ANDRO

Leading Edge Communications

Offline Reinforcement Learning and Cognitive Radio Resource Management for Space-based Radio Access Network Optimization

IEEE CCAAW

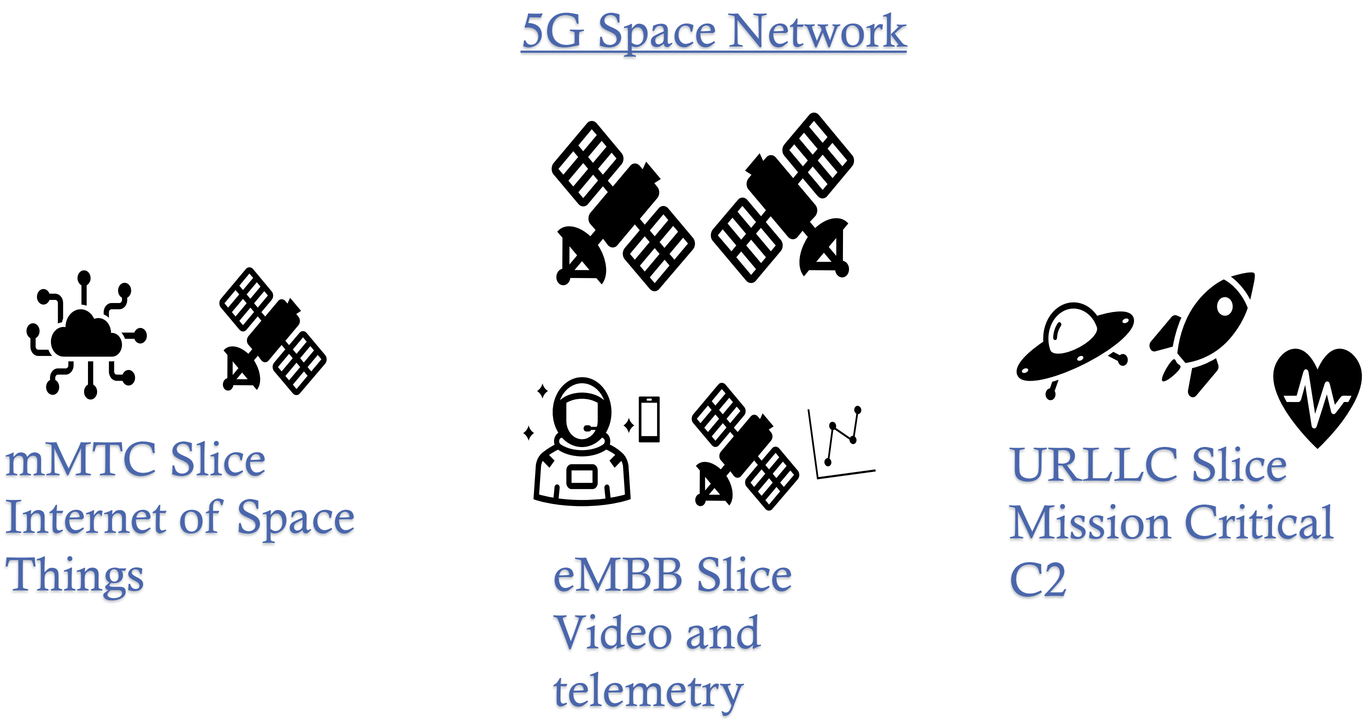
June 20th, 2023

Sean Furman

sfurman@androcs.com

MOTIVATION

- Current need to demonstrate desired network optimization for the highly diverse and dynamic quality of service (QoS) requirements of space-based networks and provide seamless inclusion of delay-tolerant networks.



