

# DEEP REINFORCEMENT LEARNING FOR POWER GRID MULTI-STAGE CASCADING FAILURE MITIGATION

000  
001  
002  
003  
004 **Anonymous authors**  
005 Paper under double-blind review  
006  
007  
008  
009

## ABSTRACT

010 Cascading failures in power grids can lead to grid collapse, causing severe dis-  
011 ruptions to social operations and economic activities. In certain cases, multi-  
012 stage cascading failures can occur. However, existing cascading-failure-mitigation  
013 strategies are usually single-stage-based, overlooking the complexity of the multi-  
014 stage scenario. This paper treats the multi-stage cascading failure problem as a  
015 reinforcement learning task and develops a simulation environment. The rein-  
016 forcelement learning agent is then trained via the deterministic policy gradient al-  
017 gorithm to achieve continuous actions. Finally, the effectiveness of the proposed  
018 approach is validated on the IEEE 14-bus and IEEE 118-bus systems.  
019

## 1 INTRODUCTION

020 The modern large power grid consists of thousands of generators, substations, and transmission  
021 lines, all intricately interconnected and interdependent, working together to maintain the stable trans-  
022 mission of electricity. However, during the operation of the power system, various events may occur,  
023 among which *cascading failures* is particularly complex and highly damaging Chen et al. (2019);  
024 Jyoti & Hayat (2023); Uwamahoro & Eftekharnejad (2023). Cascading failures in power systems  
025 are typically triggered by the failure of a single component, e.g., a transmission line. These faults  
026 can rapidly *propagate* through the tightly interconnected network, potentially causing severe distur-  
027 bances across the entire power grid and even leading to a complete system collapse Li & Tse (2024);  
028 Zhang et al. (2023); Li et al. (2024). Such kind of events can pose a significant threat to the security  
029 of the power grids and result in severe social and economic consequences.  
030

031 Cascading failures can lead to devastating outcomes Guo et al. (2017); Salehpour & Al-Anbagi  
032 (2024). For example, on June 19, 2024, at approximately 15:17, Ecuador experienced a nationwide  
033 blackout, resulting in a collapse of the nation’s power grid, affecting around 18 million people, with  
034 the power outage lasting for approximately 3 hours. The direct cause of this incident was the failure  
035 of the Milagro-Zhoray transmission line, which triggered a series of cascading failures, ultimately  
036 resulting in a widespread outage. This severe outage underscores the importance of developing fast  
037 cascading failure mitigation strategies for complex power grids.  
038

039 In the power system area, cascading failure mitigation refers to a series of control actions to prevent  
040 the chain reaction after the first fault (e.g., one-line tripping), thereby avoiding system-wide black-  
041 outs. In recent years, numerous studies have emerged in this field. For example, Guo et al. (2024)  
042 proposed a method combining transient stability analysis with interaction graphs to identify critical  
043 lines and mitigate cascading failures by reducing the fault probability of components on these critical  
044 lines. Li et al. (2023) applied network flow theory to study the process of power flow redistribution  
045 and proposed a cascading failure mitigation strategy based on adaptive power balance recovery and  
046 selective edge protection. Inspired by the propagation patterns of faults, Bhaila & Wu (2024) em-  
047 ployed graph neural networks (GNNs) to model and analyze cascading failures in power grids using  
048 an end-to-end approach. Liu et al. (2024), on the other hand, utilized an improved percolation theory  
049 to analyze the survivability of nodes in power grids and proposed an effective mitigation strategy.  
050

051 In this paper, a deep reinforcement learning (DRL) approach is developed for mitigating multi-stage  
052 cascading failures (MSCF) in power systems, with the following contributions: (1) A simulation  
053 environment for multi-stage cascading failure study is constructed; (2) The Deep Deterministic Pol-  
icy Gradient (DDPG) algorithm is adopted to address the MSFC issue; (3) The proposed model is  
validated on the IEEE 14-bus and 118-bus systems, demonstrating its effectiveness.

