Grid Mind

-A Data-driven Autonomous Grid Dispatch and Control Framework Based on PMU Measurements

Di Shi

Team: Ruisheng Diao, Jiajun Duan, Bei Zhang, Zhe Yu, Zhiwei Wang

GEIRI North America (GEIRINA)

April 26, 2019

@ CURENT Power and Energy Industry Seminar

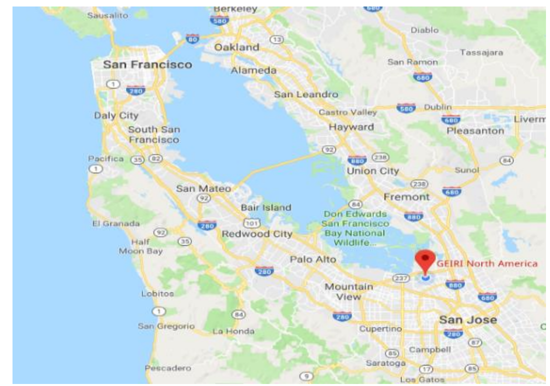
GEIRI North America (GEIRINA)

Introduction

- Founded in Dec. 2013 in Santa Clara, California, USA (www.geirina.net)
- Conducts cross-disciplinary R&D for power system modernization
- R&D subsidiary and overseas platform of SGCC
- ~50 Researchers and Engineers (70-80 in summer)
- Mentored over 60 graduate students in the past 3 years

Research Groups & Areas

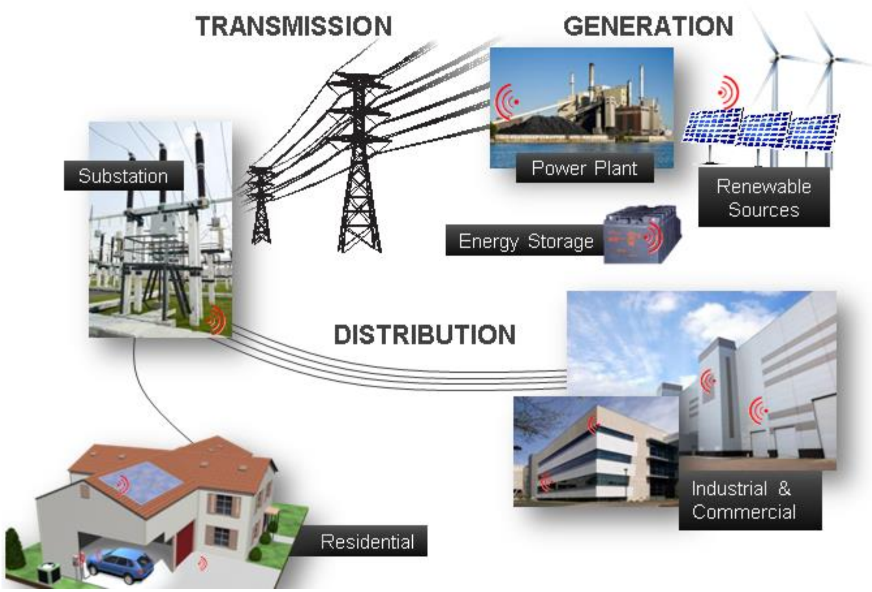
- Graph computing & Grid Modernization
- AI & System Analytics
- Advanced Computing & Data Intelligence
- Smart Chips

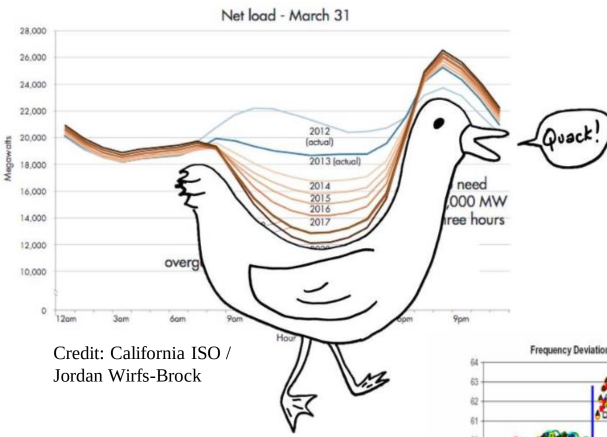


Outline

- Background and motivation
- Autonomous grid dispatch and control based on PMU measurements
    - Deep Reinforcement Learning
    - Autonomous voltage control
    - Demo
    - AC OPF?
- How to architect/tune an effective self-learning agent?
- Discussion/other applications

Challenges

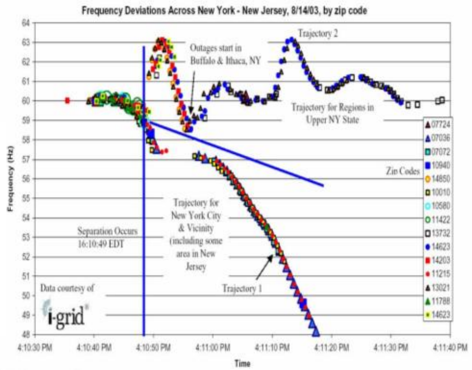




Credit: California ISO / Jordan Wirfs-Brock

The well-known Californian duck curves showing abrupt changes in system net load

System fast dynamic responses under extreme events – the August 2003 North American Blackout



Grand challenges: the increasing dynamics and stochastics in the modern power grid, making it difficult to design and implement optimal control actions in real time

- Increased penetration of renewable energy
- Demand response
- New market behavior
- Energy storage
- Experience/model based control suggestions using limited studied cases are either conservative or risky for operation

Need for accurate and fast wide-area monitoring system to detect potential issues

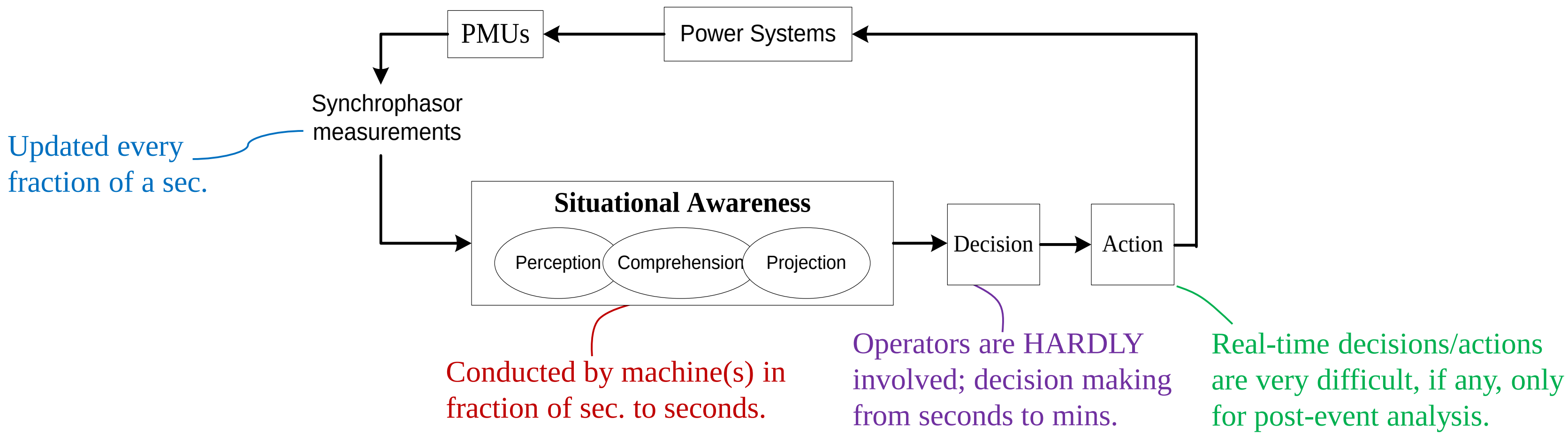
- PMU coverage is increasing, but still limited
- Known data quality issues affect apps
- Lack of preventive measures to mitigate operational risks

Need for effective optimal control suggestions in real time to support operators

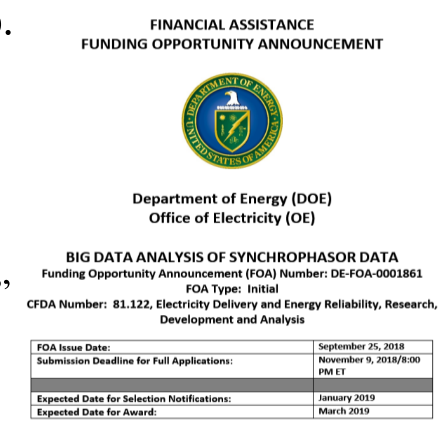
- Most operational rules are offline determined
- Either by experiences or projected simulations

The Gap

- Past efforts were mostly focused on enhancing/increasing grid situational awareness using advanced modeling, various data analysis approaches, machine learning, etc.
- Very few WAMS apps can instruct operators what to do in real time due to the lack of effective approaches that can transform massive amount of measurements directly into actionable decisions in real time.



- ❑ Potential apps of WAMS are limited, and GEIRINA wish to bridge this gap.
- ❑ On Sept. 25 2018, DOE announced investments to improve resilience and reliability of the nation's energy infrastructure using PMU measurements and big data, AI, machine learning technologies.
  - ✓ "…to inform and shape development and application of **fast grid analytics** and **sub-second automatic control** actions that preclude costly cascading grid outages"
  - ✓ "…**PMU-based automated controls**, better grid asset management, and real time monitoring for modeling…"



FINANCIAL ASSISTANCE
FUNDING OPPORTUNITY ANNOUNCEMENT

Department of Energy (DOE)
Office of Electricity (OE)

BIG DATA ANALYSIS OF SYNCHROPHASOR DATA
Funding Opportunity Announcement (FOA) Number: DE-FOA-0001861
FOA Type: Initial
CFDA Number: 81.122, Electricity Delivery and Energy Reliability, Research, Development and Analysis

FOA Issue Date:	September 25, 2018
Submission Deadline for Full Applications:	November 9, 2018/8:00 PM ET
Expected Date for Selection Notifications:	January 2019
Expected Date for Award:	March 2019

The Grid Mind Vision

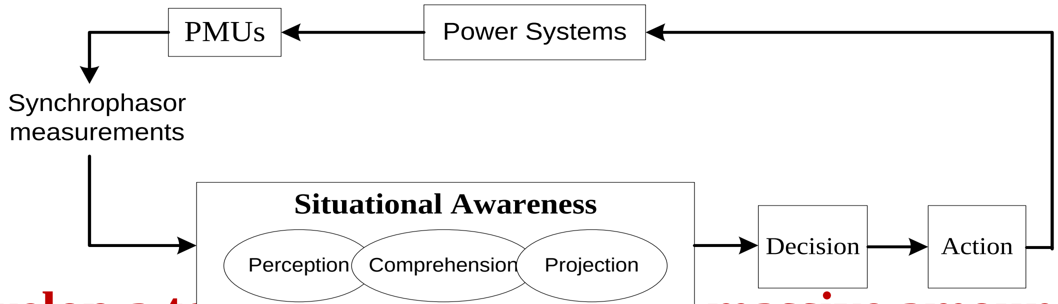
- **Grid Mind: A measurement-driven, grid-interactive, self-evolving, and open platform for power system autonomous dispatch and control.**

❑ In the short term, we want to duplicate an example of AlphaGo Zero in power systems.

❑ In the mid-term, Grid Mind serves as an assistant to grid operators.

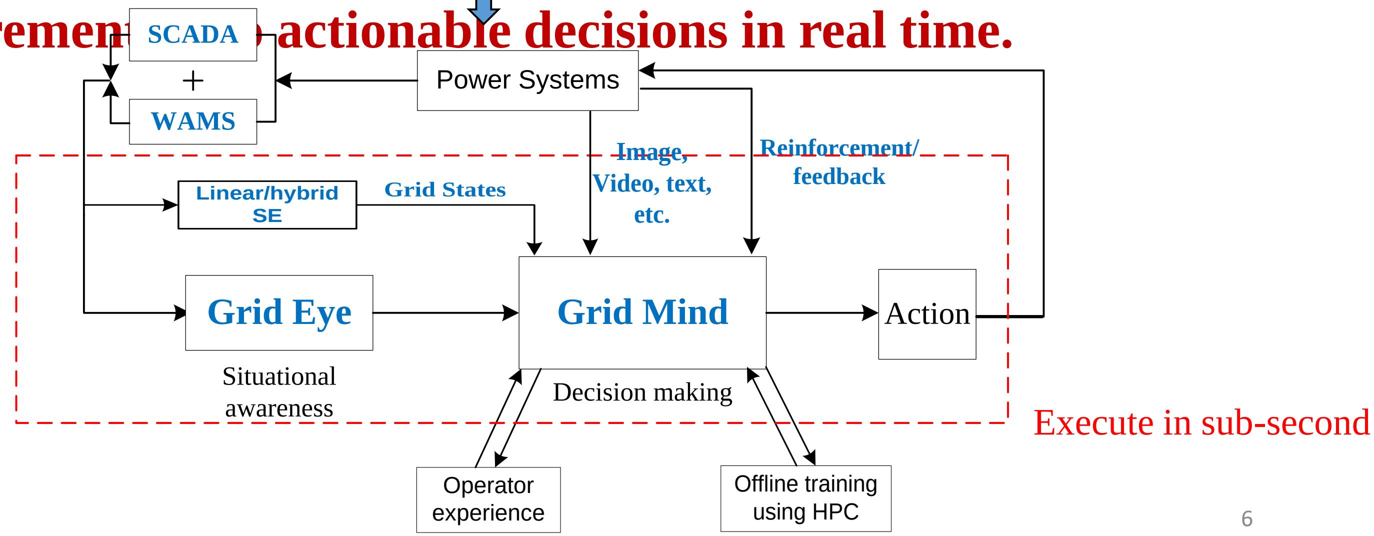
❑ In the long term, Grid Mind will be the core of power system operation ROBOT.

