Intelligent Optimization of Wireless Access Point Deployment for Communication-Based Train Control Systems Using Deep Reinforcement Learning

Kunyu Wu, Qiushi Zhao, Zihan Feng, Yunxi Mu, Hao Qin, Xinyu Zhang, and Xingqi Zhang *Senior Member, IEEE*

***Abstract*—Urban railway systems increasingly rely on communication-based train control (CBTC) systems, where optimal deployment of access points (AP) in tunnels is critical to ensure robust wireless coverage. Traditional methods, such as empirical model-based optimization algorithms, suffer from excessive measurement requirements and suboptimal solutions, while machine learning (ML) approaches often face limited adaptability in complex tunnel environments. This paper proposes a deep reinforcement learning (DRL)-driven framework that integrates parabolic wave equation (PWE) channel modeling methods, conditional generative adversarial network (cGAN)-based data augmentation, and a dueling deep Q-network (Dueling DQN) for AP placement optimization in tunnels. The PWE method produces high-fidelity path loss distributions for a small subset of AP positions, which are then expanded by the cGAN to generate realistic high-resolution path loss maps for all candidate positions, significantly reducing simulation cost while preserving physical accuracy. In the proposed DRL formulation, the state space captures AP positions and signal coverage, the action space defines discrete AP adjustments, and the reward function incentivizes signal improvement while penalizing deployment costs. The dueling DQN architecture decouples state value and action advantage estimation to enhance convergence speed, improve exploration–exploitation balance, and increase the likelihood of reaching globally optimal configurations. Comparative experiments against a conventional Hooke–Jeeves (HJ) optimizer and a traditional DQN demonstrate that the proposed hybrid method achieves superior performance, delivering AP configurations with higher average received power and better worst-case coverage, while being more computationally efficient. This work bridges high-fidelity electromagnetic simulation, generative modeling, and AI-driven optimization, offering a scalable and data-efficient solution for next-generation CBTC systems in complex tunnel environments.**

***Index Terms*—Communication-based train-control system, deep reinforcement learning, parabolic wave equation, wave propagation modeling, wireless access point.**

Received XXX; revised XXX. This work was supported in part by the XXX. (Kunyu Wu and Qiushi Zhao are equally contributed to this paper. Corresponding author: Hao Qin.)

Kunyu Wu, Qiushi Zhao and Zihan Feng are with the School of Electronics and Information Engineering, Sichuan University, Chengdu, 610017, China.

Hao Qin is with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8, Ireland, and also with the School of Electronics and Information Engineering, Sichuan University, Chengdu, 610017, China (e-mail: hao.qin@scu.edu.cn).

Yunxi Mu is with the College of Engineering, Peking University, Beijing, China.

Xingqi Zhang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada, and also with the School of Electrical and Electronic Engineering, University College Dublin, Ireland (e-mail: xingqi.zhang@ualberta.ca).

Xinyue Zhang is with the School of Electrical and Electronic Engineering, University College Dublin, Ireland (e-mail: xinyue.zhang@ucd.ie).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier: XXXX

I. Introduction

Urban railway systems are increasingly dependent on communication-based train control (CBTC) systems to ensure safety and efficiency [1]–[5], where access points (APs) serve as critical bridges between trains and ground infrastructure. In railway tunnels characterized by confined electromagnetic wave-attenuating environments, the optimal placement of APs is paramount to guarantee continuous high-quality wireless coverage for real-time data transmission and control [6], [7].

From a system design perspective, CBTC imposes stringent requirements on communication latency and reliability. This makes exhaustive electromagnetic simulations for all possible AP positions computationally prohibitive, especially when using high-fidelity models such as the parabolic wave equation (PWE) [8]–[13]. Although these simulations capture complex propagation effects with physical accuracy, the computational burden is incompatible with the rapid iteration cycles often required during network planning and real-time reconfiguration.

