en/latest/ to understand the framework in more details and use it at its full potential.

6.3 Additional Material: Starting kit, Baselines, Playground and GUI

In order to reduce the the learning curve for participants to enter such a competition, we provide various additional materials to everyone.

Starting Kit This kit contains a few explanatory notebooks (explaining the context, the goal of the competition and also the submission process). It also provides a function to help participants prepare on their local machine a submission for their agent, to further be submitted successfully on the Codalab challenge platform. This comes with a working example.

Baselines In order to help people dive more easily into machine learning, we also made a great effort at implementing some of the most well knonw Reinforcement Learning algorithm in a dedicated python package available at https://github.com/rte-france/l2rpn-baselines that also benefit from an official documentation at https://l2rpn-baselines.readthedocs.io/en/latest/. Using these baselines for the competition is illustrated in the grid2op getting started and is made as easy as possible (see figure 13 for an example). This repository can also grow with many other models through anyone's contribution by using the L2RPN baseline template.

Playground New participants can also get familiar with the problem, the platform, the data or the submission process by registrering into the less complex competitions that are already live: a sandbox competition running at https://competitions.codalab.org/competitions/24493 on a small IEEE 14 grid case & and the latest WCCI competition running at https://competitions.codalab.org/competitions/24902 on a larger IEEE grid case.

GUI (Graphical User Interface) To help diagnose the performance of the agents on this specific power network domain, we also developed a full graphical user interface Grid2Viz [18] shown on Figure 14 . This GUI lets users easily explore scenarios characteristics and compare the performance of several agents on those. It allows to also inspect specific actions taken at some timestep, to understand the context of this decision, and hence better assess the behavior of an agent. Beside understanding the pitfalls of some agent on its episodes, consistent and predictable behavior is in the end be essential to create trust with operators

[18]Grid2Viz GUI from https://github.com/mjothy/grid2viz

```
import grid2op
from grid2op.Reward import L2RPNReward
from l2rpn_baselines.utils import TrainingParam, NNParam
from l2rpn_baselines.DuelQSimple import train

# define the environment
env = grid2op.make("l2rpn_case14_sandbox",
                   reward_class=L2RPNReward  # change the reward if you want
                   )

# use the default training parameters
tp = TrainingParam()

# this will be the list of what part of the observation I want to keep
# more information on https://grid2op.readthedocs.io/en/latest/observation.html#main-observation-attributes
li_attr_obs_X = ["rho", "timestep_overflow", "line_status"]

# neural network architecture
observation_size = NNParam.get_obs_size(env, li_attr_obs_X)
sizes = [800, 800, 800, 494, 494, 494]  # sizes of each hidden layers
kwargs_archi = {'observation_size': observation_size,
                'sizes': sizes,
                'activs': ["relu" for _ in range(sizes)],  # all relu activation function
                "list_attr_obs": li_attr_obs_X}

# select some part of the action
# more information at https://grid2op.readthedocs.io/en/latest/converter.html#grid2op.Converter.IdToAct.init_converter
kwargs_converters = {"all_actions": None,
                     "set_line_status": False,
                     "change_bus_vect": True,
                     "set_topo_vect": False
                     }
# define the name of the model
nm_ = "AnneOnymous"

# train it
train(env,
      name=nm_,
      iterations=10000,
      save_path="/WHERE/I/SAVED/THE/MODEL",
      load_path=None,
      logs_dir="/WHERE/I/SAVED/THE/LOGS",
      training_param=tp,
      kwargs_converters=kwargs_converters,
      kwargs_archi=kwargs_archi)
```

Figure 13: Short example on how to train a baseline coming from the l2rn-baselines github package. This will also print, thanks to the tensorboard tool some usefull indicators to help diagnose the training process.

on the longer term, to eventually allow the deployment of such artificial agents on real systems