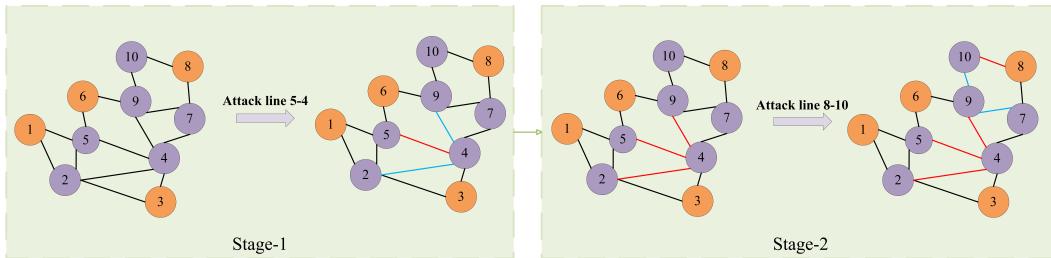


Figure 1: An example of a multi-stage cascading failure.

## 2 METHODOLOGY

### 2.1 MULTI-STAGE CASCADING FAILURE (MSCF) PROBLEM

Traditionally, single-stage cascading failure problems have been well studied Qi et al. (2017). However, in certain situations, multiple stages may occur Zhu (2021). For example, Fig. 1 depicts a multi-stage cascading failure example: an earthquake causes the loss of the power line 4-5, triggering the first stage of cascading failures (lines 2-4 and 4-9 are subsequently tripped due to the over-limit line power flow, *after* the loss of line 4-5). Suppose the (remaining) power grid does not collapse and enters a steady state. Then, after a short period, the aftershock may break another line, triggering *another stage* of cascading failures.

One approach to handling the MSCF problem is to decompose them into multiple sub-problems of single-stage and then solve each by each. However, this way might overlook the interdependence between stages. On the other hand, if we map the concept of “each *stage*” to the concept of “each *step*” in the RL context, then the MSCF problem can be investigated *holistically* under various mature frameworks of reinforcement learning, which is the motivation of this paper.

In this paper, the DDPG algorithm and the Actor-Critic framework are utilized Lillicrap et al. (2019), Mnih et al. (2016). The output of DDPG can be deterministic and real-number valued; hence, it performs well in solving problems with continuous actions Wang & Vittal (2023).

### 2.2 ENVIRONMENT IMPLEMENTATION

In our work, a simulation environment is developed for MSCF mitigation using Python and Matpower, which is a well-known MATLAB toolbox for AC power flow (ACPF) computation. Cross-tool interaction and data communication between Python and MATLAB have been achieved via a Python-MATLAB handler. Several key designs regarding this environment are described below.

#### 2.2.1 DEFINITIONS OF STEP AND EPISODE

**Step:** a step means a **stage** when the power grid is attacked (e.g., by natural disasters), causing the grid to evolve into a new state (i.e., how many buses (i.e., nodes) and “lines” (i.e., edges) are still “available”; how many islands are formed; how large is the power flow on each remaining line; etc.).

**Episode:** an episode is one specific **set of steps** when the power grid is consecutively attacked. At the end of each episode, the final status is either “Win” or “Lose ” (cf. definitions in later sections).

#### 2.2.2 STATE DESIGN

For an  $n$ -bus power grid, our *state* is defined as follows:

$$state = [line\_status, P_1, Q_1, V_1, \theta_1, \dots, P_n, Q_n, V_n, \theta_n]$$

where, *line\_status* is the percentage value obtained via dividing the actual line power flow by its maximum limit;  $P_i, Q_i, V_i, \theta_i, (i = 1, \dots, n)$  denotes the active power injection, reactive power injection, voltage magnitude and angle of the  $i$ -th bus, respectively.