Traditional AP deployment strategies, such as empirical path loss models or heuristic optimization methods, face limitations that include oversimplified channel assumptions, heavy measurement requirements, or susceptibility to local optima. Purely data-driven machine learning (ML) approaches offer faster prediction once trained but often struggle to generalize to spatially varying tunnel characteristics and may fail to capture critical physical effects.

To address these challenges, we propose augmenting the limited set of PWE-simulated path loss (PL) maps with a conditional generative adversarial network (cGAN) that learns to generate high-resolution, physically consistent PL distributions for untested AP positions. This PL data augmentation substantially reduces the simulation time while maintaining the physical realism required for CBTC-grade network planning. The augmented dataset enables a deep reinforcement learning (DRL) agent to explore a richer set of AP configurations without incurring the prohibitive cost of exhaustive PWE simulations.

We adopt a dueling deep Q-network (Dueling DQN) as the optimization engine, which decouples state value and action advantage estimation to improve convergence speed and exploration–exploitation balance in complex tunnel environments [14], [15]. In addition to the standard DRL formulation, we define a CBTC-specific interaction environment and discrete action space tailored to AP deployment in tunnels, ensuring that the learned strategies directly address the coverage and latency constraints of railway operations.

A. Prior Work

In recent years, the CBTC system has been widely used for urban rail transportation [16]. Access points, which connect wireless devices to the wired network, play a vital role in the CBTC system. Once the train crosses the signal coverage area of the AP, bidirectional communication is established between the train and the ground [6]. When deploying APs, it is critical to find AP positions that ensure efficient and stable communication, particularly in railway tunnels, which are dark and enclosed spaces. Hence, an algorithm tailored to optimize AP placement in railway tunnel environments is imperative. In wireless communication systems, there are two modules that must be addressed when developing an AP placement optimization algorithm: a channel model and an optimization algorithm [17].

For the channel model, an extensively utilized method in AP positioning optimization within the wireless domain, called the empirical path loss model, can be adopted [18], [19], and it must be simple, accurate and general [20]. The model relies firmly on measurement results [21], [22], thus requiring massive measurements. In contrast, a PWE method based on computational simulation can also be accurately applied to channel model development [23], [24]. This method is suitable for addressing complex terrain and long-distance propagation [9], particularly exhibiting high computational efficiency [8]. However, the PWE method often entails significant computational complexity, particularly when high accuracy is required. In that case, an optimization strategy capable of mitigating computational complexity without significant loss in accuracy thus becomes necessary. Our strategy is to predict explicit and elaborate path loss distribution within our research region by a few PWE-generated path loss values, and several predicting methods have utilized in the process of propagation modeling [25]–[27]. In addition, deep learning-based prediction methods-such as convolutional neural networks (CNNs) and generative adversarial networks (GANs)-have been proven useful [28], [29]. Meanwhile, GAN, which tends to generate data with random and uncontrollable categories, can be addressed by the conditional generative adversarial network (cGAN), a model that incorporates conditional information to better leverage the strengths of supervised learning in generative tasks, thereby enhancing the matching accuracy between generated data and target classes [30], [31]. Additionally, cGAN-based prediction approaches have been proven effective and accurate in predicting physical parameters within propagation modeling, such as path loss [32].