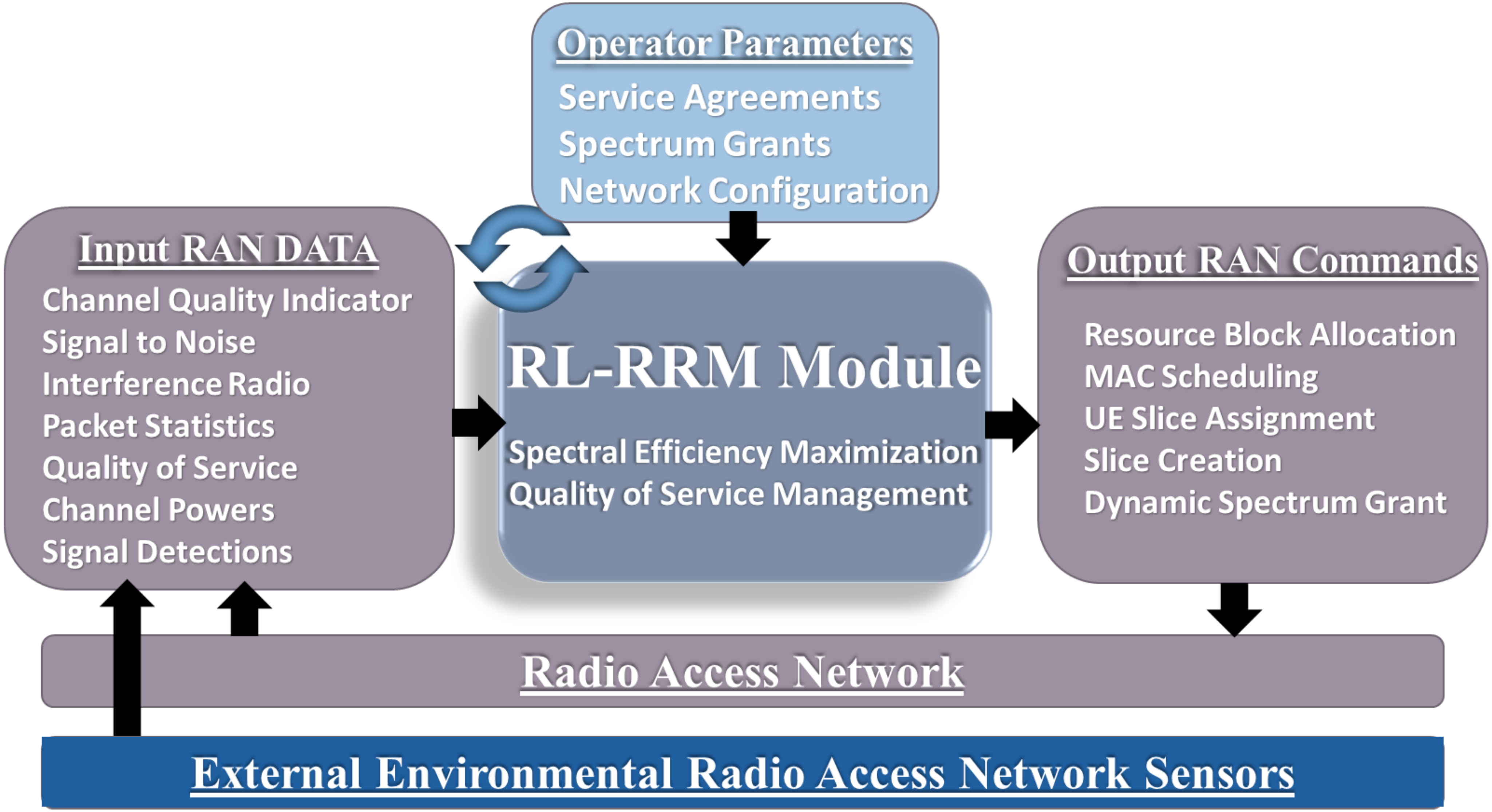
WHY 5G + OFFLINE REINFORCEMENT LEARNING

Higher Data Rates • Lower Latencies • Increased Reliability • Flexible network management

But lacks efficient and optimized radio resource management (RRM) strategies

Exploiting offline RL for optimized RRM for challenging and dynamic operating requirements

SYSTEM MODEL



WHAT LEARNING PARADIGM FOR SPACE BASED NETWORKS?

