Decision Transformer for Dynamic Radio Resource Management in Network Slicing

Harun Ur Rashid, Seong Ho Jeong*
Dept. of Information and Communications Engineering
Hankuk University of Foreign Studies (HUFS)
Seoul, Korea
Email: harun@hufs.ac.kr, *shjeong@hufs.ac.kr

***Abstract*— The rapid diversification of services in 5G and beyond networks, including enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and voice services (VoLTE), calls for more intelligent and adaptive approaches to radio resource management. Network slicing provides a way to logically partition network resources to meet these different service requirements. However, static allocation, often referred to as hard slicing, cannot respond to fluctuating traffic conditions. This limitation results in inefficient use of spectrum and a decline in the Quality of Experience (QoE) for users. Traditional reinforcement learning (RL) techniques have been explored to address this challenge, but their dependence on unstable value-function optimization and the need for online interaction make them difficult to deploy in highly variable or safety-critical wireless environments. To address these issues, this paper presents a new method for dynamic radio resource allocation on network slices using a Decision Transformer (DT), an offline RL framework that frames the slicing task as a sequence modeling problem. The DT learns to distribute bandwidth across service slices in a way that maximizes both spectral efficiency and user satisfaction. The model is trained entirely on an offline dataset created from a combination of expert heuristic policies. This training approach provides exposure to diverse and high-quality behaviors without requiring unsafe or costly online exploration. Simulation results show that the proposed DT-based method achieves clear improvements over conventional hard slicing and traditional RL-based approaches. The system delivers higher overall utility and better service-level success rates for VoLTE, eMBB, and URLLC traffic.**

Keywords*— *5G, Network Slicing, Decision Transformer, Offline Reinforcement Learning, Radio Resource Management.

I. INTRODUCTION

Modern 5G networks are designed to support an increasingly diverse set of services, each with unique performance and expectations [1]. Enhanced Mobile Broadband (eMBB) requires consistently high data rates to support bandwidth-intensive applications such as high-definition video streaming. Ultra-Reliable Low-Latency Communications (URLLC) demands extremely low delay and high reliability for mission-critical applications, including autonomous systems and industrial automation. Massive Machine-Type Communications (mMTC) involves connecting large numbers of low-power devices [2]. In addition to these service classes, operators must also support legacy traffic such as Voice over LTE (VoLTE). Accommodating these heterogeneous requirements on a common physical infrastructure makes radio resource management (RRM) increasingly challenging [3].

Network slicing has emerged as one of the most promising mechanisms to address this challenge. By creating multiple virtualized, end-to-end networks over a shared physical infrastructure [4], slicing allows operators to tailor resources to the needs of different service categories. In the Radio Access Network (RAN), this often involves partitioning bandwidth into separate slices. The simplest approach, known as hard slicing, assigns fixed portions of bandwidth to each slice. While easy to deploy, static allocation cannot keep pace with the bursty and unpredictable nature of modern wireless traffic. This mismatch often results in underutilized resources in one slice and unmet demand in another, ultimately degrading the Quality of Experience (QoE). Dynamic resource allocation schemes are therefore essential. Deep Reinforcement Learning (DRL) has been widely explored as a potential solution, owing to its ability to learn effective control strategies in complex environments. However, most DRL approaches rely on online training and require repeated interaction with live network during learning. This dependence can introduce performance instability, long training times, and safety concerns, particularly in operation on cellular systems [5] where exploratory actions may disrupt ongoing services.

To overcome these limitations, this work adopts an offline reinforcement learning (RL) approach and introduces Decision Transformer (DT) architecture [6] for dynamic resource management in network slicing. Instead of learning through trial-and-error in the live network, the DT learns directly from pre-collected datasets, reframing the RL problem as a conditional sequence modeling task. This makes the method naturally compatible with telecommunications systems, where large volumes of network logs can be gathered without interfering with real users. The main contributions of this paper are summarized as follows:

- We formulate dynamic bandwidth allocation across VoLTE, eMBB, and URLLC slices as an offline DRL problem.
- We design and implement a DT that learns a slicing policy from a dataset generated using multiple expert heuristics.
- Through extensive simulations, we demonstrate that our method significantly outperforms static hard slicing and existing RL-based state-of-the-art approaches in terms of overall system utility and per-slice Successful Service Rate (SSR).