Numerous methods exist to optimize the AP positions. A method called Hooke and Jeeves (HJ) optimization is proposed, which moves APs by setting certain step sizes and compares a cost function generated by the path loss [17]. This method is not only effective in simulations, but is also validated in a real subway tunnel environment [17]. In addition, equally distributed placement (EDP), optimal placement (OP) method based on sequential quadratic programming (SQP) to solve the constrained nonlinear optimization problem, hybrid placement (HP) method integrating both the above mentioned approaches, and measurement-based placement (MBP) by placing APs step by step, measuring path loss, and selecting the point with the maximum PL to deploy the next AP are still useful to optimize the placement of the AP along the track [7]. Furthermore, the equally distributed placement (EDP) method, particle swarm optimization (PSO) and the combined strategy are proposed to determine the location of APs while deployed in underground mines [33]. Meanwhile, the optimization method based on the multi-objective genetic algorithm (MOGA) has also played an optimal role in the placement of wireless local area networks (WLANs) [34]. In the area of radio wave transmission, plenty of machine leaning (ML) algorithms are used for optimization [35]–[37]. Among them, indoor positioning is achieved using machine learning models, such as learning based on received signal strength (RSS) maps to estimate user locations. This includes both regression methods for calculating user coordinates and classification methods for determining the user's subspace. Furthermore, [35], [38] discusses various ML methods applied to radio wave research, which provides inspiration to solve the optimization problem of AP placement in tunnel environments. Among them, a deep Q-learning network (DQN) method belonging to the ML category is applied to the location optimization problem. DQN based on the epsilon-greedy strategy is used to optimize security container placement [39]. In addition, DQN is also combined with the whale optimization algorithm (WOA) to address the deployment problem of fiber Bragg grating (FBG) sensors in tunnels [40]. Meanwhile, the cloud native network function (CNF) deployment problem is also solved by the DQN algorithm [41]. When using DQN for the planning of the path of unmanned aerial vehicles (UAVs) [5], DQN is optimized into the dueling DQN method, achieving better results. Therefore, when solving the AP deployment problem, dueling DQN method can be used as an alternative to the traditional DQN method.

B. Contributions

Based on the aforementioned facts, we propose a framework based on deep reinforcement learning. We dedicate ourselves to using PWE to generate the electric field distribution in railway tunnel environments and applying DQN network to optimize the AP positions to enhance the received power of the receivers on the trains. To overcome the high computational cost of generating complete PWE datasets, we introduce a conditional GAN-based path loss data augmentation scheme, enabling realistic, high-resolution PL map generation from a small set of simulations. On top of the standard DRL architecture, we design a CBTC-specific interaction environment and action space that reflect operational constraints and latency

requirements. The main contributions of this paper are as follows:

1) We propose a hybrid AP placement optimization framework that combines high-fidelity path loss data generated by the parabolic wave equation method with a conditional GAN-based data augmentation module. The cGAN learns to generate realistic PL maps for untested AP positions, substantially reducing the computational burden of exhaustive PWE simulations.
2) We design a CBTC-specific deep reinforcement learning environment, defining tailored state and action spaces that reflect tunnel geometry, coverage requirements, and latency constraints. This customization ensures that the learned AP placement strategies are directly applicable to real-world CBTC operations.
3) We employ a dueling deep Q-network to perform AP placement optimization, leveraging its decoupled estimation of state value and action advantage to enhance convergence speed, improve exploration–exploitation balance, and avoid suboptimal local solutions.
4) We conduct comprehensive comparisons against Hooke and Jeeves optimization and traditional DQN methods. Results demonstrate that our cGAN-Dueling DQN framework achieves superior average and worst-case received power performance, validating its ability to produce cost-efficient and robust AP deployment strategies.

II. Systme Model and Problem Formulation

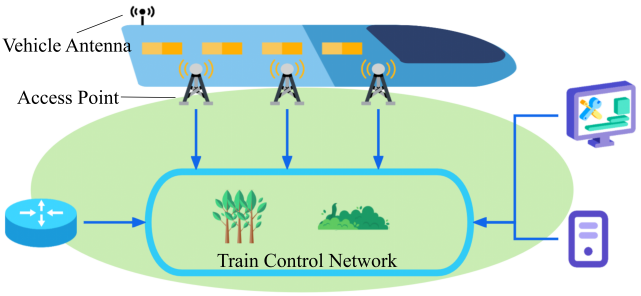

Fig. 1: Illustration of communication-based train-control system.