Now



Goal: **To develop a tool that can transform massive amount of measurements to actionable decisions in real time.**

Goal / Future

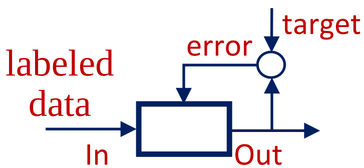


Outline

- Background and motivation
- Autonomous grid dispatch and control based on PMU measurements
  - Deep Reinforcement Learning
  - Autonomous voltage control
  - Demo
  - AC OPF?
- How to architect/tune an effective self-learning agent?
- Discussion/Other applications

ML in a Nutshell

Supervised Learning



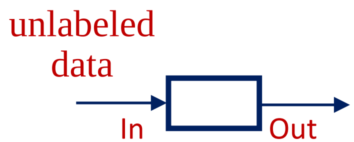
Application

- ✓ Classification
- ✓ Predict a target numeric value

Common Algorithms

- *k*-Nearest Neighbors
- Linear Regression
- Decision Trees
- Naïve Bayes
- SVM
- Neural Networks

Unsupervised Learning



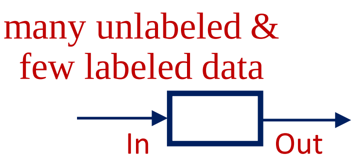
Application

- ✓ Clustering
- ✓ Visualization
- ✓ Dimensionality reduction
- ✓ Anomaly detection

Common Algorithms

- *k*-Means
- Hierarchical Cluster Analysis
- Principal Component Analysis

Semi-supervised Learning



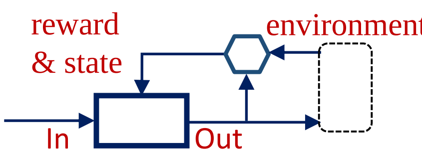
Application

- ✓ Google Photos
- ✓ Webpage classification

Common Algorithms

- Combination of unsupervised and supervised learning

Reinforcement Learning



Application

- ✓ DeepMind's AlphaGo
- ✓ Fire-extinguish robots
- ✓ Grid Mind

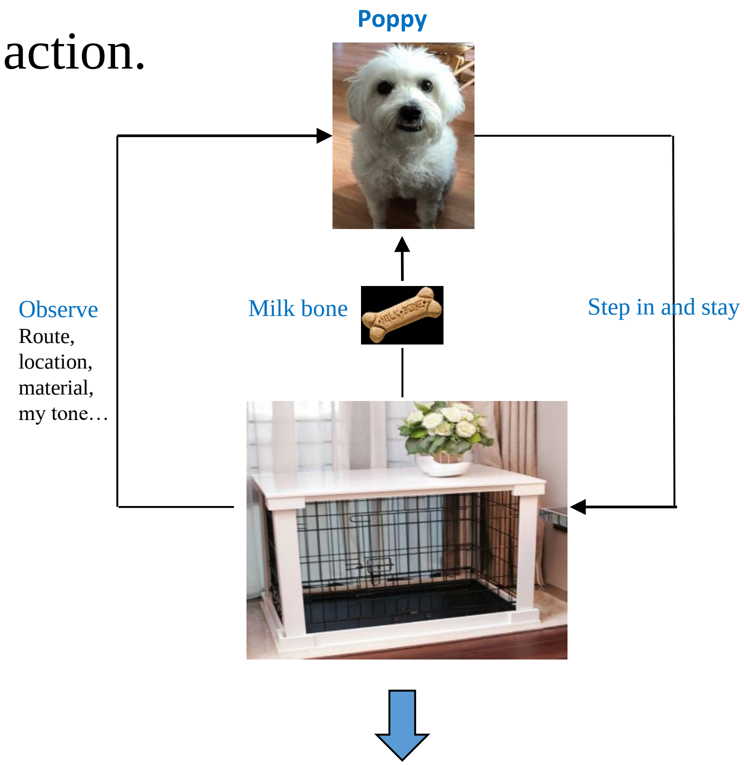
Common Algorithms

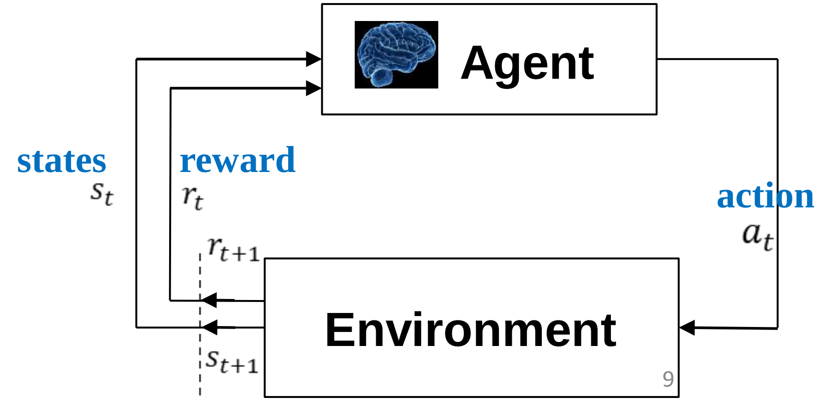
- Dynamic programming
- Monte Carlo
- Temporal Difference (TD)
  - ✓ Q-Learning
  - ✓ SARSA

Credit: Elena Mocanu, TU/e

Reinforcement Learning (RL)

- Learn what to do and how to map situation to action.
- Poppy's example.
- The RL system: agent and environment. At each time step t :
  - The agent
    - ✓1) executes action a_t
    - ✓2) observes states s_t
    - ✓3) receives a scalar reward r_t
  - The environment
    - ✓1) receives action a_t
    - ✓2) emits states s_{t+1}
    - ✓3) issues a reward r_{t+1}
- Reinforcement function
  - Trial-and-error interactions
  - Mapping states/action pair to reinforcement
  - Maximization of the sum of reward/value





RL Agent

- An RL agent may include one or more of the following components:
  - **Policy**: agent's behavior function
    - ✓ A map from state to action
      - Deterministic policy $a = \pi(s)$
      - Stochastic policy $\pi(a|s) = \mathrm{P}(a|s)$
  - **Value function**: prediction of future reward
    - ✓ How much reward can be obtained if I perform action *a* in state *s*
  - **Model**: agent's representation of the environment
- Q-value function gives expected total reward
  - ✓ from state *s* and action *a*
  - ✓ under policy π
  - ✓ with discount factor γ

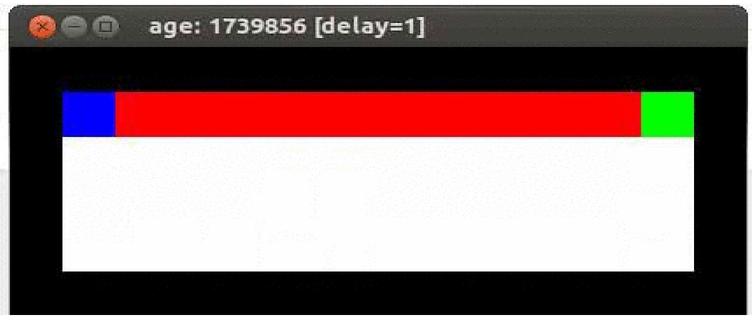
$$Q^{\pi}(s,a) = \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | s,a)$$

- An optimal value function is the maximum achievable value

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) = Q^{\pi^{*}}(s,a)$$

Q-Learning

Example: Mouse vs Cliff[1]



Blue-mouse
Red-cliff
Green-cheese

Q Table

{up, down, right, left}

position/location

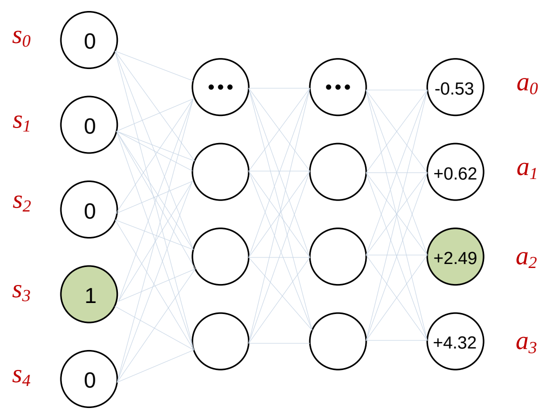
Q(S,A) → Q(3,2) →

	s_0	s_1	s_2	s_3	s_4	…
a_0	+1.53	+0.97	+0.83	-0.53	-0.02	…
a_1	+2.19	+3.85	-1.24	+0.62	+0.19	…
a_2	-0.23	+5.39	+0.62	$\boxed{+2.49}$	+0.82	…
a_3	+0.18	+1.43	+0.65	+4.32	+1.83	…

→ +2.49

Neural Net

Q(S,A) → Q(3,2) →

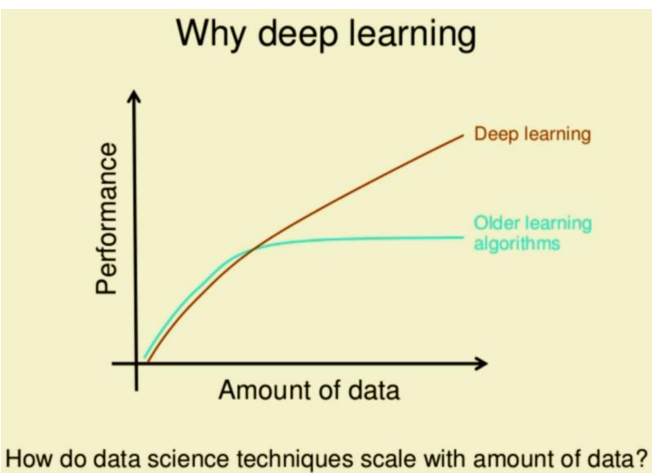


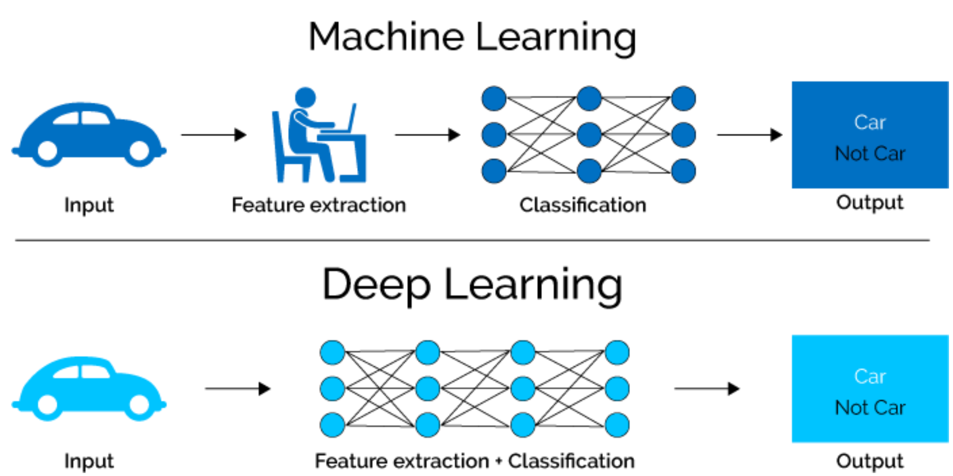
→ +2.49

[1]https://studywolf.wordpress.com/2013/07/01/reinforcement-learning-sarsa-vs-q-learning/

Deep Learning in a Nutshell

- Deep learning is a general-purpose framework for representation learning
  - Given an objective
  - Learn representation that is required to achieve objective
  - Directly from raw inputs
  - Using minimal domain knowledge
  - Represent the world using nested hierarchy of concepts (each using simpler ones)





Source: https://towardsdatascience.com

Deep Reinforcement Learning (DRL)

- DRL=DL+RL
- DL is a general-purpose framework for representation learning
- RL is a general-purpose framework for decision-making in a dynamic environment
- We seek a single agent that can solve a human-level task
  - RL defines the objective
  - DL gives the mechanism
  - RL+DL → general intelligence
- Use deep neural networks to represent
  - Value function
  - Policy
  - Model

Outline

- Background and motivation
- Autonomous grid dispatch and control based on PMU measurements
  - Deep Reinforcement Learning
  - Autonomous voltage control
  - Demo
  - AC OPF?
- How to architect/tune an effective self-learning agent?
- Discussion/Other applications

Autonomous Voltage Control (AVC)

(Considering load variation, renewable intermittency and contingency conditions)

Objective:
Maintain steady-state voltages at all buses within the range of 0.95-1.05pu after disturbance(s) or contingencies from any given initial operating point.