II. Related Studies

Research on intelligent resource management on network slice for 5G and beyond 5G networks has expanded rapidly, moving from supervised learning for basic classification tasks to advanced reinforcement learning for dynamic and large-scale decision making. Early supervised approaches [7] primarily addressed foundational functions such as slice identification and traffic categorization. These methods offered useful preprocessing capabilities but were unable to handle the temporal dependencies and rapid fluctuations inherent in real radio access networks, which limits their impact on real-time resource control. To address dynamic resource allocation, most prior work relies on online DRL. For example, GAN (generative adversarial networks)- assisted distributional Q-learning methods [8] attempt to learn full return distributions to improve policy robustness under uncertainty, yet they still require extensive online exploration. Transformer-based architectures [9], [10] have also been integrated into online actor-critic frameworks for tasks such as sequence-aware service function chaining or one-shot slice placement. Although such approaches improve the capacity to model long-range temporal relationships, they inherit fundamental limitations of online RL including sample inefficiency, unstable value-function optimization, and the need for continuous interaction with a live network environment.

Recent studies further illustrate the limitations of online multi-agent reinforcement learning (MARL). The UAV (unmanned aerial vehicle)- assisted slicing framework MADDPG-M&L (Multi-Agent Deep Deterministic Policy Gradient based on Matching Game and Lagrangian Dual) proposes joint user association and slice resource allocation using stable matching and multi-agent policy updates [11]. While effective in dynamic UAV scenarios, the method depends entirely on online MARL, which introduces considerable instability and requires repeated interactions with the environment. This reliance on continuous exploration can be problematic for network slicing where mistaken decisions immediately degrade user experience. Similarly, the heuristic-assisted multi-agent DRL scheme for QoS-security tradeoff in RAN slicing addresses slice isolation and user mobility but still operates purely in an online multi-agent setting [12]. While the method introduces security-aware objectives, its dependence on real-time environment interaction restricts scalability and increases deployment risk. Moreover, the approach uses handcrafted heuristics to guide learning, which may bias the policy and limit its generalization to unseen scenarios.

Another line of work [13] applies MARL frameworks to VLC-NOMA (Visible Light Communication – Non-Orthogonal Multiple Access) environments to jointly manage power allocation and stability concerns such as interference and handovers. These studies demonstrate the ability of MARL to capture complex interactions among multiple access points and users, but they again rely on online updates and full environment availability during training [13]. In practice, gathering such training data is expensive and often unsafe because it requires exploration that can degrade user quality of service (QoS). Only very recent research [14] explores offline learning. An offline multi-agent RL framework [14] has been proposed for radio resource management, showing that policies can be learned entirely from static datasets using conservative Q-learning to mitigate out-of-distribution errors. This work demonstrates the potential of offline MARL for scalability and safety. However, it focuses on generic RRM tasks such as scheduling and power control and does not consider the unique constraints of network slicing, including isolation, service differentiation, or slice-level utility tradeoffs. Furthermore, it does not integrate sequence modeling and therefore cannot exploit temporal structure as effectively as transformer-based approaches.

Therefore, existing research overwhelmingly depends on online RL or online MARL, which increases operational costs, and exposes the network to instability. Transformer-empowered RL is emerging but has not yet been explored in an offline context for network slicing. Literature lacks a framework capable of learning high-quality slicing decisions directly from logged trajectories without online exploration. This gap motivates the use of a Decision Transformer [6] that reframes radio resource allocation as a conditional sequence modeling problem, enabling robust learning from historical multi-slice data while avoiding the pitfalls of traditional online RL.

III. System Model and Problem Formulation

A. Network and Service Model

We model a single-cell downlink scenario where a Base Station (BS) serves multiple User Equipment's (UEs) with diverse service requirements. The system consists of a single BS located at the center of a circular cell, serving N UEs. We consider three distinct service slices: VoLTE, eMBB, and URLLC. Each UE is randomly assigned to one of these slices. Let, $K = (1,2,3)$ denote the set of slices, corresponding to VoLTE, eMBB and URLLC, respectively. The total system bandwidth B_{total} is shared among the slices and dynamically partitioned at each time slot.