The key components of a CBTC system include on-board controllers, wayside infrastructure, and communication networks (e.g., LTE, Wi-Fi), as shown in Figure 1. Access point is the core equipment of wayside infrastructure to complete real-time interaction with the speeding train. To receive the signal transmitted from the wayside APs, a vehicle antenna is deployed in the specific location on the train. In a train tunnel scenario, a certain number of access points are installed every few hundred meters on either side of the rail tracks, with transmitting antennas attached to them. As demonstrated in Figure 2, when a train moves along the rail tracks, the vehicle antenna can be seen as a series of receivers placed along the route of the train since the speed of the train is quite fast. The entire network of access points is responsible for serving all the receivers along the train tunnel. During signal transmission, path loss occurs due to signal attenuation, which means that the signal power arrived at the remote receivers could not meet the demand for real-time communication between access points and the speeding train. Therefore, the optimization problem is to deploy the access points in a way that minimizes the average path loss at each receiver points and prevent extremely high path loss somewhere remote of the receiver points from happening.

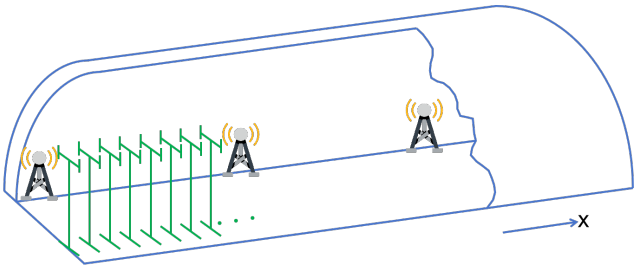

Fig. 2: Simplified CBTC tunnel model.

A. Propagation Environment Model

In this optimization problem, we consider a relatively simple circumstance with a crooked tunnel and the access points are placed along one-dimensional line. With the tunnel topology and the specifications of transmitting antennas and receiving antennas, the received power was obtained at each receiver point by using the method of PWE. The received power is taken as the power received by the antenna on the train, and for a given location of a access point, the path loss(in dB) at the i^{th} receiver point can then be calculated by:

$$\text{pl}_i = P_0 - P_i \tag{1}$$

where P_0 is the transmit power from the access point, P_i is the power received at the i^{th} receiver location.

When there are multiple access points deployed in the tunnel, each AP will serve every receiver points along the curved tunnel, however, the eventually signal behavior of each receiver point is determined by the largest power transmitted from one of the access points, corresponding to lowest path loss. Therefore, for a specific deployment of certain number of access points, the path loss at each receiver point is given by:

$$\text{PL}_i(\mathbf{X}) = \min_j \{\text{pl}_{i,j}(x_j)\} \tag{2}$$

where $\text{pl}_{i,j}(x_j)$ is the path loss arrived at the i^{th} receiver location generated by j^{th} AP. $\text{PL}_i(\mathbf{X})$ represents the ultimately path loss at each receiver point under current AP positioning. $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ denotes the coordinate of n access points along the tunnel direction.

B. Optimization Model

In order to obtain the optimal spacial position of access points mathematically, cost functions of path loss regarding to the deployment position of APs should be formulated.

We construct two type of cost functions that represent the signal power received by receiver points: average path loss and maximum path loss.

The first type of cost function is defined as

$$g_1(\mathbf{X}) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{PL}_i(\mathbf{X}) \tag{3}$$

where m is the total number of sampling points of the receiver. This kind of cost funciton tend to appraisal the average behavior of the signal power arrived at each receiver points. It aims to elevate the overall condition of the communication system. However, it has a drawback that can not prevent some of the remote receiver points from having extremely high path loss, which is bad for the whole system.

The second type cost function is defined as

$$g_2(\mathbf{X}) = \max_i \{\mathrm{PL}_i(\mathbf{X})\} \tag{4}$$

This kind of cost function ensures that the worst circumstance of the path loss still below some number of threshold, compensating the weakness of the first cost function.