Bus Voltages → Voltage Controller →

- ✓ Change generator voltage set point
- ✓ Switchable shunts on & off
- ✓ Adjust transformer tap ratios

→ All buses stay within a secure range

Challenges for conventional technologies:

- Increasing complexity, e.g., renewable energies
- Increasing scale, e.g., wide-area power systems
- High nonlinearity, e.g., nonlinear loads
- Fast response speed, e.g., power electronics

DRL Formulation for AVC

Firstly, let's define $\boldsymbol{V_i}$ as the voltage phasor of bus i (including both magnitude and phase angle).

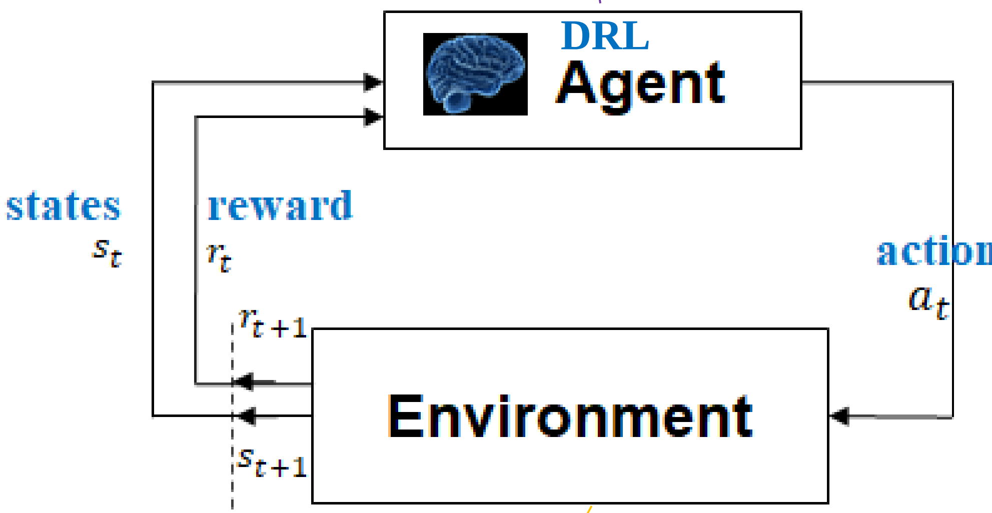
Control objective

All V_i's (of interest) stay within normal operation zone

App of Grid Mind

V_i of buses of interest

P, Q of branches



Generator voltage set points [0.95, 0.975, 1.0, 1.025, 1.05] ,

Shunt cap value, transformer tap, etc.

Power System

Note: Grid Mind does not know the model of the system or its electrical parameters.

AVC Training Algorithm

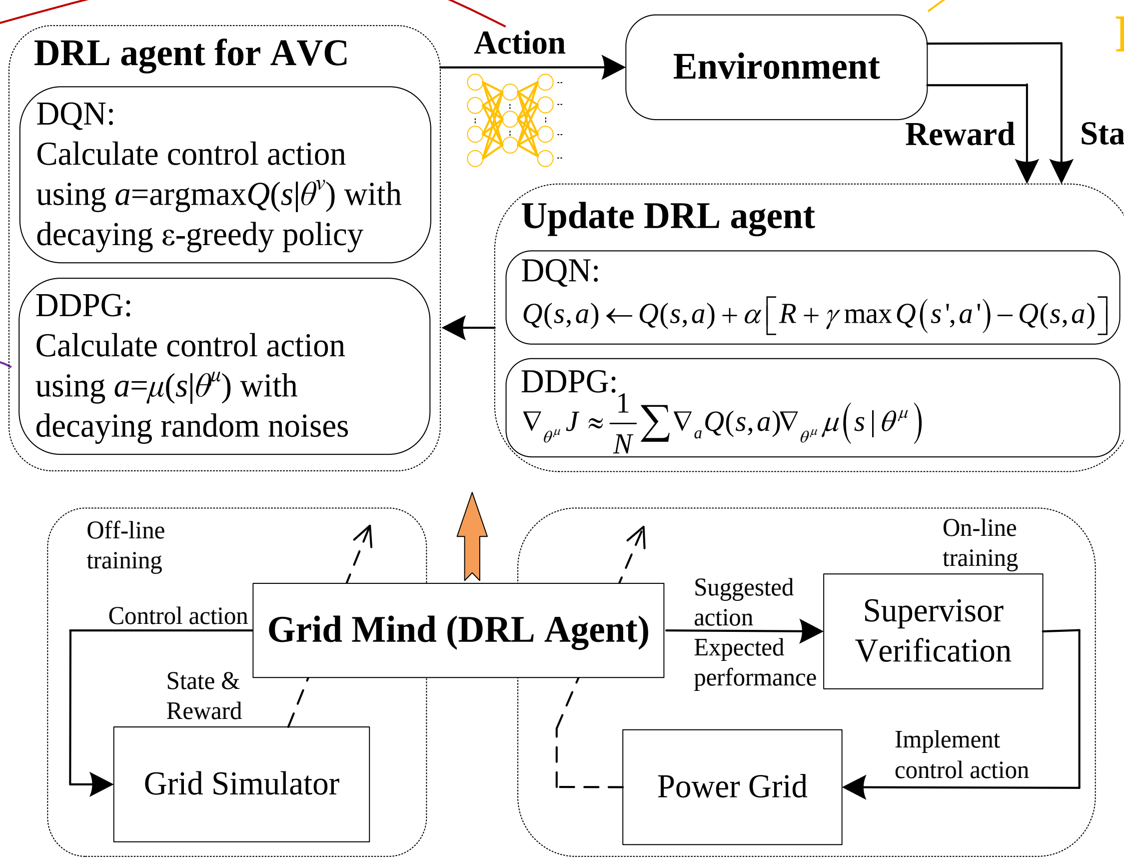
Control objective
All V_i's (of interest) stay within normal operation zone

Generator voltage set points

DQN (discrete): [0.95, 0.975, 1.0, 1.025, 1.05]

DDPG (continuous): 0.95-1.05

App of Grid Mind

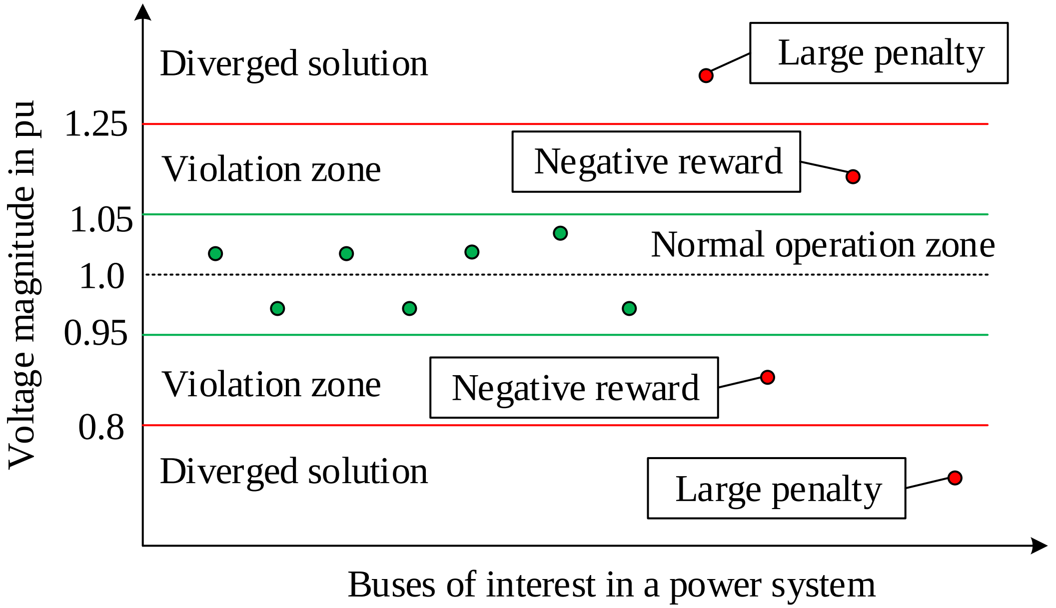


Power System

V_i of buses of interest

P, Q of branches

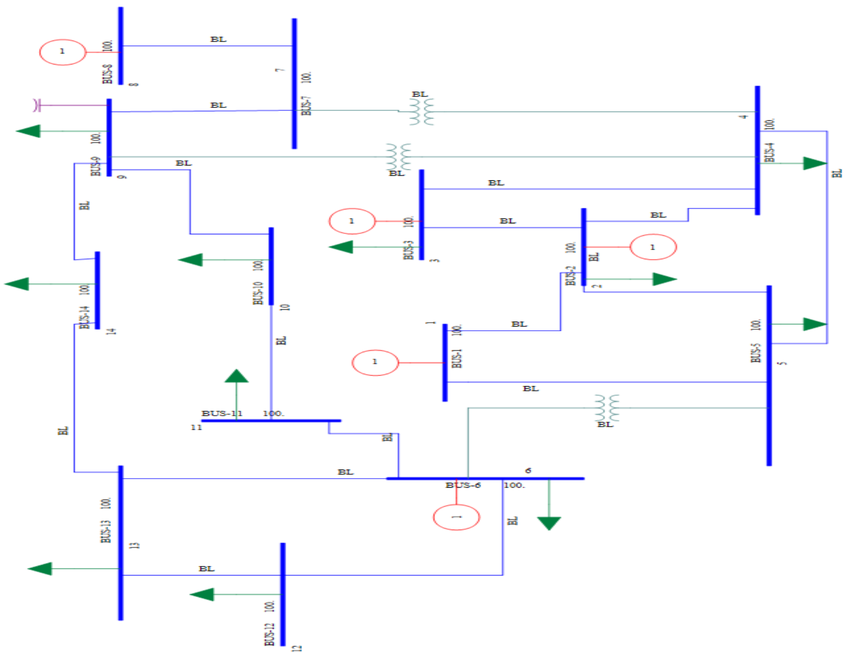
DRL Formulation for Voltage Control-Reward



$$\textbf{Reward} \text{ at one iteration} = \begin{cases} \text{Large reward } (+R_P), & \forall\, V_i \in \text{normal operation zone} \\ \text{Large penalty } (-P_e), & \exists\, V_i \in \text{diverged solution} \\ \text{Negtive Reward } (-R_N), & \exists\, V_i \in \text{violation zone} \end{cases}$$

Final Reward =Sum(**Reward**)/number of iterations

Case Study



Testing system: IEEE 14-Bus system

System Info.

- 14 buses
- 5 generators
- 11 loads
- 17 lines
- 3 transformers
- Active load: 259 MW
- Reactive load: 73.5 MVAr

Testing Condition

- IEEE 14-bus System
- 10k episodes (created randomly)
- 60%~120% random load change
- A single-NN DQN agent
- 2 layers with 20 neurons/layer
- Without using regularization
- 120 action space (permutation of 5 choices)

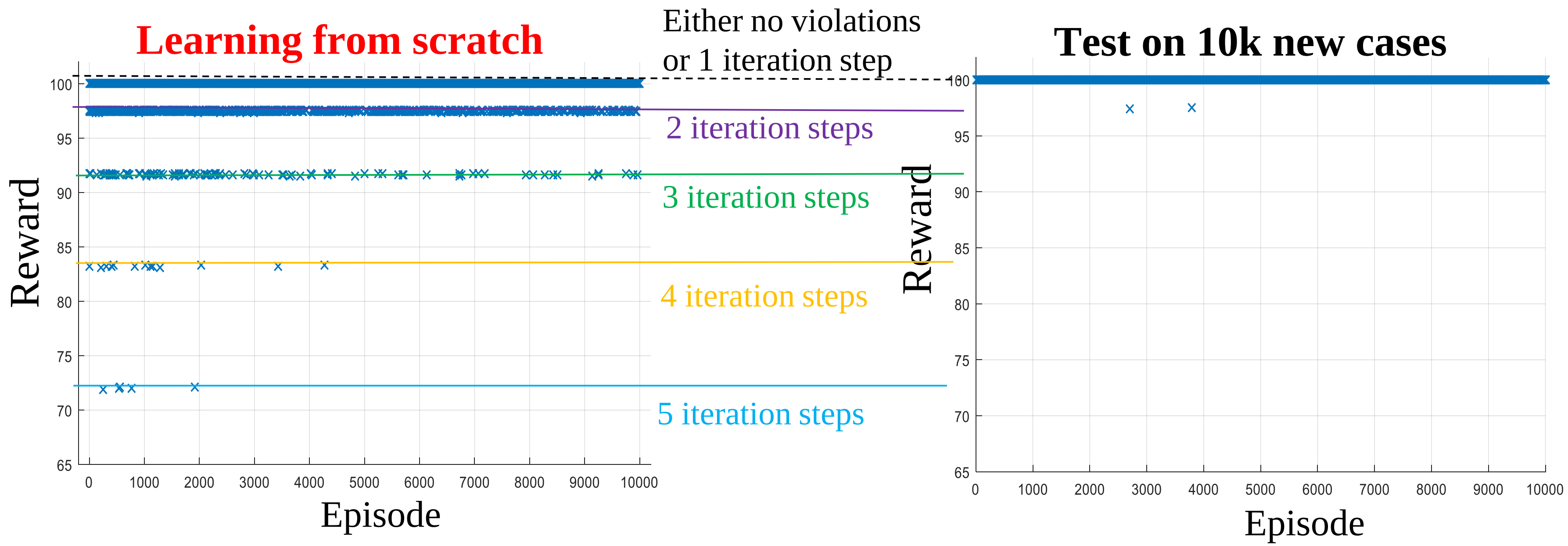
Note: Grid Mind does not know the model of the system or its electrical parameters.