Supervised learning?

Do not have labeled data for optimal actions

Reinforcement learning?

Good for learning optimal control policies

Active and online process

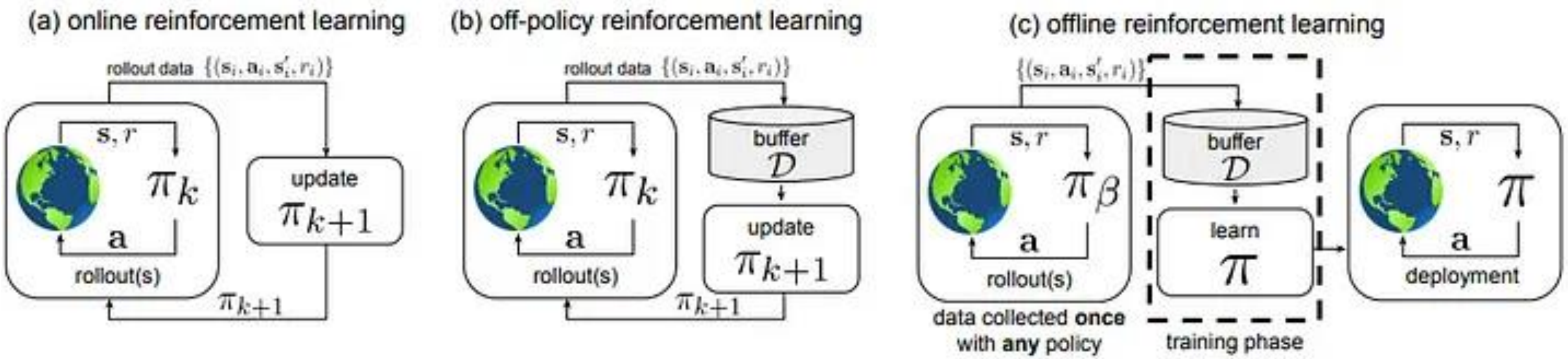
Requires many iterations to converge

Need to re-collect data each time algorithm is trained

Offline reinforcement learning?

Train on large previously collected datasets from arbitrary policies to learn a better policy

Potential to achieve high performance and generalization capacity



[4] S. Levine, A. Kumar, G. Tucker and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," arXiv preprint, vol. arXiv:2005.01643v3, 2020.