In this train communication system in tunnel, every position that vehicle antenna passed through should at least remain some value of signal power, that is, the path loss of each receiver point should always smaller than a treshold value, in order to keep the train running safely:

$$\mathrm{PL}_i(\mathbf{X}) \leq th_i \tag{5}$$

where th_i is the maximum tolorated path loss of the i^{th} receiver point. This constraint can be incorporated in the previous cost function as a penalty term. If the path loss of i^{th} receiver point exceeds the presuppose threshold, penalty should be add to the cost function to make it larger. If the path loss does not break the limit, then nothing would be added. By adopting this method, our model ensures the signal quality of every receiver point and improve the convergence of the optimization process. Eventually, the two cost function can then be represented as:

$$f_1(\mathbf{X}) = \frac{1}{m}\sum_{i=1}^{m} \left[\mathrm{PL}_i(\mathbf{X}) + \lambda_i \max\{0, \mathrm{PL}_i(\mathbf{X}) - th_i\}\right] \tag{6}$$

$$f_2(\mathbf{X}) = \max_i \left[\mathrm{PL}_i(\mathbf{X}) + \lambda_i \max\{0, \mathrm{PL}_i(\mathbf{X}) - th_i\}\right] \tag{7}$$

where λ_i is the penalty coefficient for violating prescribed path loss threshold at the i^{th} receiver point.

In this paper,a convex combination of these two cost functions is adopted. We defind α as the convex combinational weight, and its value can be adjusted by system user at the range of $[0, 1]$ according to the operating requirements. Thus, the convex combination of cost functions can be formulated as:

$$\min \ \alpha f_1(\mathbf{X}) + (1-\alpha) f_2(\mathbf{X}) \tag{8}$$

subject to

$$0 \leq x_j \leq L_j, \ j = 1, \ldots, n$$

where L_j denotes the total length of the curved tunnel, providing limitation to this convex combination optimizaiton problem. This combination method endows our system with more comprehensive communication behaviour, guaranteeing all received power overtop some value of threshold while maintaining a relatively low average path loss simultaneously.

We summarize the optimization problem of AP location as the following formula:

$$(\mathrm{P1}): \min_{\mathbf{X}} [\alpha f_1(\mathbf{X}) + (1-\alpha) f_2(\mathbf{X})] \tag{9}$$

$$\mathrm{s.t.f_1}(\mathbf{X}) = \frac{1}{\mathrm{m}}\sum_{\mathrm{i=1}}^{\mathrm{m}} [\mathrm{PL_i}(\mathbf{X}) + \lambda_\mathrm{i} \max\{0, \mathrm{PL_i}(\mathbf{X}) - \mathrm{th_i}\}] \tag{9a}$$

$$\mathrm{f_2}(\mathbf{X}) = \max_{\mathrm{i}} [\mathrm{PL_i}(\mathbf{X}) + \lambda_\mathrm{i} \max\{0, \mathrm{PL_i}(\mathbf{X}) - \mathrm{th_i}\}] \tag{9b}$$

$$\mathrm{PL_i}(\mathbf{X}) = \min_{\mathrm{j}} \{\mathrm{pl_{i,j}}(\mathrm{x_j})\} \tag{9c}$$

$$\mathrm{pl_{i,j}}(\mathrm{x_j}) = \mathrm{P}_0 - \mathrm{P_{i,j}} \tag{9d}$$

$$0 \leq \mathrm{x_j} \leq \mathrm{L_j}, \quad \forall \mathrm{j} = 1, 2, \ldots, \mathrm{n} \tag{9e}$$

$$\mathrm{PL_i}(\mathbf{X}) \leq \mathrm{th_i} \tag{9f}$$

$$\alpha \in [0, 1] \tag{9g}$$

III. Deep Q-Learning Network

A. Basic Conceptions

Reinforcement learning is a type of machine learning method that involves learning how to map states to actions in order to maximize the rewards obtained. An agent in this context needs to continuously experiment within its environment, using feedback (rewards) from the environment to continuously optimize the mapping between states and actions.