Traffic Models: The arrival of data packets for each service follows distinct statistical models based on 3GPP specifications. VoLTE traffic is modeled as a VoIP source, eMBB follows a Pareto distribution to simulate bursty video traffic, and URLLC traffic is based on an exponential model for sporadic, critical data packets [8]. Let $D_k(t)$ denote the total packet arrivals for slice k at time step t.

Channel Model: The channel gain for each UE i, denoted by g_i, incorporates distance-based path loss according to the 3GPP TR 36.814 model [15] and log-normal shadowing. The achievable data rate of UE i is given by

$$R_i = L_i B_k log_2\left(1 + \frac{P_i g_i}{N_0 B_k}\right) \quad (1)$$

where L_i is the number of MIMO layers, B_k is the bandwidth allocated to the slice of UE i, P_i is the transmit power, and N_0 is the noise spectral density.

B. Reinforcement Learning Formulation

We frame the dynamic slicing problem as a Markov Decision Process (MDP), which is defined by a tuple (S, A, P, R, γ).

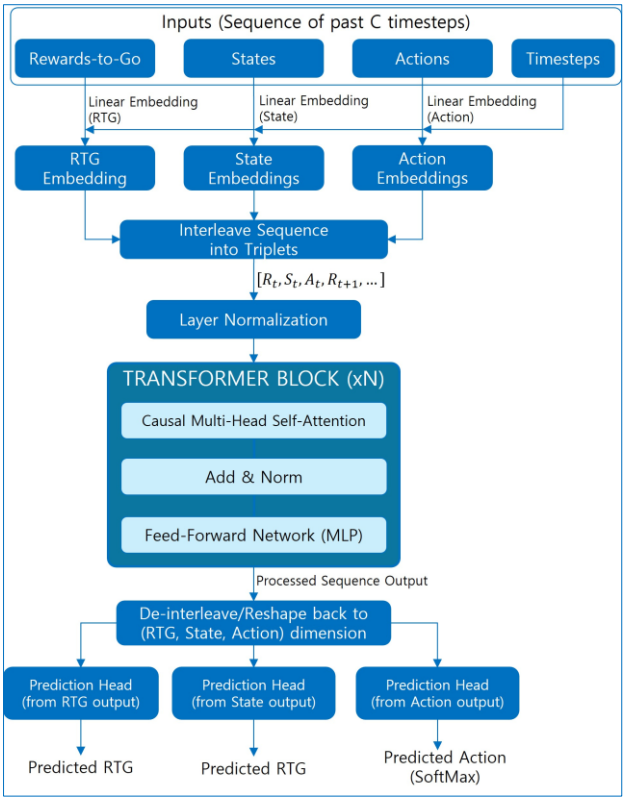


Fig. 1. The Decision Transformer architecture. It takes sequences of rewards-to-go, states, and actions as input and autoregressively predicts the next action that achieves the target return.

State Space (S): The state $s_t \in S$ at time step t must capture the essential network conditions. We define the state as a concatenated vector containing three key metrics for each of the K service slices:

$$s_t = [u_1, \dots, u_K, b_1, \dots, b_K, l_1, \dots, l_K] \tag{2}$$

where, for each slice $k \in \{1, \dots, K\}$, u_k is the number of active UEs, b_k is the average buffer size of active UEs, and l_k is the average packet latency. This state representation provides a comprehensive yet compact snapshot of the current load and QoE for each slice.

Action Space (A): The action $a_t \in A$ is the agent's decision on how to partition the total system bandwidth B_{total}. The action space is discretized, where each action is a vector:

$$a_t = [B_1, B_2 \dots, B_K] \quad \text{s.t.} \sum_{k=1}^{K} B_k = \mathrm{B}_{\text{total}} \tag{3}$$

The action space is discretized using a predefined resolution parameter that determines the minimum allocatable bandwidth unit.