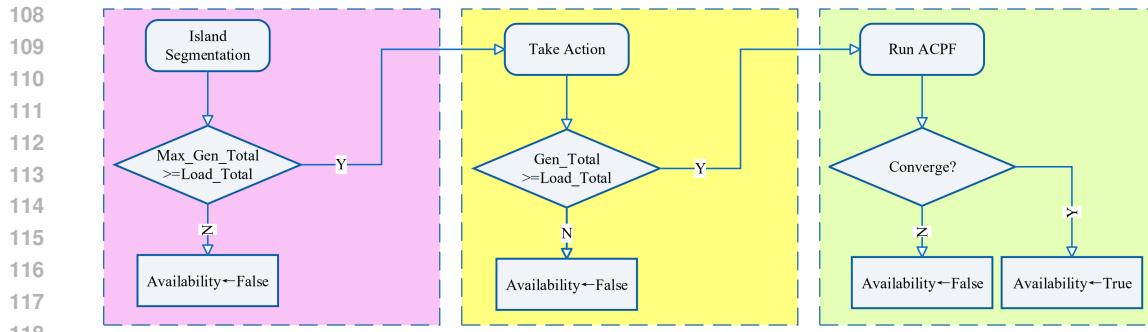


Figure 2: Island Availability assessment.

### 2.2.3 ACTION DESIGN

Cascading failure might be mitigated by adjusting the generator’s power generation. Thus, the generation coefficients  $[a_1, \dots, a_m]$  of all  $m$  generators are considered as the *action*. The power output of the  $i$ -th generator is the product of  $a_i$  and its power capacity (i.e., the maximum power).

### 2.2.4 ISLAND DETECTION AND AVAILABILITY ASSESSMENT

The grid can become disconnected when lines are lost (due to an incident or line overload). Therefore, the first step is to assess the connectivity of the grid. To that end, we employ the union-find algorithm (c.f. Appendix A.1) to locate all the remaining islands.

The “availability” of an island means whether it is still *alive* at the end of a specific cascading failure stage; if not, it will be discarded in later stages. The availability assessment is carried out after the island detection. The criteria for island availability are described in Fig. 2. *Max\_Gen\_Total* and *Gen\_Total* are respectively the total power capacity and the total actual power output of all the remaining generators in a specific island, and *Load\_Total* is the total load demand in that island.

### 2.2.5 REWARD DESIGN

- **Total cost of generation:**  $-c_1 \cdot cost$ . Here,  $c_1$  is a hyperparameter. *cost* means the total generation cost (\$) of all islands whose *availability* is true.
- **Loss of load penalty:**  $-BaseReward_1 \cdot P_{loss}/P_{total}$ .  $P_{loss}$  is the total load on *unavailable* islands at current stage, while  $P_{total}$  represents the original total load of initial power grid.
- **Convergence reward:**  $BaseReward_2$ . This reward is given when half or more of all the currently remaining islands have converged.
- **Win reward:**  $BaseReward_3 \cdot (P_{available}/P_{total})^{c_2}$ . This reward is given when the win conditions are met.  $P_{available}$  is the total load of *available* islands.

Here,  $c_1, c_2, BaseReward_1, BaseReward_2$ , and  $BaseReward_3$  are constants related to a specific power grid. A basic idea in picking those constants is to make the above four parts in the same order of magnitude. Finally, the overall workflow for the RL-based MSCF study is shown in Fig. 3(a).

## 3 EXPERIMENTS AND RESULTS

The proposed approach is tested on the IEEE 14-bus and modified IEEE 118-bus systems. The IEEE 14-bus system has 5 generators and 20 lines, with its topology shown in Fig. 3(b). For other details about the experiment settings and hyperparameters, please refer to Appendix A.5.

For each power grid, a DRL model is trained for 300 episodes. After training, the model interacted with the environment for an additional 1000 episodes, during which the total reward in each episode is recorded, and the final win rate is computed.

The model is compared with three baseline strategies, as shown in Table 1 and Fig. 4. **Baseline 1** means each generator output a random power. **Baseline 2** means all generators output the maximum power. **Baseline 3** means all generators operate at half of their maximum power output. It can