It learns from the scratch

1. Initializing the probability of using random control actions to be $p_r(0)=1$
2. for Episode i
3. $p_r(i+1)=0.95p_r(i)$

DQN Agent for IEEE 14-bus System

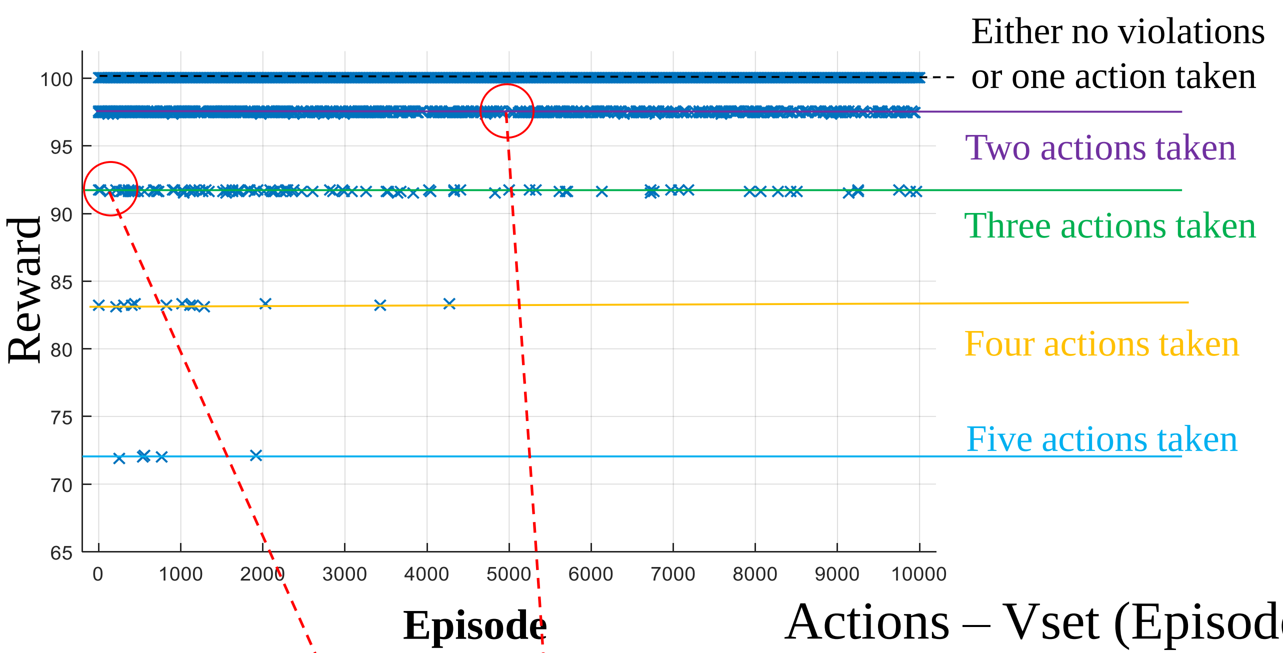
60%-120% random load changes are applied to each episode



After 10,000 episodes' learning, the designed DQN agent starts to master the voltage control problem by making decisions autonomously.

A Closer Look at the Results

GEIRINA
GEIRI NORTH AMERICA



60%-120% random system load changes

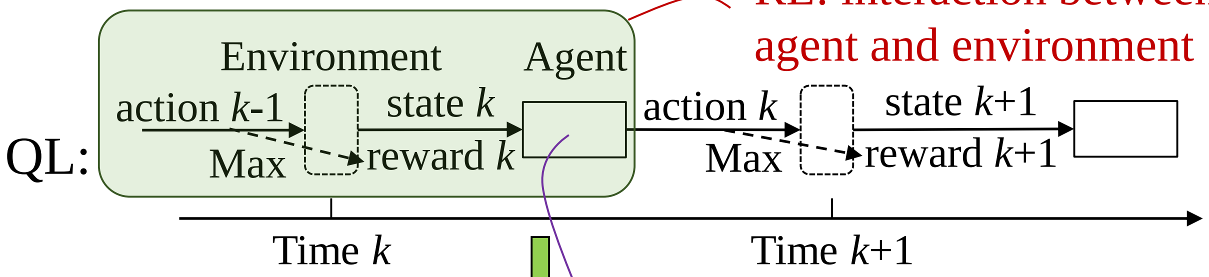
Actions – Vset (Episode 8 and 5000)

gen1_vset	gen2_vset	gen3_vset	gen6_vset	gen8_vset	episode
1.05	1.025	1	0.95	0.975	8
1.025	0.975	0.95	1	1.05	8
0.975	1	0.95	1.025	1.05	5000

States – Bus Voltage (Episode 8 and 5000)

bus1	bus2	bus3	bus4	bus5	bus6	bus7	bus8	bus9	bus10	bus11	bus12	bus13	bus14	episode
1.06	1.045	1.01	1.01797	1.02025	1.07	1.06204	1.09	1.05682	1.05137	1.05756	1.05568	1.05237	1.03698	8
1.05	1.025	1	0.97375	0.9756	0.95	0.974	0.975	0.96352	0.95255	0.94802	0.93591	0.9342	0.93076	8
1.025	0.975	0.95	0.95572	0.95909	1	1.00554	1.05	0.99402	0.98678	0.99011	0.98523	0.98225	0.96972	8
1.06	1.045	1.01	1.01699	1.01936	1.07	1.06047	1.09	1.05409	1.04913	1.05583	1.05456	1.05036	1.03339	5000
0.975	1	0.95	0.9627	0.96341	1.025	1.01158	1.05	1.00331	0.99898	1.00803	1.00845	1.00369	0.98341	5000

Discrete vs Continuous Action Space



RL: interaction between agent and environment

DL (DNN): policy and Q-function updating

DQN: using multiple layers of nonlinear process units (neural network) for feature extraction and transformation; Using value function to select action (e.g., ε-greedy)

DDPG: using one deep neural network for actor and another one for critic. The action is directly generated by actor based on the value from critic.

Discrete state (Q-Table) & **discrete action** (a=argmax$Q(s|\theta^{v})$)

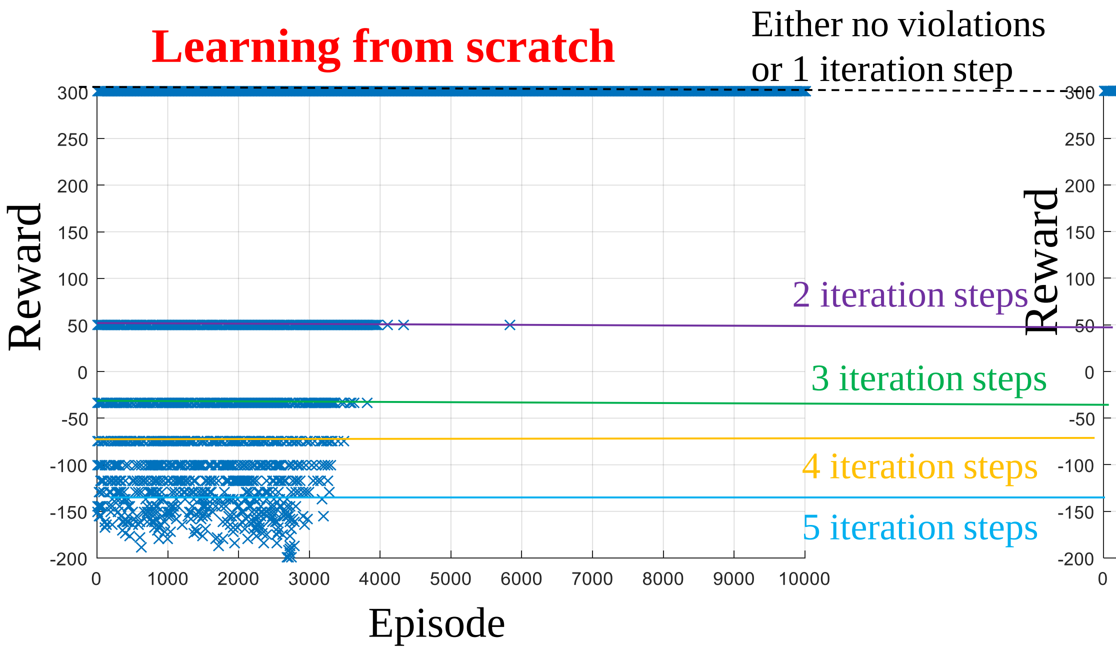
Continuous state (Q-Network) & **discrete action** a=argmax$Q(s|\theta^{v})$

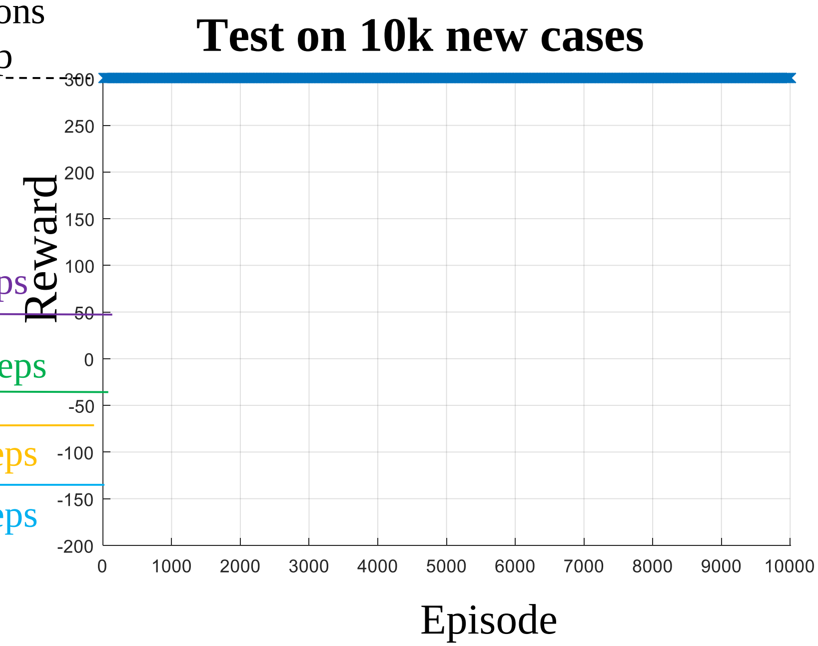
Continuous state (Q-Network) & **continuous action** $a=\mu(s|\theta^{\mu})$

DDPG Agent for IEEE 14-bus System

GEIRINA
GEIRI NORTH AMERICA

60%-120% random load changes are applied to each episode





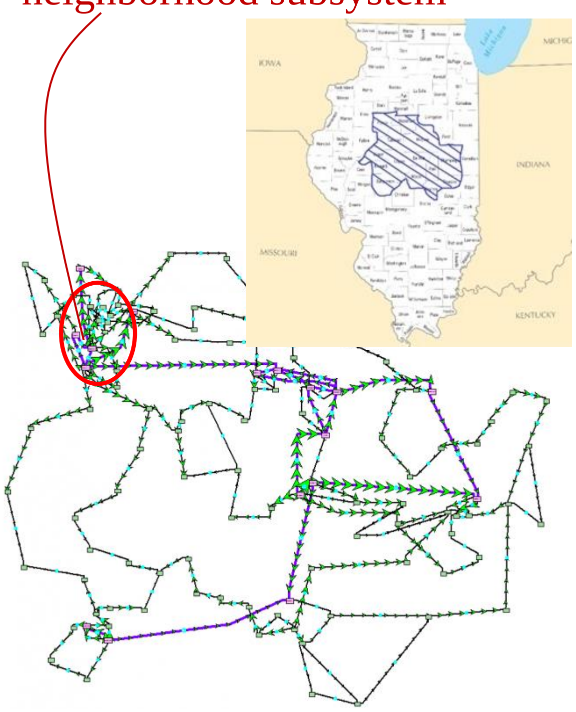
After 6,000 episodes' learning, the designed DDPG agent starts to master the voltage control problem by making decisions autonomously.