Reward Function (R): To jointly optimize network efficiency and user satisfaction, the reward r_t is defined as a weighted sum of the system-level Spectral Efficiency (SE) and SSR, a QoE metric.

$$r_t = w_{se} \,.\, \mathrm{SE}_t + \sum_{k=1}^{K} w_k \,.\mathrm{SSR}_{k,t} \tag{4}$$

where w_{se} and w_k are tunable weighting factors.
The system spectral efficiency at time t is given by

$$SE_t = \frac{\sum_{i=1}^{N} R_i}{B_{\text{total}}} \tag{5}$$

The SLA Satisfaction Ration for slice k is defined as

$$\mathrm{SSR}_{k,t} = \frac{N_{k,t}^{\text{succ}}}{D_k(t)} \tag{6}$$

where $N_{k,t}^{\text{succ}}$ is the number of packets successfully delivered within the slice-specific rate and latency constraints, and $D_k(t)$ is the total number of packet arrivals for slice k.

IV. Decision Transformer for Resource Slicing

Instead of learning online, our approach uses the decision transformer to learn a slicing policy from a pre-collected dataset of interactions.

A. *Offline Data Generation*

A crucial component of offline RL is the quality of the training dataset. A dataset collected from a purely random policy may lack examples of high-reward behavior. To address this, we generate our dataset using an ϵ-greedy policy built upon a mixture of expert heuristics by using cellular environment [8]. With probability $1 - \epsilon$, the agent executes a random action to ensure exploration. With probability ϵ, it randomly chooses one of three expert policies:

Latency-Aware Expert: Allocates the majority of bandwidth to the slice with the highest average packet latency (prioritizing URLLC).

Buffer-Aware Expert: Allocates the majority of bandwidth to the slice with the largest average data buffer (prioritizing eMBB).

Fairness Expert: Allocates bandwidth as evenly as possible among the active slices.

This approach populates the dataset with diverse and effective trajectories, providing a rich learning signal for the DT model.

B. *Decision Transformer Architecture*

The decision transformer models the joint distribution of a trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots)$ using a GPT-like Transformer architecture. Instead of conditioning on past states to predict an action, the DT conditions on a desired future outcome, specified as the reward-to-go (RTG), $\hat{R}_t = \sum_{t'=t}^{T} r_{t'}$.

The model input at each time step t is a sequence of the last C triplets of RTGs, states, and actions:

$$(\hat{R}_{t-C+1}, s_{t-C+1}, a_{t-C+1}, \dots, \hat{R}_t, s_t, a_t) \tag{7}$$

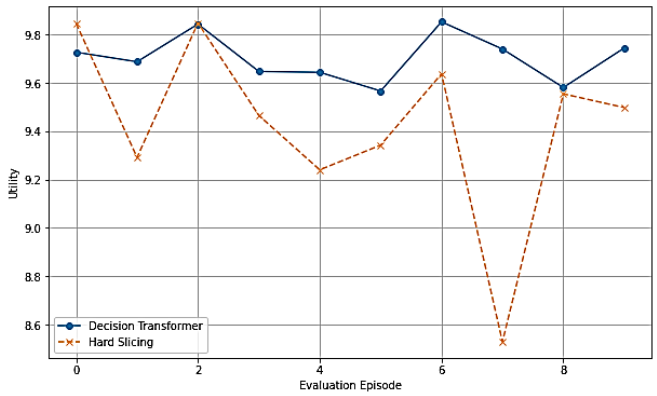


Fig. 2. System utility of hard and DT based slicing baseline

Each modality (RTG, state, action) is first passed through an embedding layer to project it into a high-dimensional space. A timestep embedding is added to each token to provide positional context. The resulting sequence of embeddings is processed by a series of self-attention-based Transformer blocks. The model is trained with a causal self-attention mask to ensure that the prediction for timestep t only depends on past inputs. The DT is trained to predict the next state, the next action, and the next RTG, but during inference, we only use the action head to select the next bandwidth allocation, as shown in Fig. 1. By providing a high target RTG during evaluation, we steer the agent to generate actions that lead to high cumulative rewards.