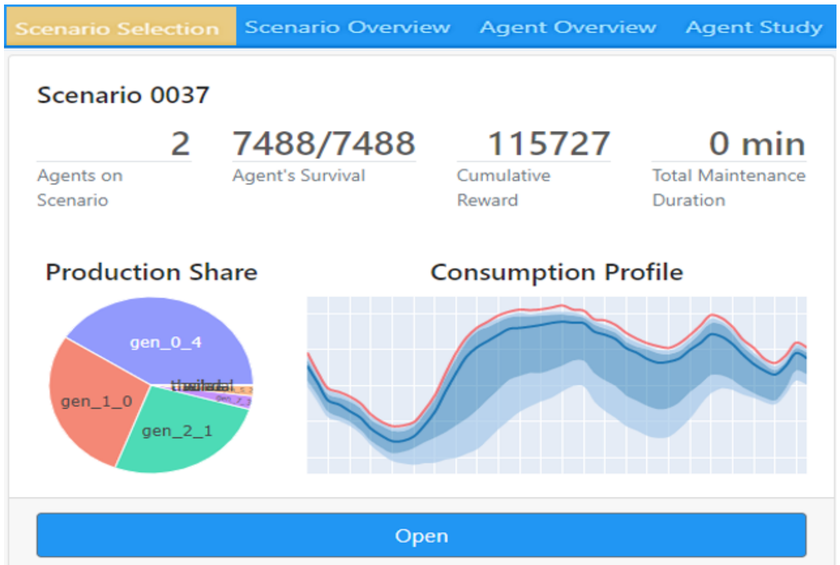


Figure 14: **Grid2viz GUI for scenario & agent analysis**. Screenshot of the “scenario selection” section to identify which episode can be interesting to study after training agents. “scenarios overview”, “agent selection” and “agent study” are the 3 other sections.

In case of all the preceding materials are not enough, some participants may find the need to look deeper into the actual code of the platform. We emphasize that every components used by the platform, the powerflow solver (pandapower), the user interface, etc. are publicly available on github under open source licenses. Participants will then be free to inspect the code, documentation or any other materials put at their disposal in these repositories.

7 Conclusion

In this paper, we have presented the design of ”L2RPN in a sustainable world” competition. This competition target a real-word problem of ensuring the safety of a critical system the powergird with a focus on real-time power network operations. We described the tasks, metrics and all available material to make such a competition engaging, interesting and impactful. We first hope that the AI community will join forces with the power network community to make advances on this critical problem. This competition will hopefully lead to already useful new approaches, possibly advancing on some aspects the related fields of Machine Learning as well. It could develop as a new benchmark on the longer term. We finally hope this will help the research community to design other such real-world challenges in the future.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[2] Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. In *Advances in neural information processing systems*, pages 689–699, 2017.

[3] Florin Capitanescu. Critical review of recent advances and further developments needed in ac optimal power flow. *Electric Power Systems Research*, 136:57–68, 2016.

[4] Hatim Djelassi, Stéphane Fliscounakis, Alexander Mitsos, and Patrick Panciatici. Hierarchical programming for worst-case analysis of power grids. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7. IEEE, 2018.

[5] Benjamin Donnot, Isabelle Guyon, Marc Schoenauer, Patrick Panciatici, and Antoine Marot. Introducing machine learning for power system operation support. *arXiv preprint arXiv:1709.09527*, 2017.

[6] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

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

[8] William H Guss, Cayden Codel, Katja Hofmann, Brandon Houghton, Noboru Kuno, Stephanie Milani, Sharada Mohanty, Diego Perez Liebana, Ruslan Salakhutdinov, Nicholay Topin, et al. The minerl competition on sample efficient reinforcement learning using human priors. *arXiv preprint arXiv:1904.10079*, 2019.

[9] Bri-Mathias Hodge, Carlo Brancucci Martinez-Anido, Qin Wang, Erol Chartan, Anthony Florita, and Juha Kiviluoma. The combined value of wind and solar power forecasting improvements and electricity storage. *Applied Energy*, 214:1–15, 2018.

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

[11] Adriana Mar, Pedro Pereira, and João F Martins. A survey on power grid faults and their origins: A contribution to improving power grid resilience. *Energies*, 12(24):4667, 2019.

[12] Antoine Marot, Benjamin Donnot, Camilo Romero, Luca Veyrin-Forrer, Marvin Lerousseau, Balthazar Donon, and Isabelle Guyon. Learning to run a power network challenge for training topology controllers. *PSCC Conference 2020, arXiv preprint arXiv:1912.04211*, 2020.

[13] Alexander M Prostejovsky, Christoph Brosinsky, Kai Heussen, Dirk Westermann, Jochen Kreusel, and Mattia Marinelli. The future role of human operators in highly automated electric power systems. *Electric Power Systems Research*, 175:105883, 2019.

[14] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[15] Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems. *IEEE Transactions on Power Systems*, 33(6):6510–6521, 2018.