OFFLINE RL – BENEFITS AND CHALLENGES

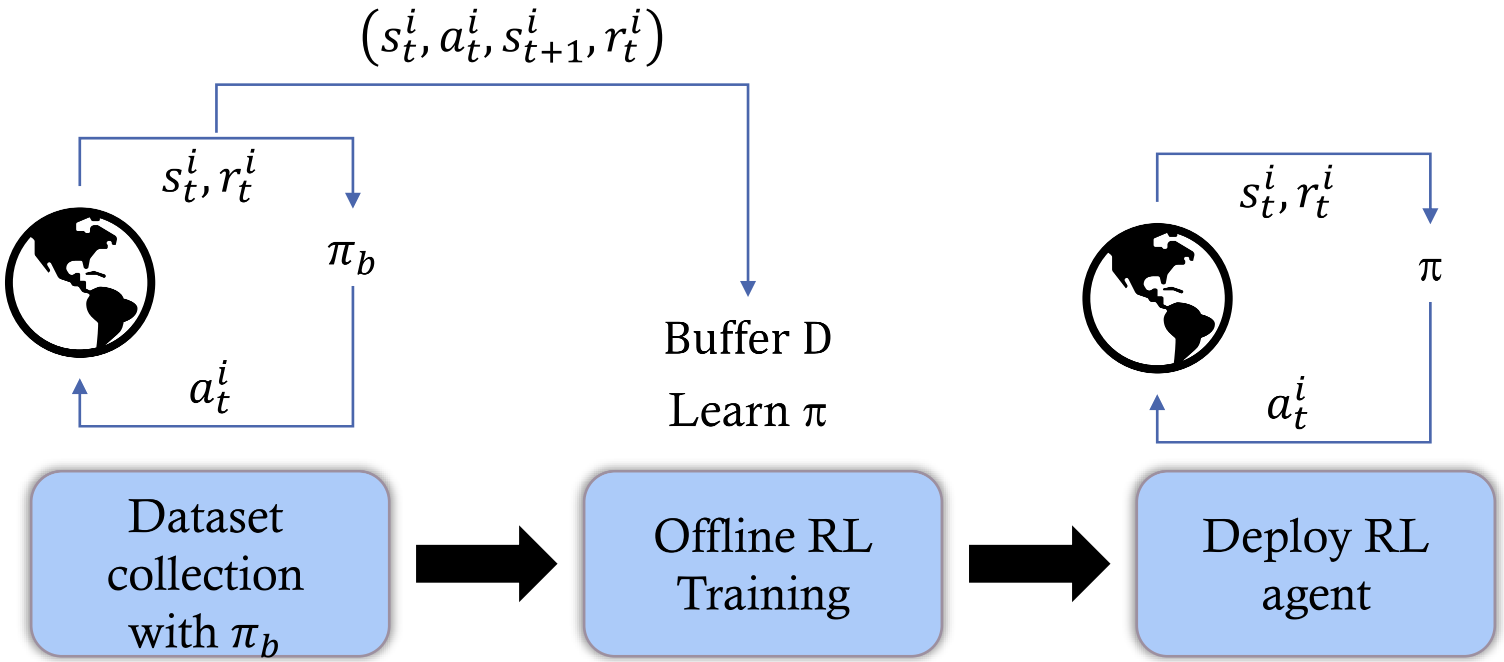
- Benefits
  - Large datasets -> better generalization and performance
  - Re-use previous datasets
- Challenges
  - Static dataset
  - Distribution shift
- Solutions
  - Policy Constraints
  - Conservative Algorithms
  - Uncertainty Estimation

OFFLINE RL FOR NETWORK SLICING

Dataset Collection
- Deploy arbitrary policies onto 5G network and record transition data
- This work uses transitions from training online RL algorithms

Offline RL Training
- Train offline RL agent using dataset D

Deploy RL Agent
- Use trained RL agent for 5G network slicing



RRM FORMULATION

- Dynamic wireless bandwidth allocation in 5G downlink for single base station [5]
  - N - no. of network slices
  - W - total bandwidth
  - $\boldsymbol{d} = (d_1, \ldots, d_N)$ – current demands of each slice
  - System utility - $\alpha SE + \sum_{n \in N} \beta_n SSR_n$
  - Find $\boldsymbol{w} = (w_1, \ldots, w_N)$ that maximizes system utility, $\max_{\mathbf{w}}(\alpha SE + \sum_{n \in N} \beta_n SSR_n)$
- RL formulation
  - Observation space – no. of arrived packets for each slice in time window
  - Action space – bandwidth allocation to each slice
  - Reward function – utility function

[5] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao and H. Zhang, "Deep Reinforcement Learning for Resource Management in Network Slicing," IEEE Access, pp. 74429-74441, 2018.

SIMULATION SETTINGS

- VoLTE, Video, and URLLC slices
- UEs within 100m radius of base station
- Each slice's network traffic is modeled with inter-arrival time and packet size distributions
- Each slice has service level agreements specified by data rate and latency
- Bandwidth allocation resolution: 1 MHz
- System utility settings: $\alpha = 0.001$ and $\beta = (1,1,2)$.
- Python simulation environment from [6-7]