V. Performance Evaluation

This section presents a comprehensive evaluation of the proposed decision transformer compared with a static hard slicing baseline. All experiments are conducted in a custom Python/TensorFlow simulation environment designed to emulate realistic traffic dynamics and QoE constraints across VoLTE, eMBB, and URLLC services.

A. Simulation Setup

Table I summarizes the key parameters used in our experiments. The DT is trained for 200 epoch using an offline dataset composed of 100 episodes generated by a mixture of heuristic slicing policies. The hard slicing baseline allocates bandwidth equally among the three services at all times, irrespective of traffic variations. The experiment uses 1000kHz action resolution, resulting in a substantially larger action space and making the dynamic allocation problem more challenging.

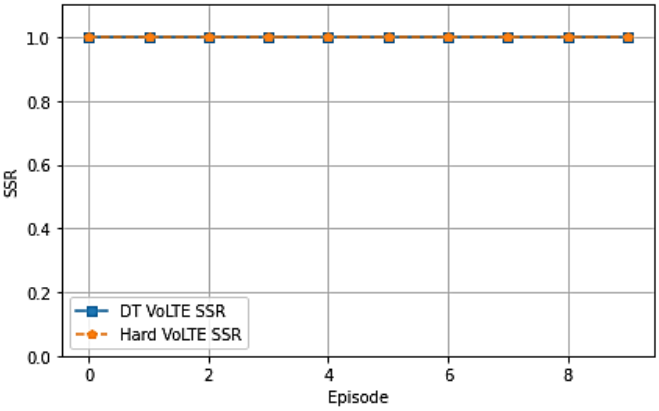
TABLE I. Simulation Parameters

Parameter	Values
Total Bandwidth	20 MHz
BS Transmit Power	46 dBm
Cell Radius	40 m
Number of UEs	100
Service Types	VoLTE, eMBB, URLLC
MIMO Configuration	64 layers
Context Length C	50 timesteps
QoE Weights (w_k)	VoLTE: 1, eMBB: 1, URLLC: 6
SE Weight (w_{se})	0.01

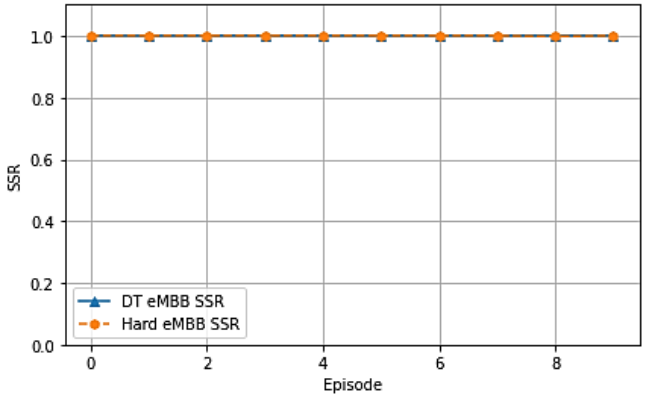
B. Reinforcement Learning Formulation

The DT agent learns a bandwidth allocation strategy by modeling the slicing process as a conditional sequence prediction task. Using only offline trajectories, the agent attempts to maximize a composite utility function showing in Fig. 2 that accounts for both spectral efficiency and slice-level QoE satisfaction. The hard slicing method does not involve learning; it simply enforces a fixed partitioning.

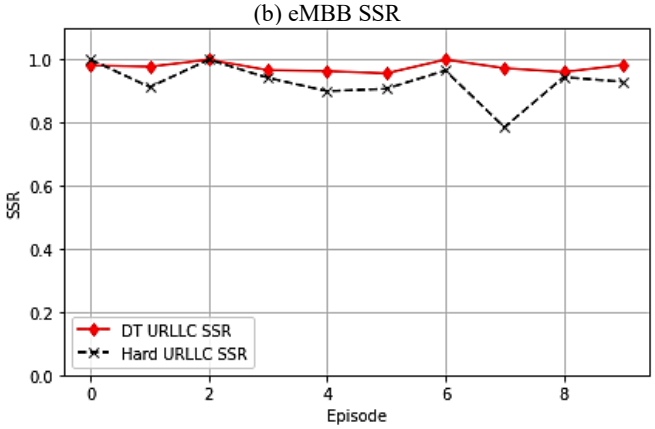
A more detailed view of service-level performance is shown in the per-slice SSR comparisons in Fig. 3. Both approaches maintain near-perfect SSR for the VoLTE and eMBB slices, as these services generally demand less stringent latency guarantees. However, a clear performance gap emerges for the URLLC slice. The DT consistently sustains a high SSR, even during episodes characterized by fluctuating queue lengths or sudden bursts of latency-sensitive packets.