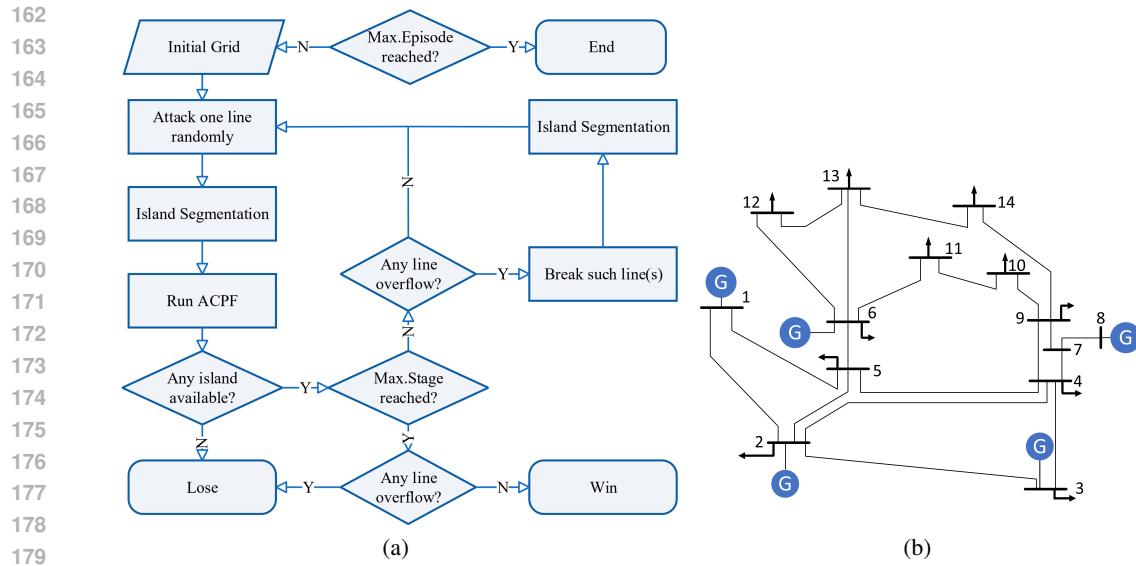


Figure 3: (a) The overall workflow of grid simulation for MSCF study; (b) The IEEE 14-bus system.

Table 1: Win rate comparison.

Method	IEEE 14-bus System	IEEE 118-bus System
DDPG	<b>95.5%</b>	<b>97.8%</b>
Baseline 1	52.0%	51.7%
Baseline 2	93.3%	8.40%
Baseline 3	85.6%	97.0%

be observed that the DRL achieves a good performance, with the highest win rate, large average rewards, and more stable behaviors.

## 4 CONCLUSION

This paper implements and validates a DRL-based solution for multi-stage cascading failure mitigation. One limitation of the current solution is that the differences in the states are relatively small, causing the majority of the model’s actions to be similar. In future work, we will explore other state designs to improve the action’s variability.

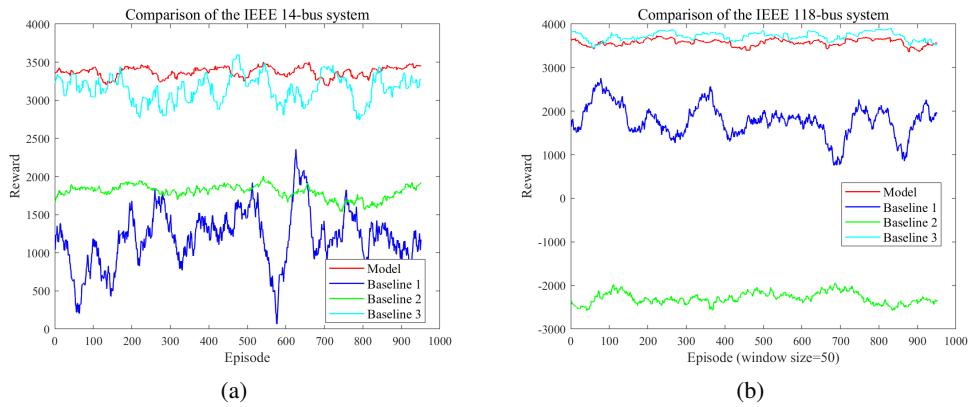


Figure 4: The moving-average reward comparison.