Setting	VoLTE	Video	URLLC
Bandwidth	10 MHz		
Scheduling	Round robin per slot (0.5 ms)		
Slice band adjustment	1 second (2,000 scheduling slots)		
Channel	Rayleigh fading		
User No	46	46	8
Distribution of Inter-Arrival Time per user	Uniform [Min = 0, Max = 160 ms]	Truncated stationary distribution [Exponential Para = 1.2, Mean = 6 ms, Max = 12.5 ms]	Exponential [Mean = 180 ms]
Distribution of Packet Size	Constant (40 byte)	Truncated Pareto [Exponential Para = 1.2, Mean = 100 byte, Max = 250 byte]	Variable constant: {0.3, 0.4, 0.5, 0.6, 0.7} Mbyte
SLA: Rate	51 Kbps	100 Mbps	10 Mbps
SLA: Latency	10 ms	10 ms	3 ms

[6] Y. Hua, R. Li, Z. Zhao, X. Chen and H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," IEEE Journal on Selected Areas in Communications, vol. 38, no. 2, pp. 334-349, 2020.
[7] R. Li, C. Wang, Z. Zhao, R. Guo and H. Zhang, "The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility," IEEE Communications Letters, vol. 24, no. 9, pp. 2005-2009, 2020.

ALGORITHMS EVALUATED

- Offline RL algorithms
  - Conservative Q Learning (CQL) [10]
  - Batch Constrained Deep Q Learning (BCQ) [11]
  - Trained on transitions from training online RL algorithms
  - d3rlpy library [12]
- Online RL algorithms
  - DQN
  - GAN-DDQN [6]
  - LSTM A2C [7]
- Hard Slicing – equal bandwidth allocated to each slice
- No Slicing – round robin scheduling across all slices

[6] Y. Hua, R. Li, Z. Zhao, X. Chen and H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing”
[7] R. Li, C. Wang, Z. Zhao, R. Guo and H. Zhang, "The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility"
[10] A. Kumar, A. Zhou, G. Tucker and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning,"
[11] S. Fujimoto, D. Meger and D. Precup, "Off-Policy Deep Reinforcement Learning without Exploration”
[12] T. Seno and M. Imai, "d3rlpy: An Offline Deep Reinforcement Library"

RESULTS

Scenario: Demand-aware Resource Management

Performance Metric	System Utility	Spectrum Efficiency	SSR of VoLTE Service	SSR of eMBB service	SSR of URLLC service	Bandwidth (MHz) of VoLTE	Bandwidth (MHz) of eMBB	Bandwidth (MHz) of URLLC
CQL	4.05	164.74	1.0	0.99	0.95	1	3	6
BCQ	4.05	164.74	1.0	0.99	0.95	1	3	6
DQN	4.05	164.74	1.0	0.99	0.95	1	3	6
Dueling GAN-DDQN	4.06	164.51	1.0	0.99	0.95	1	3	6
Hard Slicing	3.06	204.74	1.0	0.99	0.43	3.33	3.33	3.33
No Slicing	2.91	443.70	1.0	1.0	0.23	0.70	8.46	0.40

Findings

- Offline RL algorithms achieved similar performance as online RL algorithms
- Each RL algorithm converged to constant BW solution
- RL algorithms achieve SSRs of at least 95% for each slice
- Hard slicing’s 3.33 MHz bandwidth is insufficient for URLLC
- No Slicing’s global round robin scheduling does not account for larger size and lower rate of URLLC packets. RR updates every scheduling interval, resulting in highest SE
- RL algorithms trade off lower SE to meet QoS requirements due to time and frequency resolution of bandwidth allocation

Need for RL algorithms to allocate bandwidth at finer resolution.

CONCLUSIONS AND FUTURE WORK

- Motivated 5G and network slicing for improving space-based networks
- Evaluated offline RL algorithms for optimization of 5G radio resource management
- Future work
  - Improve realism in simulator
    - Dynamic scenarios
    - Increase frequency-time resolution of resource allocation
    - Delays and disruptions of space networks
  - Flexible resource allocation with RL
  - Offline RL with online RL for fine-tuning
  - Training dataset comparisons

ACKNOWLEDGEMENTS

- ANDRO Computational Solutions
- Co Authors
  - Timothy Woods twoods@androcs.com
  - Chris Maracchion cmaracchion@androcs.com
  - Andy Drozd adrozd@androcs.com
- IEEE CCAAW
- Audience

REFERENCES

[1] I. Leyva-Mayorga et al., "LEO Small-Satellite Constellations for 5G and Beyond-5G Communications," in IEEE Access, vol. 8, pp. 184955-184964, 2020, doi: 10.1109/ACCESS.2020.3029620.