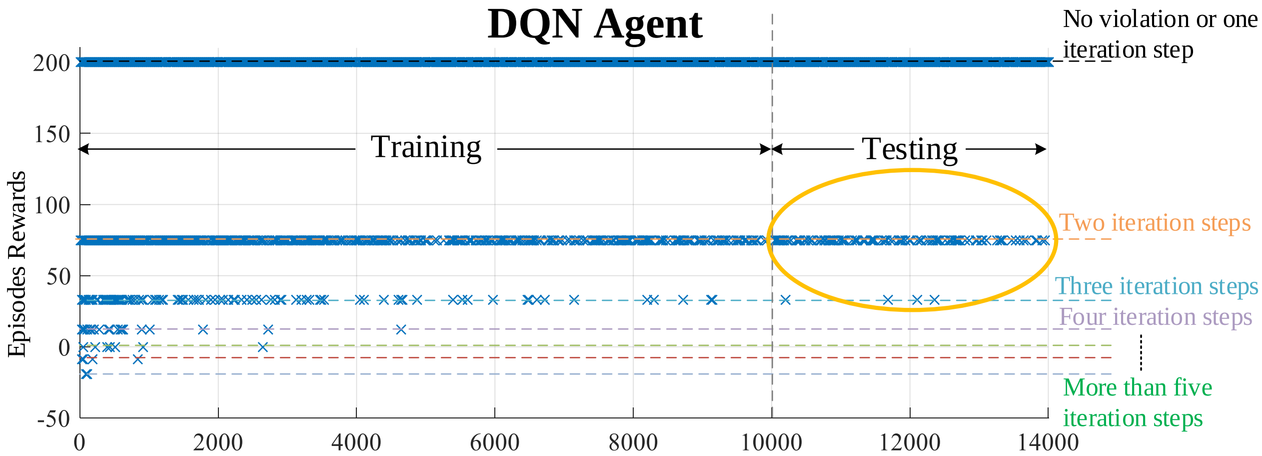
GEIRINA
GEIRI NORTH AMERICA

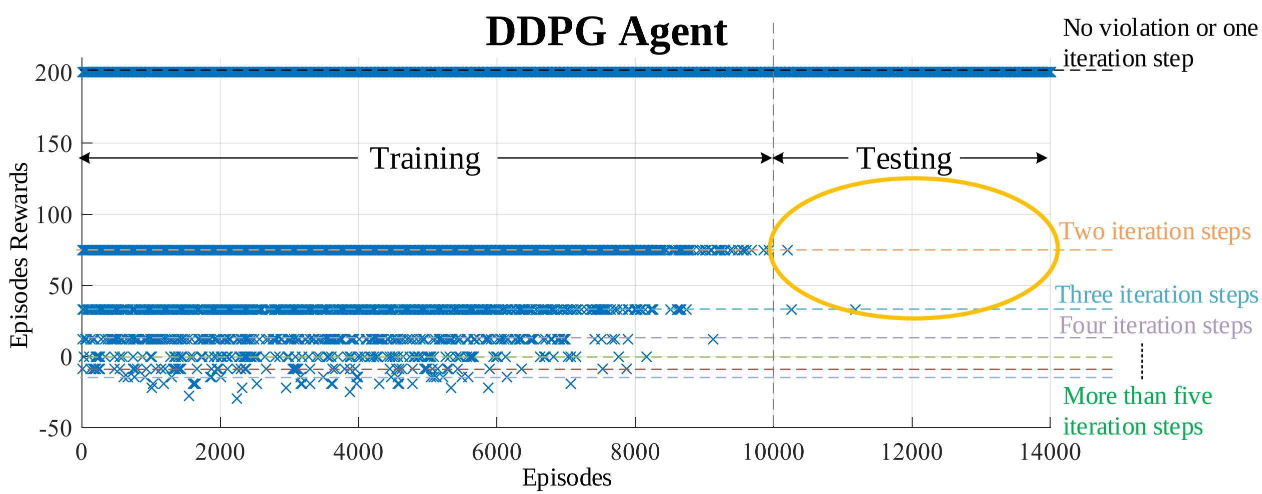
DQN and DDPG Agents for Illinois 200-bus System

60%-120% random load changes are applied to each episode

Regional voltage control is considered for DQN agent:
5 adjacent generators with 30 interconnected buses in the neighborhood subsystem





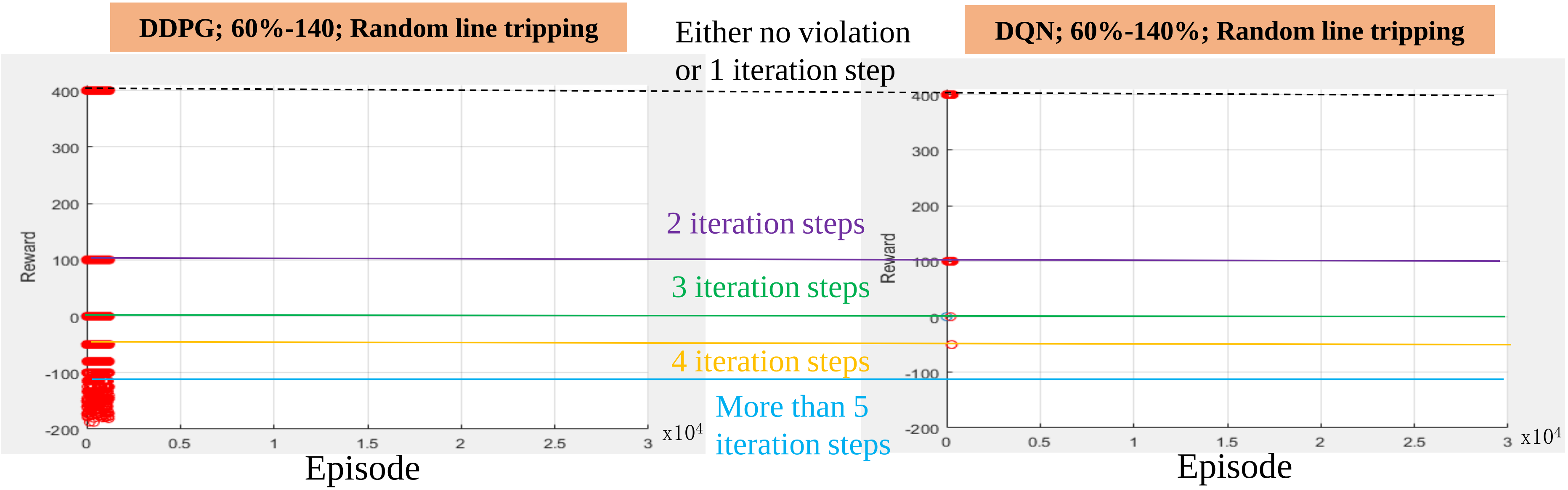


After 10,000 episodes' learning, the designed DRL agents start to master the voltage control problem in the 200-bus system by making decisions autonomously.

Further Testing Results-200 Bus System

GEIRINA
GEIRI NORTH AMERICA

- Test the DRL agent under different loading conditions: heavily loaded, fully loaded, and lightly loaded.
- **Consider different topological changes. For example, random line tripping contingency or N-1 conditions.**



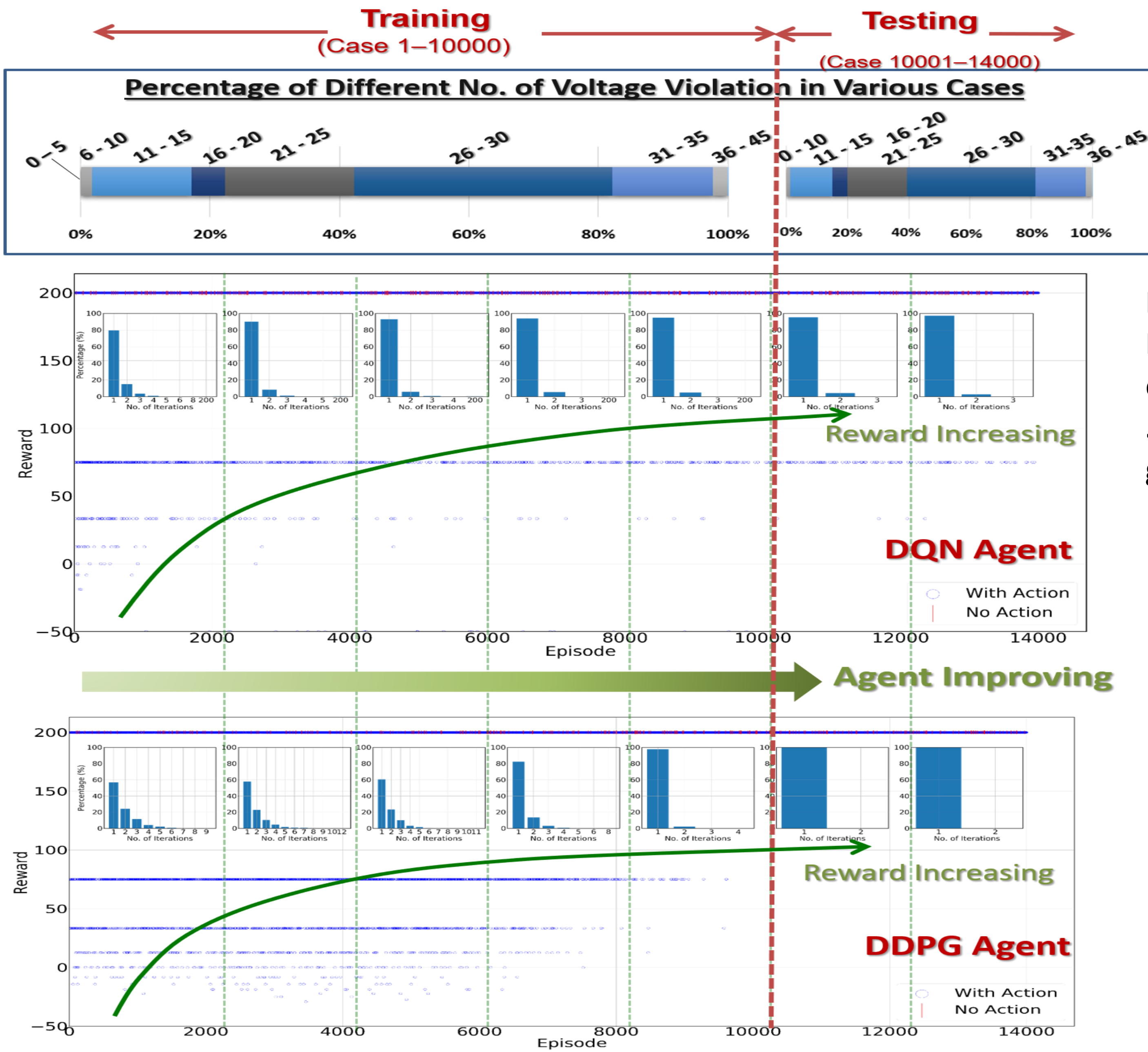
Observations:

1. The designed agents work very well under all testing conditions.
2. The results comply with basic power system principles and engineering judgement very well.
3. The proposed framework is promising for power system autonomous operation and control.