216 REFERENCES  
217

- 218 Karuna Bhaila and Xintao Wu. Cascading failure prediction in power grid using node and edge  
219 attributed graph neural networks. In *2024 IEEE Green Technologies Conference (GreenTech)*, pp.  
220 155–156, 2024. doi: 10.1109/GreenTech58819.2024.10520535.
- 221 Changsheng Chen, Wenyun Ju, Kai Sun, and Shiying Ma. Mitigation of cascading outages using  
222 a dynamic interaction graph-based optimal power flow model. *IEEE Access*, 7:168637–168648,  
223 2019. doi: 10.1109/ACCESS.2019.2953774.
- 224 Hengdao Guo, Ciyan Zheng, Herbert Ho-Ching Iu, and Tyrone Lucius Fernando. A critical re-  
225 view of cascading failure analysis and modeling of power system. *Renewable & Sustainable En-*  
226 *ergy Reviews*, 80:9–22, 2017. URL [https://api.semanticscholar.org/CorpusID:  
227 114562742](https://api.semanticscholar.org/CorpusID:114562742).
- 228 Zhenping Guo, Xiaowen Su, Kai Sun, and Srdjan Simunovic. Analysis and mitigation of cascading  
229 outages using an interaction graph addressing transient stability. In *2024 IEEE Power Energy  
230 Society General Meeting (PESGM)*, pp. 1–5, 2024. doi: 10.1109/PESGM51994.2024.10689239.
- 231 Jamir Shariar Jyoti and Majeed M. Hayat. Topological attributes of cascading failures in power  
232 grids. In *2023 IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, 2023. doi:  
233 10.1109/PESGM52003.2023.10252476.
- 234 Biwei Li, Dong Liu, Junyuan Fang, Xi Zhang, and Chi K. Tse. Strengthening critical power network  
235 branches for cascading failure mitigation. In *2024 IEEE International Symposium on Circuits and  
236 Systems (ISCAS)*, pp. 1–5, 2024. doi: 10.1109/ISCAS58744.2024.10558306.
- 237 Meixuan Jade Li and Chi K. Tse. Quantification of cascading failure propagation in power systems.  
238 *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(8):3717–3725, 2024. doi:  
239 10.1109/TCSI.2024.3383450.
- 240 Meixuan Jade Li, Chi Kong Tse, Dong Liu, and Xi Zhang. Cascading failure propagation and  
241 mitigation strategies in power systems. *IEEE Systems Journal*, 17(2):3282–3293, 2023. doi:  
242 10.1109/JYST.2023.3248044.
- 243 Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa,  
244 David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.  
245 URL <https://arxiv.org/abs/1509.02971>.
- 246 Xinyu Liu, Yan Li, and Tianqi Xu. Cascading failure model of cyber-physical power systems con-  
247 sidering overloaded edges. In *2024 IEEE 2nd International Conference on Power Science and  
248 Technology (ICPST)*, pp. 982–987, 2024. doi: 10.1109/ICPST61417.2024.10601896.
- 249 Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim  
250 Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement  
251 learning. *CoRR*, abs/1602.01783, 2016. URL <http://arxiv.org/abs/1602.01783>.
- 252 Junjian Qi, Wenyun Ju, and Kai Sun. Estimating the propagation of interdependent cascading out-  
253 ages with multi-type branching processes. *IEEE Transactions on Power Systems*, 32(2):1212–  
254 1223, 2017. doi: 10.1109/TPWRS.2016.2577633.
- 255 Ali Salehpour and Irfan Al-Anbagi. Resp: A real-time early stage prediction mechanism for cas-  
256 cading failures in smart grid systems. *IEEE Systems Journal*, 18(3):1593–1604, 2024. doi:  
257 10.1109/JYST.2024.3420950.
- 258 Nathalie Uwamahoro and Sara Eftekharnejad. A comparative study of data-driven power grid cas-  
259 cading failure prediction methods. In *2023 North American Power Symposium (NAPS)*, pp. 1–6,  
260 2023. doi: 10.1109/NAPS58826.2023.10318537.
- 261 Yuling Wang and Vijay Vittal. Real-time excitation control-based voltage regulation using ddpg  
262 considering system dynamic performance. *IEEE Open Access Journal of Power and Energy*, 10:  
263 643–653, 2023. doi: 10.1109/OAJPE.2023.3331884.