[2] O. Somerlock, A. Sharma and G. W. Heckler, "Adapting Commercial 5G Terrestrial Networks for Space," 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-7, doi: 10.1109/AERO53065.2022.9843534.

[3] J. A. H. Sanchez, K. Casilimas and O. M. C. Rendon, "Deep Reinforcement Learning for Resource Management on Network Slicing: A Survey," sensors, vol. 22, no. 8, 2022.

[4] S. Levine, A. Kumar, G. Tucker and J. Fu, "Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems," arXiv preprint, vol. arXiv:2005.01643v3, 2020.

[5] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao and H. Zhang, "Deep Reinforcement Learning for Resource Management in Network Slicing," IEEE Access, pp. 74429-74441, 2018.

[6] Y. Hua, R. Li, Z. Zhao, X. Chen and H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," IEEE Journal on Selected Areas in Communications, vol. 38, no. 2, pp. 334-349, 2020.

[7] R. Li, C. Wang, Z. Zhao, R. Guo and H. Zhang, "The LSTM-Based Advantage Actor-Critic Learning for Resource Management in Network Slicing With User Mobility," IEEE Communications Letters, vol. 24, no. 9, pp. 2005-2009, 2020.

[8] Y. Liu, J. Ding, Z.-L. Zhang and X. Liu, "CLARA: A Constrained Reinforcement Learning Based Resource Allocation Framework for Network Slicing," in IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021.

[9] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno and H. Mineno, "Flexible Resource Block Allocation to Multiple Slices for Radio Access Network Slicing Using Deep Reinforcement Learning," IEEE Access, vol. 8, pp. 68183-68198, 2020.

[10] A. Kumar, A. Zhou, G. Tucker and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," in Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020.

[11] S. Fujimoto, D. Meger and D. Precup, "Off-Policy Deep Reinforcement Learning without Exploration," in International Conference on Machine Learning, 2019.

[12] T. Seno and M. Imai, "d3rlpy: An Offline Deep Reinforcement Library," in Offline Reinforcement Learning Workshop at Neural Information Processing Systems, 2021.

BACKUP

RESULTS

Scenario: Dynamic Environment

Differences

- Mobile UEs.
- System utility settings: $\alpha = 0.01$ and $\beta = (1,1,1)$.
- URLLC latency 1 ms
- URLLC packet size 0.3 Mbyte
- Reward shaping

	System Utility/Reward	Spectrum Efficiency	SSR of VoLTE Service	SSR of eMBB service	SSR of URLLC service	Bandwidth (MHz) of VoLTE	Bandwidth (MHz) of eMBB	Bandwidth (MHz) of URLLC
CQL	5.47/1.48	265.96	1.0	1.0	0.81	1.62	3.66	4.72
BCQ	5.44/0.96	267.84	1.0	1.0	0.77	1.85	3.66	4.49
DQN offline	5.44/3.1	250.58	1.0	1.0	0.94	1.1	3.26	5.68
LSTM A2C	5.35/3.6	238.63	1.0	1.0	0.96	1	3	6
Hard Slicing	5.11/-1.53	256.87	1.0	1.0	0.55	3.33	3.33	3.33
No Slicing	7.39/1.13	460.06	1.0	1.0	0.80	0.33	7.20	1.77

Findings

- RL algorithms generally improve SSR of URLLC slice
- CQL and BCQ failed to converge - lower reward than DQN and LSTM A2C. Insufficient BW to URLLC slice
- RL algorithms have lower SE due to low time and frequency resolution in bandwidth allocation.

Need for RL algorithms to allocate bandwidth at finer resolution.