Summary of Results: IEEE 14-bus System

GEIRINA
GEIRI NORTH AMERICA

Summary of Results: Illinois 200-bus System

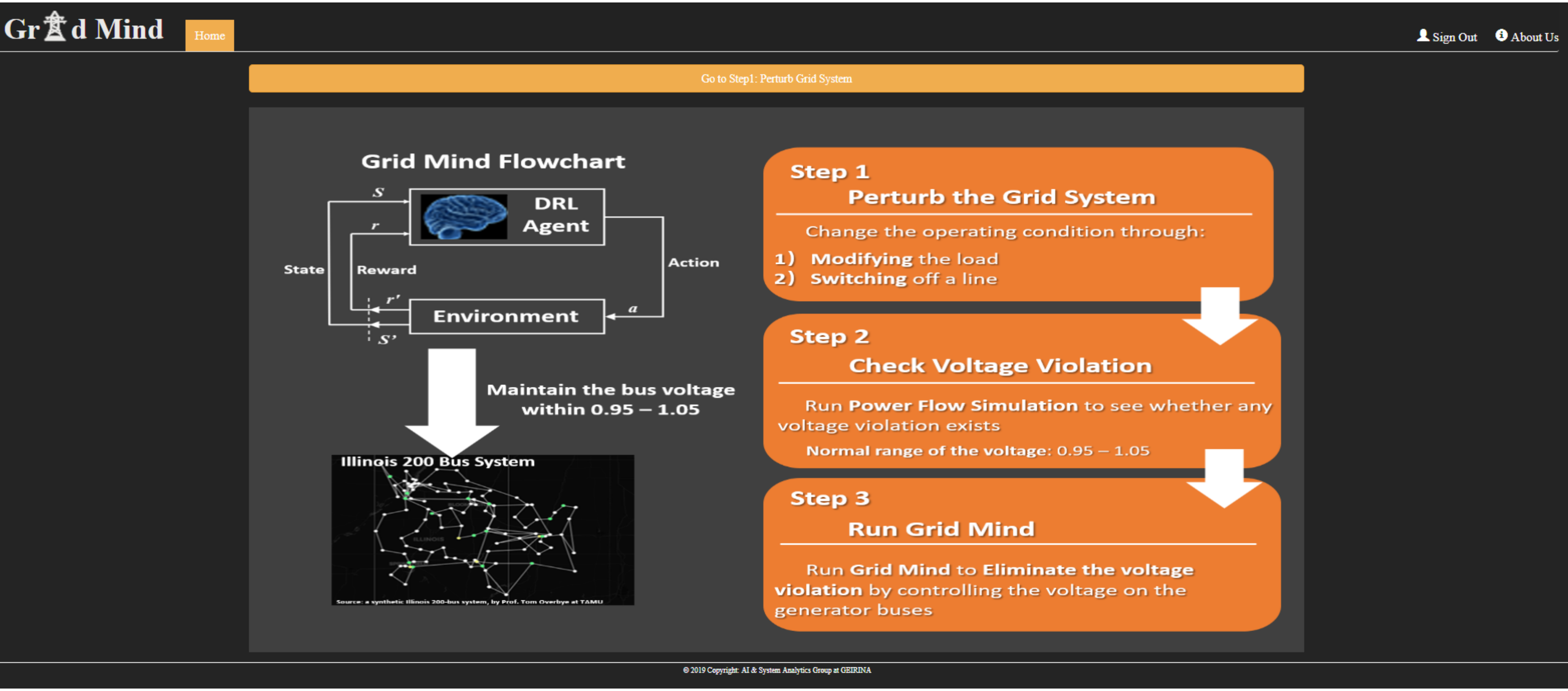


Note that the DQN Agent only controls 5 adjacent generators

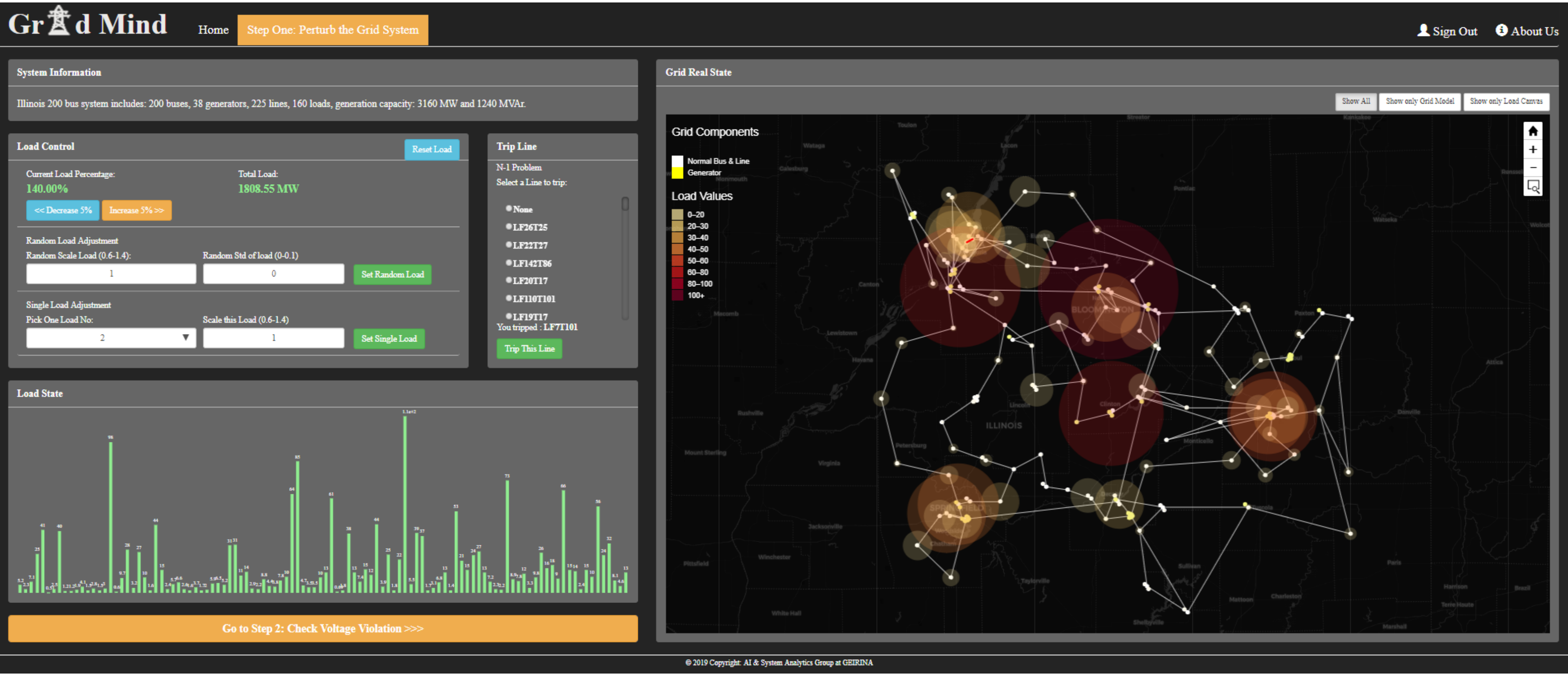
Outline

- Background and motivation
- Grid Mind: Autonomous grid dispatch and control based on PMU measurements
  - Deep Reinforcement Learning
  - Autonomous grid voltage control
  - Demo
  - AC OPF?
- How to architect/tune an effective self-learning agent?
- Discussion/Other applications