270 Xinzhe Zhang, Wenping Qin, Xiang Jing, Jiaxin Liu, Xiaoqing Han, and Peng Wang. Power  
271 system resilience assessment considering the occurrence of cascading failures. In *2023 International Conference on Power System Technology (PowerCon)*, pp. 1–5, 2023. doi: 10.1109/  
272 PowerCon58120.2023.10331511.  
273  
274

275 Yongli Zhu. Power grid cascading failure mitigation by reinforcement learning. In *ICML 2021  
276 Workshop on Tackling Climate Change with Machine Learning*, 2021. URL <https://www.climatechange.ai/papers/icml2021/30>.  
277  
278  
279

280 **A APPENDIX**  
281

282 **A.1 THE UNION-FIND ALGORITHM FOR POWER GRID ISLAND DETECTION**  
283

284 The union-find algorithm is a data structure used to handle dynamic connectivity problems. Its basic  
285 idea is to determine whether elements belong to the same set recursively and to merge sets when  
286 necessary. Based on the results of island detection, the original grid may need to be divided into  
287 multiple islands, which provides the basis for later evaluation of the system status.  
288

---

290 **Algorithm 1:** Island Detection

---

291 **Input:** A power grid  $G$  with bus set  $N$  and line set  $E$   
292 **Output:** Islands  $I$   
293 1 Initialize an array  $p$  such that  $p[n] \leftarrow n$  for all  $n$ ;  
294 2 **for**  $(u,v)$  in  $E$  **do**  
295 | Perform  $Union(u, v, p)$  to merge their sets;  
296 4 **end**  
297 5 **for**  $n$  in  $N$  **do**  
298 | Perform  $Find(n)$  to determine the root;  
299 7 **end**  
8 Group all buses by their root into disjoint sets  $I$ ;

---

301  
302 **A.2 THE TOPOLOGY OF THE IEEE 118-BUS SYSTEM**  
303

304 The topology of the IEEE 118-bus system is shown in Fig. 5.  
305  
306

307 **A.3 REWARD COMPARISON**  
308

309 The reward comparison is shown in Fig. 6.  
310  
311

312 **A.4 CROSS-TOOL INTERACTION**  
313

314 The process of cross-tool interaction is shown in Fig. 7.  
315  
316

317 **A.5 EXPERIMENT SETTINGS AND HYPERPARAMETERS**  
318

319 The experiments are carried out on a computer with an Intel Core i5-12400F CPU, 32 GB RAM,  
320 and a GeForce RTX 4060ti GPU. The development environments are Python 3.11, PyTorch 2.3.1,  
321 and MATPOWER 8.0.

322 The IEEE 118-bus system contains 54 generators and 179 lines. Its topology is shown in Fig.  
323 5. The environment parameters for both 14-bus and 118-bus systems are summarized in Table 2.  
324 *stage\_max* represents the maximum number of stages in the MSCF problem, and *line\_limit* refers  
325 to the maximum allowed power flow on the lines.