(a) VoLTE SSR

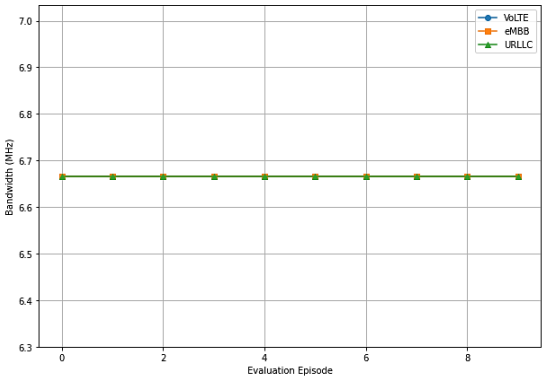


(b) eMBB SSR

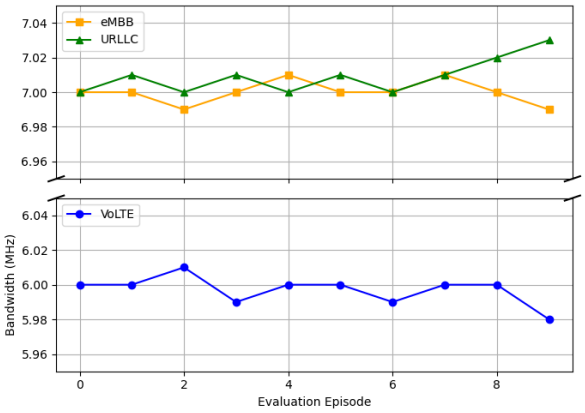


(c) URLLC SSR

Fig. 3. SSR comparison on network slices



(a) Bandwidth allocation per Episode on hard slicing



(b) Bandwidth allocation per Episode on DT

Fig. 4. Dynamic bandwidth allocation over evaluation episode

The higher URLLC SSR achieved by the DT-based approach can be attributed to its trajectory-aware decision-making and conditioning on reward-to-go. By considering the historical state transitions and anticipating future SLA violations, the DT dynamically reallocates bandwidth to URLLC during transient congestion events. In contrast, the hard slicing baseline relies on static resource partitioning and lacks the ability to adapt to sudden bursts of latency-sensitive traffic, resulting in occasional SSR degradation. This instability reflects its inability to temporarily prioritize URLLC traffic in moments where rapid adaptation is essential.

The adaptive behavior of the DT is further illustrated in Fig. 4, which visualizes the evolution of per-slice bandwidth allocation over evaluation episodes. Due to the fixed total system bandwidth constraint, the absolute variations in per-slice bandwidth allocation occur within a narrow numerical range, potentially obscuring meaningful adaptations when plotted on a single continuous axis. To improve interpretability, a broken y-axis is employed, separating the operating ranges of VoLTE and eMBB/URLLC slices.

In Fig. 4(a), the hard slicing baseline assigns static and identical bandwidth shares to all slices, resulting in flat allocation curves across episodes. However, in Fig. 4(b) shows that the DT-based policy dynamically reallocates bandwidth in response to instantaneous network conditions. The upper axis reveals fine-grained bandwidth adjustments between eMBB and URLLC, with URLLC receiving temporary allocation boosts during latency-critical episodes. Meanwhile, the lower axis shows smaller but deliberate VoLTE adjustments, reflecting its relatively stable traffic characteristics.