Demo of Grid Mind: Autonomous Voltage Control

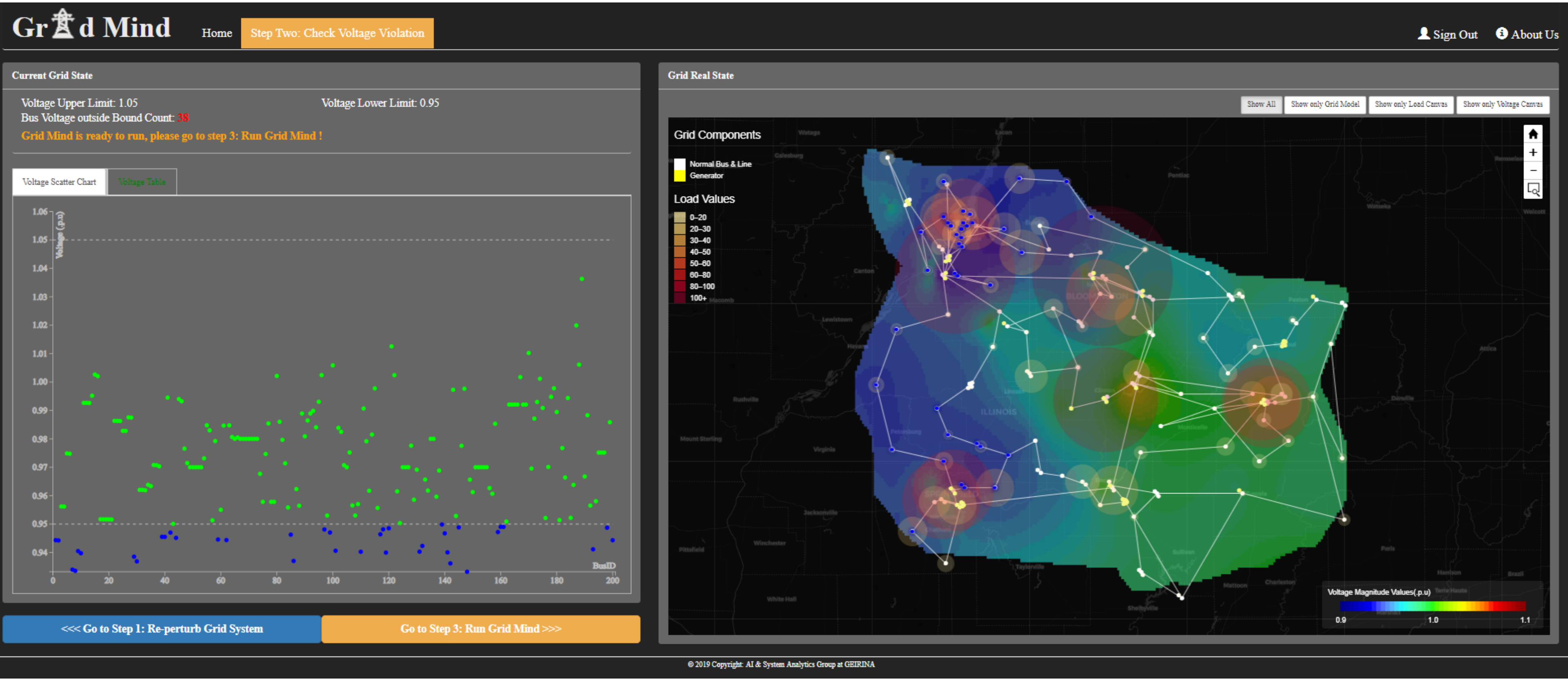


Step 1: Perturb the System

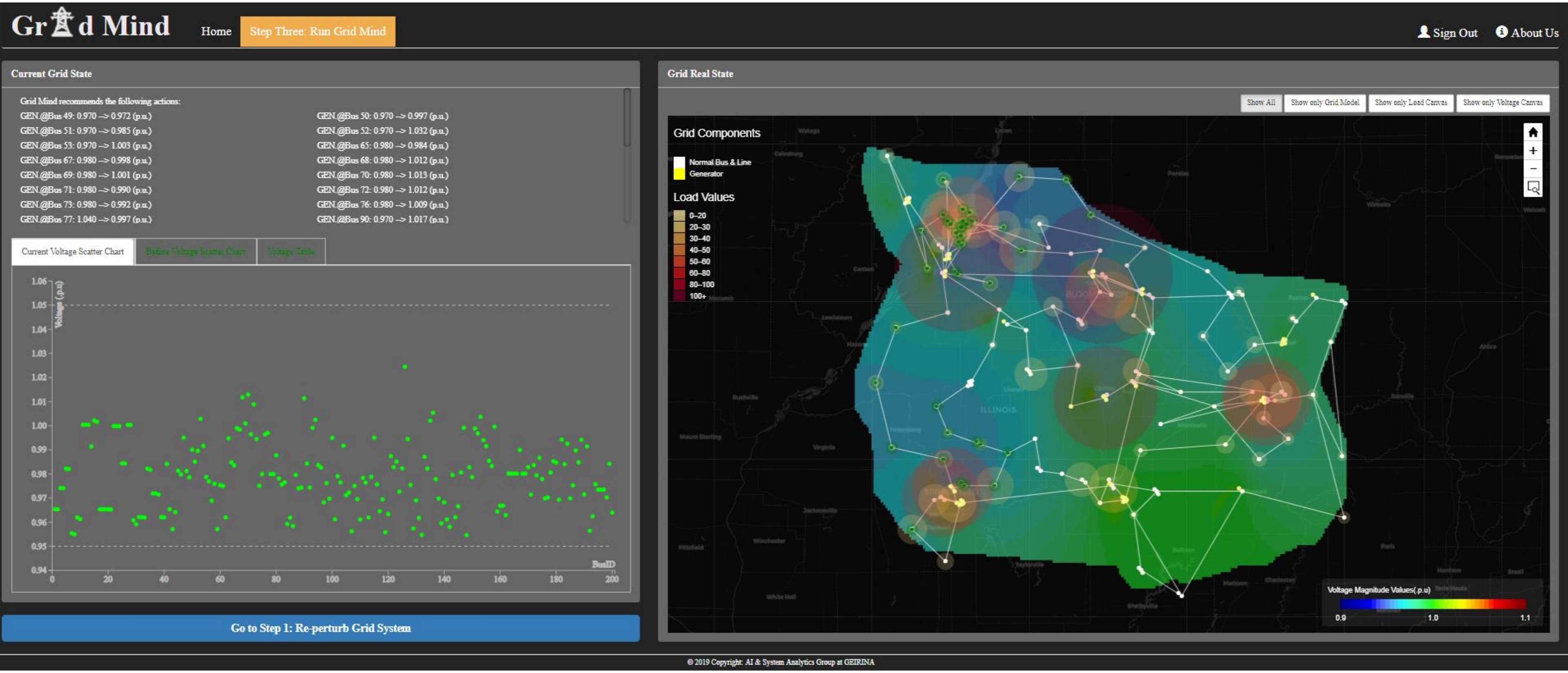


GEIRINA
GEIRI NORTH AMERICA

Step 2: Check for Voltage Violations



Step 3: Grid Mind Suggests Actions and Performance



DRL for AC OPF

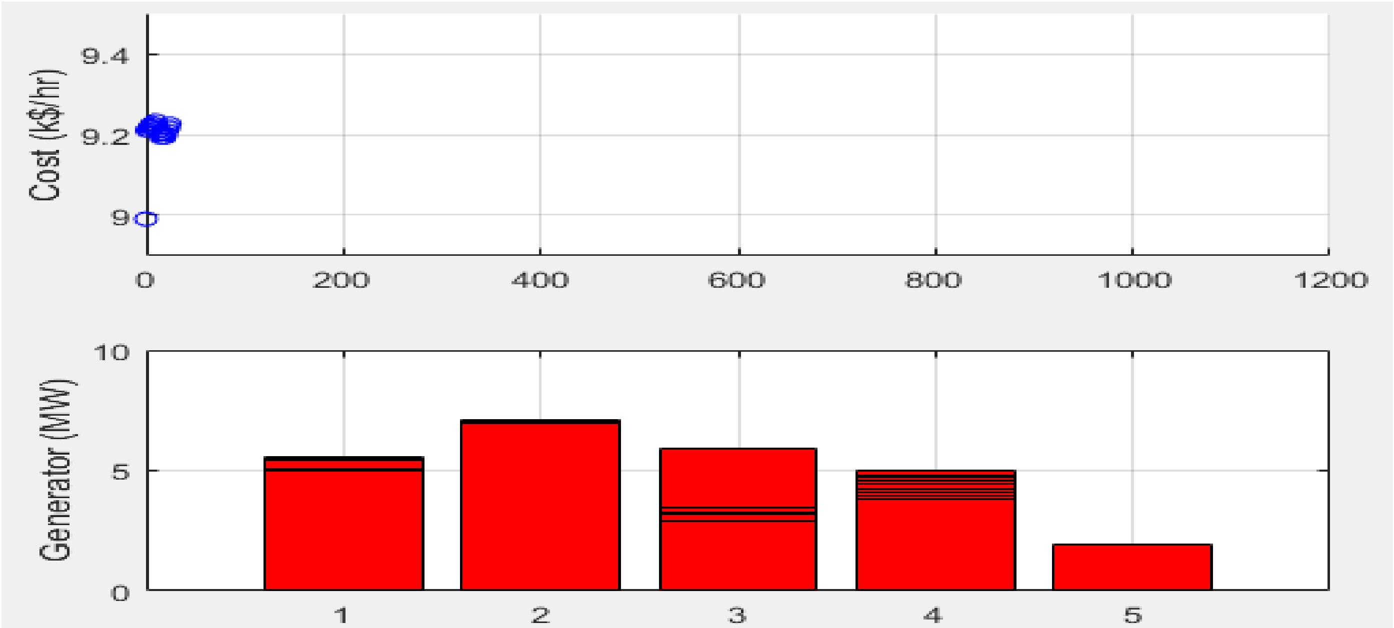
Constraints: (a) active power outputs of generators - defined in action space, (b) reactive power of generators - fulfilled by PF solver, (c) bus voltages - learned by agent

Optimal Objective: $$\text{cost} = \sum_{i=1}^{n} a_i P_i^2 + b_i P_i \quad (1)$$

Each generator has three actions, i.e. [+0.5, 0, -0.5 MW]

Define Reward as,

$$\text{Reward} = \begin{cases} 300 - 0.02 * \text{cost}, \forall \text{ voltage bounded} \\ -500, \forall \text{ voltage violated} \end{cases} \quad (2)$$



DRL Based Training/Solution Process for ACOPF

DRL for AC OPF-Numerical Results

GEIRINA
GEIRI NORTH AMERICA

Comparison with classical Matlab-coded solutions

MATPOWER Optimal Solution			
Load	Pgen（MW）	Cost（$/hr）	Power Mismatch (MW)
120%	[195.39, 48.10, 26.28, 11.40, 9.71]	8994.35	1e-09
110%	[192.55, 48.10, 14.93, 11.40, 0.31]	8054.64	1e-09
100%	[176.94, 48.15, 5.80, 11.46, 0.32]	7131.94	1e-09
90%	[151.77, 48.10, 5.80, 11.40, 0.30]	6267.91	1e-09
80%	[126.73, 48.10, 5.8, 11.40, 0.30]	5467.28	1e-09
DRL Agent Optimal Solution			
Load	Pgen（MW）	Cost（$/hr）	Power Mismatch (1e-2 MW)
120%	[215.4, 48.2, 8.3, 14.0, 6.3]	8994.93	4.9619
110%	[201.7, 48.2, 5.8, 11.5, 0.3]	8033.83	4.9089
100%	[176.8, 48.2, 5.8, 11.5, 0.3]	7130.57	4.9619
90%	[151.6, 48.2, 5.8, 11.5, 0.3]	6270.71	-0.2949
80%	[125.0, 49.2, 5.8, 11.5, 0.3]	5466.69	3.3988

Outline

- Background and motivation
- Grid Mind: Autonomous grid dispatch and control based on PMU measurements
  - Deep Reinforcement Learning
  - Autonomous voltage control
  - Demo
  - AC OPF?
- How to architect/tune an effective self-learning agent?
- Discussion/Other applications

How to Design/Train an Effective DRL Agent

GEIRINA
GEIRI NORTH AMERICA

➢ Considerations

There are tons of parameters, settings, and different formulations that need to be designed and specified. And subtle difference in them may generate very different results.

➢ Testing Roadmap

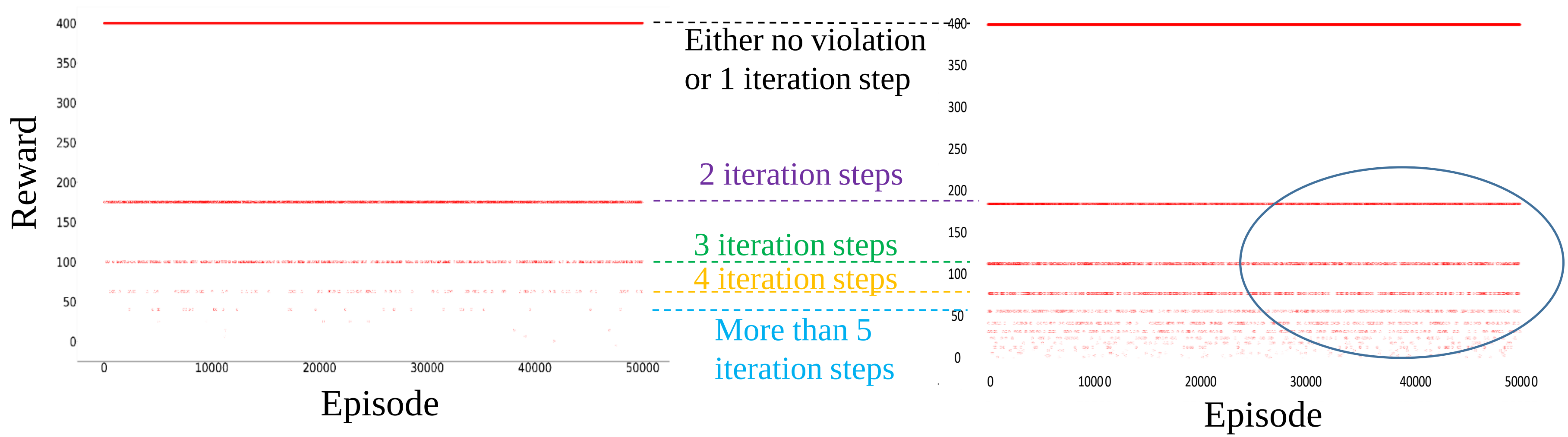
1. Consider different sizes of action space
2. Consider different neural network structures
   - Number of neural networks
   - Number of layers
   - Number of neurons
3. Consider different regularization methods
   - Batch normalization
   - Layer dropout
4. Consider different DRL formulations
   - Deep-Q-Network (DQN)
   - Deep-Deterministic-Policy-Gradient (DDPG)
5. Consider dynamic adjustment process

Case Study 1: Different Sizes of Action Space

Objective: Evaluate influence of different sizes of action space for DQN-based framework.

4^5 predefined action space

5^5 predefined action space



Observation:

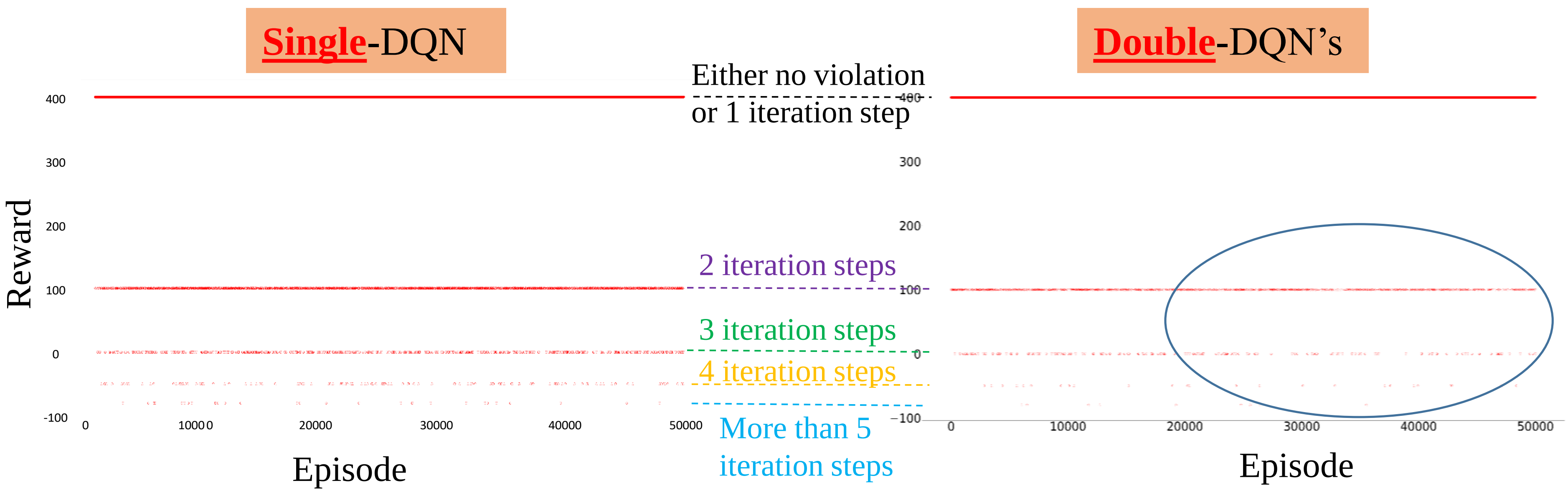
With other settings/parameters the same,

1. Performance of a DQN-based agent deteriorates when the size of action space grows.
2. Reducing the action space (4^5) can significantly improve the performance.

Case Study 2: Different NN Structure-No. of NN

Objective: evaluate the influence of two different types of DQN structure.

- **Single-DQN**: use only **one** NN to estimate the *Q*-function
- **Double-DQN**: use **two** NNs - one NN updates periodically as a **target NN**, while the second NN updates more frequently as an **evaluation NN**.



Observation:

With other settings/parameters the same,

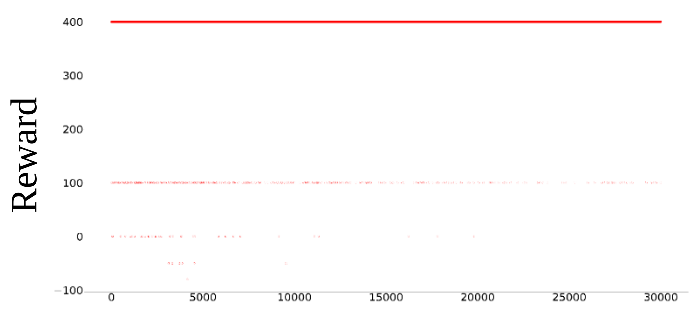
A double-DQN has a more stable training process and improves the overall performance.