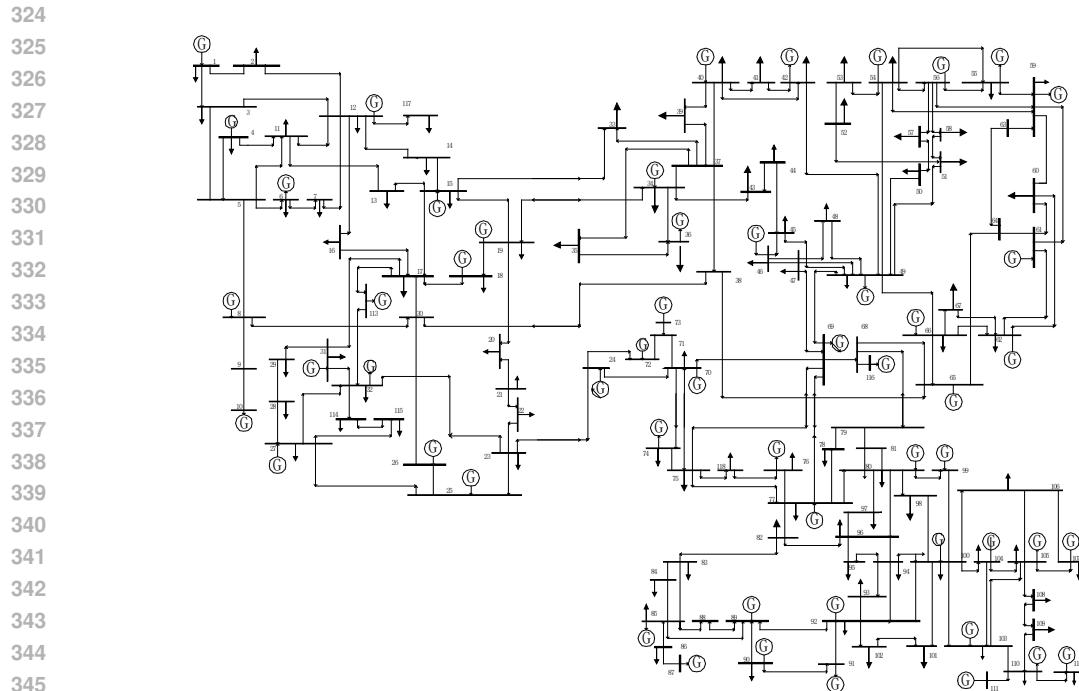


Figure 5: The topology of the IEEE 118-bus system.

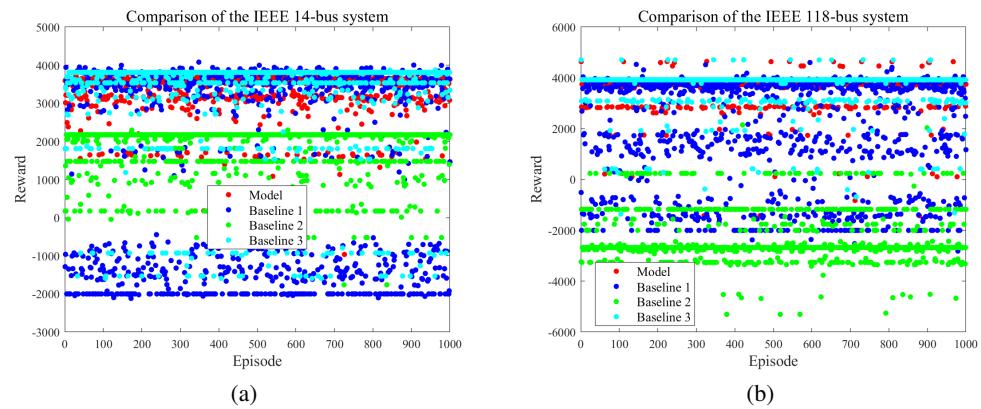


Figure 6: The reward comparison.

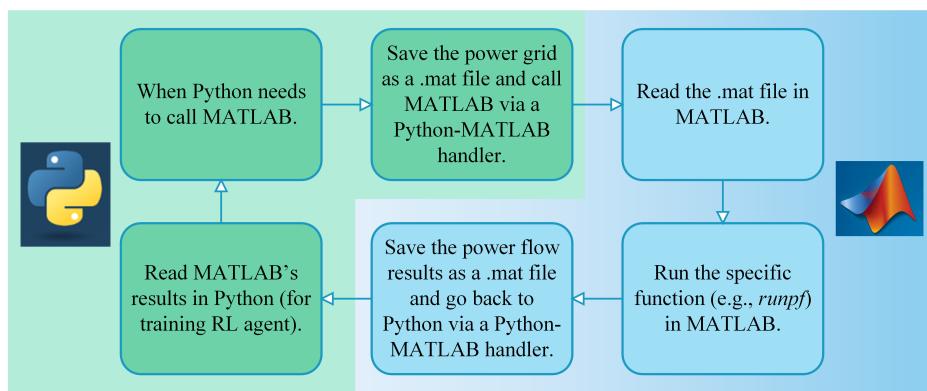


Figure 7: The process of cross-tool interaction.

378

Table 2: Environment parameters.

379

380

Parameter	IEEE 14-bus System	IEEE 118-bus System
<i>stage_max</i>	3	3
<i>line_limit</i>	200	450
$c_1$	0.03	0.005
$c_2$	1.7	1.7
$BaseReward_1$	2000	2000
$BaseReward_2$	1000	1000
$BaseReward_3$	2000	2000

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

Table 3: Model parameters.

Parameter	Value
learning rate	$1 \times 10^{-4}$
batch size	128
discount factor( $\gamma$ )	0.99
update rate( $\tau$ )	0.001