Although the magnitude of these bandwidth reallocations is small in absolute terms, they are sufficient to significantly impact URLLC latency performance and SSR, as demonstrated in Fig. 3. These results confirm that the DT learns precise and service-aware prioritization policies, rather than relying on static or myopic resource allocation. Its ability to learn from offline data, adapt bandwidth allocation to instantaneous network states, and maintain high per-slice reliability, especially for URLLC - positions it as a promising alternative to both static slicing approaches and traditional online reinforcement learning techniques.

Finally, Fig. 3(a) and Fig. 4(b) should be interpreted jointly. While Fig. 4(b) illustrates how the DT reallocates bandwidth across slices, Fig. 3 reflects whether slice-specific SLA constraints are satisfied. Importantly, SSR is a threshold-based metric: once the minimum rate and latency requirements of a slice are met, additional bandwidth does not further improve SSR. This explains why the DT assigns slightly lower bandwidth to VoLTE compared to hard slicing while still achieving identical VoLTE SSR. In contrast, URLLC SSR is highly sensitive to transient congestion, and the DT's ability to temporarily boost URLLC bandwidth directly translates into improved and more stable URLLC reliability.

C. Simulation Environment and DT Training Details

The above results are obtained using a custom-built system-level simulator designed in [8] to capture the essential dynamics of downlink radio resource management in a network slicing scenario. The simulator models per-slice traffic arrivals based on 3GPP-inspired [15] statistical distributions, realistic channel conditions including path loss and shadowing, per-UE buffering behavior, and slice-specific latency constraints. At each scheduling interval, bandwidth allocation decisions are applied, and packet-level outcomes are tracked to compute latency and SSR metrics.

TABLE II. DECISION TRANSFORMER TRAINING AND INFERENCE CONFIGURATION.

Component	Description
Training Mode	Offline supervised sequence modeling
Training Data	State–action–reward trajectories from simulator
Context Length	50 timesteps
Input Tokens	Reward-to-go, state, action
Training Objective	Minimize action prediction loss
Online Interaction	Not required
Inference Cost	Single forward pass per timestep
Deployment Suitability	Real-time RAN control

The Decision Transformer is trained entirely offline using trajectories collected from the simulator under a mixture of baseline and exploratory slicing policies. Each trajectory consists of sequences of states, actions, and rewards, where rewards encode a weighted combination of spectral efficiency and slice-level SSR. During training, the DT learns to predict bandwidth allocation actions conditioned on the observed state and a desired reward-to-go, without requiring online interaction with the environment.

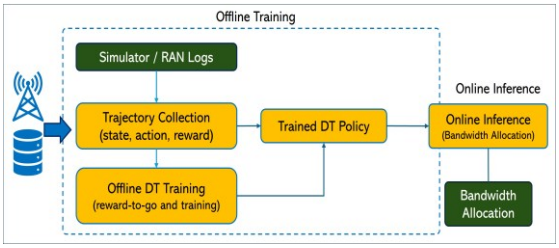


Fig. 5. Offline training and online inference pipeline of the Decision Transformer for dynamic radio resource management

At inference time, the trained DT operates as a lightweight policy that produces bandwidth allocation decisions via a single forward pass, making it suitable for real-time RAN control. No online policy updates or value function evaluations are required. Table II summarizes the DT training and inference configuration used in this work.

Furthermore, while the experiments in this study rely on synthetic traffic and channel models, the proposed framework is not limited to simulated data. In practical deployments, DT training can leverage historical RAN logs containing traffic statistics, buffer states, and QoS measurements, making it particularly well suited for offline, data-driven optimization. Fig. 5 illustrates the overall offline training and online inference pipeline of the proposed DT-based slicing framework. Incorporating real-world network traces is identified as an important direction for future work.

VI. Conclusions

In this paper, we proposed and evaluated a decision transformer for dynamic radio resource slicing in a 5G wireless network scenario. By formulating the problem within an offline reinforcement learning framework, we were able to train a sophisticated policy on a static dataset without the need for risky and inefficient online exploration. Our results clearly show that the DT has learnt an effective, dynamic policy that adapts to fluctuating traffic demands. It significantly outperforms a static hard slicing baseline and state-of-the-art studies [8], delivering higher overall system utility and improved SSR for all service types. Future work will explore the application of this model in more complex multi-cell environments and incorporate additional resource dimensions such as power and scheduling priorities.