GEIRINA
GEIRI NORTH AMERICA

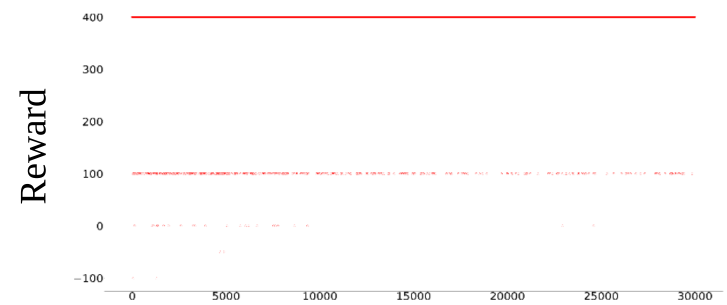
Case Study 3: Different NN Structure-Layer & No. of Neurons

Objective: evaluate the influence of different layers and neuron numbers for DQN agent.

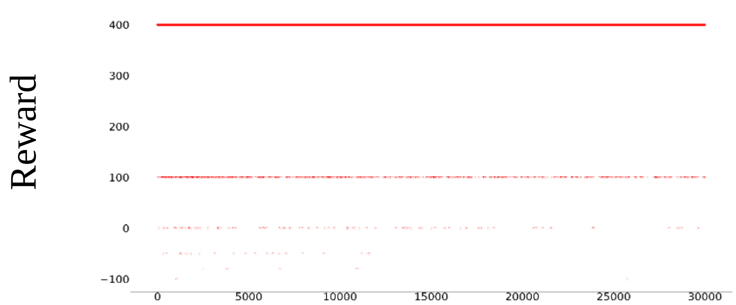
#layer=**2**, #neurons/layer=**20**



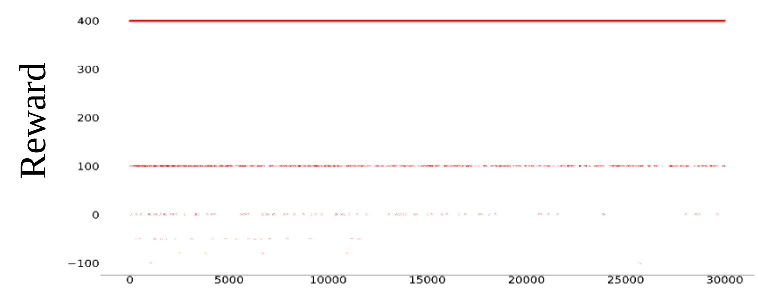
#layer=**3**, #neurons/layer=**20**



#layer=**2**, #neurons/layer=**40**



#layer=**3**, #neurons/layer=**40**



Observation:

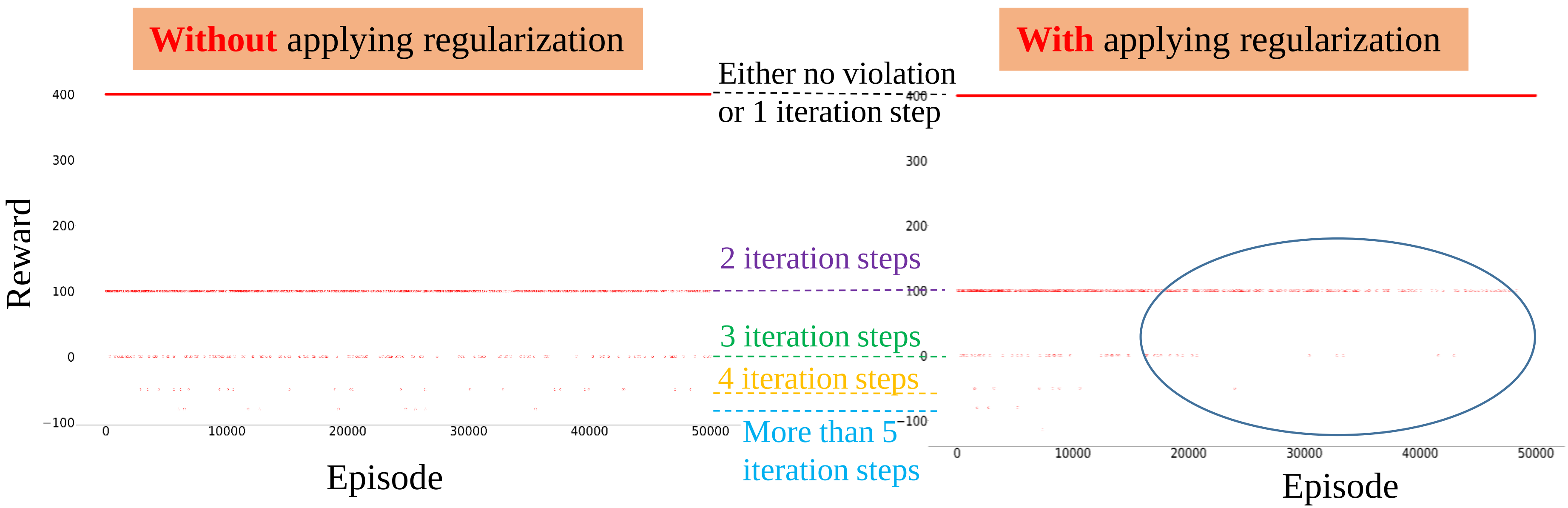
With other settings/parameters the same,

1. Changing number of layers or neurons does not make a significant influence.
2. Subtle performance degradation is observed when increasing layers or neurons.
3. It is partially because the testing system is quite simple and small.

Case Study 4: Influence of Regularization

Objective: evaluate the influence of regularization methods for DQN agent.

- **Feature & Batch Normalization**: normalize all variables to the same level of magnitude
- **Layer Dropout** (0.5 dropout rate): random dropout the output of certain neuron to make the NN not fully-connected



Observation:

With other settings/parameters the same,

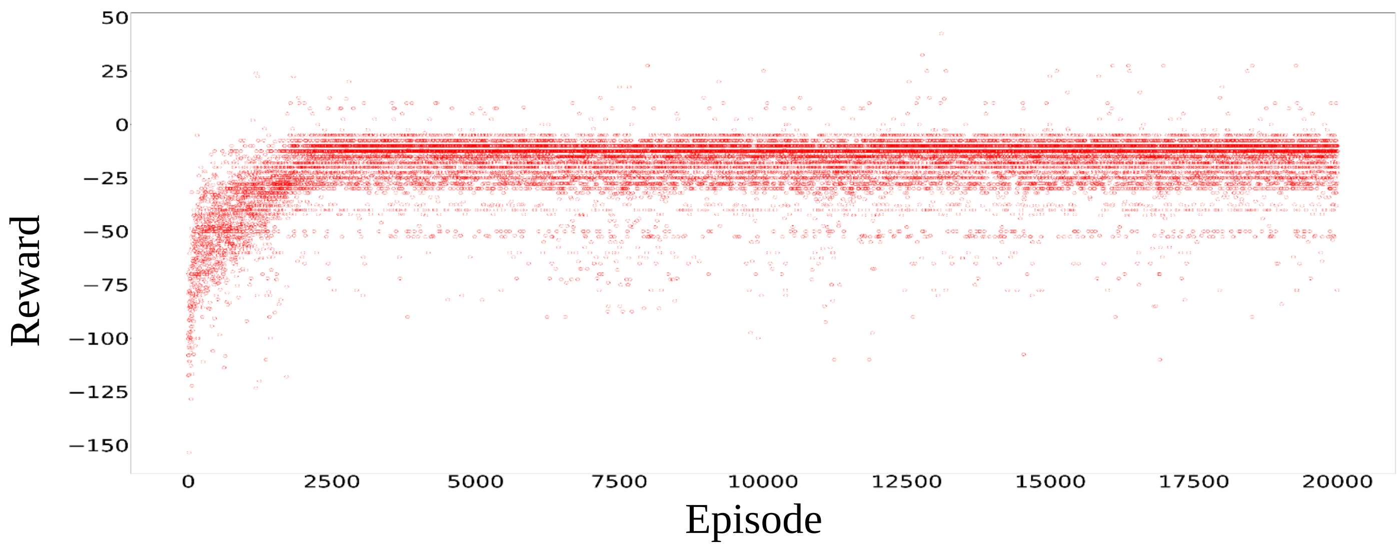
Applying regularization methods significantly improves performance of the agent.

GEIRINA
GEIRI NORTH AMERICA

Case Study 5: Dynamic Adjustment

Objective and background: DDPG may have over-estimation problem. In order to avoid this problem, the control actions is dynamically adjusted, so that the agent can better learn from more dynamic historical information.

1. **Dynamically** increase or decrease generator voltage within a bound of ±0.03
2. Rewards are reformulated, i.e., the more bus voltages closer to 1 pu, the higher the positive reward; the more violation buses, the lower the negative reward.



Observation:

With other settings/parameters the same,

1. The over-estimation problem is solved and the learning rate is no longer limited.
2. The average reward increases along the training process.

Lessons Learned, After Hundreds of Thousands of Numerical Experiments

GEIRINA
GEIRI NORTH AMERICA

Summary of Tuning Results

DQN Agent		
Objective	Measures	Conclusion
Evaluate influence of different sizes of action space	Change action space from 5^4 to 5^5	Performance deteriorates when action space size grows
Evaluate the influence of two different types of DQN structure	Single-DQN and double-DQN are tested	A double-DQN has a better performance over a signle-DQN
Evaluate the influence of layers and neuron numbers	Test with 2/3 layers with 20/40 neurons	Subtle performance degradation is observed when increasing lay. or neu.
Evaluate the influence of regularization methods	Using batch normalization and layer dropout	Applying regularization methods significantly improves performance
DDPG Agent		
Objective	Measures	Conclusion
Evaluate the influence of regularization methods	Using batch normalization and layer dropout	Applying regularization methods significantly improves performance
Evaluate a different formulation way to control voltage	Dynamically increase or decrease the voltage setting point for a small step in each iteration	The agent is able to solve the voltage problem using minimum iterations after well trained.

Conclusion and Future work

- The proposed DRL framework demonstrates very promising results for power system autonomous dispatch and control, using measurements from advanced sensors, PMU as an example.
  - When reactive resources are sufficient and/or distributed unevenly, Grid Mind can find very fast and effective solutions for fixing voltage issues.
  - Research team will train and enhance AI agents to find optimal solutions for scenarios with limited reactive resources.
- Thorough testing has been carried out to study the influence of various factors, which sheds light on the design of an effective agent/robot.
- Therefore, we have duplicated an example of Alpha Zero, Grid Mind, for power systems.
- With extensive offline calculation and online learning, in the mid-term, Grid Mind serves as an assistant to grid operators; in the long term, Grid Mind will be the core of power system operation ROBOT.
- With proper modifications, the proposed framework can be applied to many other applications.

Related Publications

- R. Diao, Z. Wang, D. Shi, Q. Chang, J. Duan, and X. Zhang, “Autonomous Voltage Control for Grid Operation Using Deep Reinforcement Learning,” IEEE PES General Meeting, Atlanta, GA, USA, 2019.
- J. Duan, Z. Yi, D. Shi, and Z. Wang, "Reinforcement-Learning-Based Optimal Control for Hybrid Energy Storage Systems in Hybrid AC/DC Microgrids", IEEE Transactions on Industrial Informatics, 2019.
- J. Duan, D. Shi, R. Diao, B. Zhang, Z. Wang, etc., “Deep-Reinforcement-Learning-Based Autonomous Control for Power Grid Operations,” IEEE PES Letters, under 2nd-round review.
- D. Bian, Z. Yu, D. Shi, R. Diao, Z. Wang, "A Real-time Robust Low-Frequency Oscillation Detection and Analysis (LFODA) System with Innovative Ensemble Filtering," CSEE Journal of Power and Energy Systems, 2019.
- L. Mang, D. Shi, Z. Yu, Z. Yi, Z. Wang, and Y. Xiang, “An ADMM Based Approach for Phasor Measurement Unit Data Recovery,” IEEE Transations on Smart Grid, 2018.
- H. Banna, Z. Yu, D. Shi, Z. Wang, D. Su, C. Xu, S. Solanki, and J. Solanki, “Online Coherence Identification Using Dynamic Time Warping for Controlled Islanding,” Journal of Modern Power System and Clean Energy, vol. 7, no. 1, pp. 38-54, Jan. 2019.
- Z. Yu, D. Shi, Z. Wang, Q. Zhang, J. Huang, and S. Pan, “Distributed Estimation of Oscillations in Power Systems: an Extended Kalman Filtering Approach,” CSEE Journal of Power and Energy Systems, 2018.
- X. Lu, D. Shi, B. Zhu, Z. Wang, J. Luo, D. Su, and C. Xu, “PMU Assisted Power System Parameter Calibration at Jiangsu Electric Power Company,” IEEE PES General Meeting, Chicago, IL, USA, 2017. [Best Paper]
- F. Hu, K. Sun, D. Shi, and Z. Wang, “Measurement-based Voltage Stability Assessment for Load Areas Addressing n-1 Contingencies,” IET Generation, Transmission & Distribution, vol. 11, no. 15, pp. 3731-3738, 2017.

Thank you!

di.shi@geirina.net

www.geirina.net/research/2