Acknowledgment

This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation) – ITRC (Information Technology Research Center) (IITP-2026-RS-2024-00436887, 50%) grant funded by the Korea government (Ministry of Science and ICT). This work was supported by Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0028596).

References

[1] S. Zhang, "An Overview of Network Slicing for 5G," *IEEE Wirel. Commun.*, vol. 26, no. 3, pp. 111–117, June 2019, doi: 10.1109/MWC.2019.1800234.

[2] H. U. Rashid and S. H. Jeong, "AI empowered 6G technologies and network layers: Recent trends, opportunities, and challenges," *Expert Syst. Appl.*, vol. 267, p. 125985, Apr. 2025, doi: 10.1016/j.eswa.2024.125985.

[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network Slicing in 5G: Survey and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017, doi: 10.1109/MCOM.2017.1600951.

[4] C. De Alwis, P. Porambage, K. Dev, T. R. Gadekallu, and M. Liyanage, "A Survey on Network Slicing Security: Attacks, Challenges, Solutions and Research Directions," *IEEE Commun. Surv. Tutor.*, vol. 26, no. 1, pp. 534–570, 2024, doi: 10.1109/COMST.2023.3312349.

[5] N. C. Luong *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019, doi: 10.1109/COMST.2019.2916583.

[6] L. Chen *et al.*, "Decision Transformer: Reinforcement Learning via Sequence Modeling," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2021, pp. 15084–15097. Accessed: Aug. 19, 2025. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html

[7] H. U. Rashid and S. H. Jeong, "Deep Learning-based Network Slice Recognition," in *2023 Fourteenth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2023, pp. 297–299. doi: 10.1109/ICUFN57995.2023.10199606.

[8] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-Powered Deep Distributional Reinforcement Learning for Resource Management in Network Slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020, doi: 10.1109/JSAC.2019.2959185.

[9] C. S.-H. Hsu, A. Dalgkitsis, C. Papagianni, and P. Grosso, "Transformer-Empowered Actor-Critic Reinforcement Learning for Sequence-Aware Service Function Chain Partitioning," Apr. 26, 2025, *arXiv*: arXiv:2504.18902. doi: 10.48550/arXiv.2504.18902.

[10] R. Sahraoui, F. Bannour, O. Houidi, and B. Jouaber, "An Intelligent E2E Network Slicing Framework using Transformer-Enhanced DRL," in *IEEE International Conference on Network Softwarization*, Budapast, Hungary, June 2025. Accessed: Aug. 19, 2025. [Online]. Available: https://hal.science/hal-05113591

[11] G. Chen, F. Sun, H. Liang, Q. Zeng, and Y.-D. Zhang, "MADDPG-M&L: UAV-Assisted Joint User Association and Slicing Resource Allocation in HetNets," *IEEE Trans. Netw. Sci. Eng.*, vol. 12, no. 4, pp. 2878–2894, July 2025, doi: 10.1109/TNSE.2025.3554991.

[12] Y. Sun, Z. Shi, J. Liu, and J. Wang, "Heuristic-Assisted MADRL-Based Resource Allocation Scheme for QoS-Security Tradeoff in RAN Slicing With User Mobility," *IEEE Trans. Wirel. Commun.*, vol. 24, no. 10, pp. 8863–8877, Oct. 2025, doi: 10.1109/TWC.2025.3569346.

[13] A. A. Al-Hameed, S. Hafeedh Younus, M. A. Ahmed, and A. Baz, "Optimizing Resource Allocation for QoS and Stability in Dynamic VLC-NOMA Networks via MARL," *IEEE Access*, vol. 13, pp. 151258–151274, 2025, doi: 10.1109/ACCESS.2025.3599664.

[14] E. Eldeeb and H. Alves, "An Offline Multi-Agent Reinforcement Learning Framework for Radio Resource Management," Jan. 22, 2025, *arXiv*: arXiv:2501.12991. doi: 10.48550/arXiv.2501.12991.

[15] Technical Report (TR) 38.843, *Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects*, Jan. 22, 2015. Accessed: Dec. 08, 2025. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2493