Markov Decision Process (MDP) is a mathematical modeling framework for RL. It is a standard model for describing sequential decision-making problems, defined by the five-tuple $\mathcal{M} = (S, A, P, R, \gamma)$:

- **State (S)**: The set of discrete/continuous states of the environment (e.g., the joint angles of a robot, the game screen).
- **Action (A)**: The set of actions that the agent can perform (e.g., movement commands of a robotic arm, game controller operations).
- **Transition Probability (P)**: The conditional probability of state transition $P(s'|s, a)$ (the dynamic model of the environment).
- **Reward (R)**: The immediate reward function $R(s, a, s')$ (defining the task objective).
- **Discount Factor (γ)**: The discount factor for future rewards (balancing short-term and long-term benefits).

Q-learning is a classic model-free algorithm in Reinforcement Learning (RL), used to find the optimal policy in discrete state and action spaces, addressing the Markov decision problems in reinforcement learning. The core of Q-learning is the Q-table (state-action value table), which stores the Q-value $Q(s, a)$ corresponding to each state s and action a. The Q-value represents the maximum cumulative reward (discounted) that can be obtained by following the current policy after performing action a in state s.

The update of Q-values is based on the Bellman Equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right] \tag{10}$$

where α is the learning rate, controls the step size of the update. γ denotes discount factor, balances the importance of current and future rewards ($0 \leq \gamma < 1$). r_t is the immediate reward obtained after performing action a_t.

However, the Q-learning algorithm is not suitable for our problem scenario due to its inability to handle high-dimensional state spaces, which may encounter the curse of dimensionality—specifically, the number of required state-action pairs grows exponentially with the increase in the dimensionality of the state space. Therefore, we introduce Deep Q-learning Network, a combination algorithm of Q-learning and artificial neural network. DQN utilizes neural network to replace the Q-table in Q-learning algortithm with the network update mode:

$$Q(S_t, A_t, w) \leftarrow Q(S_t, A_t, w) + \alpha \left[R_{t+1} + \gamma \max_{a} \hat{q}(s_{t+1}, a_t, w) - Q(S_t, A_t, w) \right] \tag{11}$$

where w is the network parameter.

The ε-greedy strategy commonly used in DQN is a key mechanism for balancing exploration and exploitation. This strategy selects actions randomly with probability ε to explore actions that are not well understood, thus preventing the agent from falling into local optima; at the same time, it exploits with probability $1-\varepsilon$ by choosing the action with the highest estimated value from the current Q-network. The mathematical expression is as follows:

$$\pi(a|s) = \begin{cases} \arg\max_{a'} Q(s, a'; \theta) & \text{with } 1-\varepsilon \\ \text{random action} & \text{with } \varepsilon \end{cases} \tag{12}$$

When executing, for the current state s, a random number ξ is generated. If $\xi < \varepsilon$, an action is selected randomly from the action space; if $\xi \geq \varepsilon$, the action that maximizes $Q(s, a; \theta)$ is chosen. In our work, we adopt exponential decay method to update ε:

$$\varepsilon_t = \varepsilon_{\min} + (\varepsilon_{\max} - \varepsilon_{\min}) \cdot e^{-\lambda t} \tag{13}$$

This strategy is significant as it not only prevents the agent from relying too early on suboptimal strategies but also ensures convergence through exploration.

B. CBTC Environment Definition for Deep Q-Network

In our context of the railway tunnel communication system, the AP position problem is a Markov decision problem, for the next state AP position is solely dependent on the current state AP position and its consequent path loss generated at each receiver point. Here in our problem setting, a 1500-meters-long train tunnel is considered, and 3 access points are then managed to be placed in this tunnel. At the beginning, we first determine the initial points of positioning, say $[0, 500, 1000]$, and then define the elemantary process of MDP.

In the current state t, the deployment of three APs with a 3-dimensional positioning vector $\langle \boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3} \rangle$ produces a cost function g that we previously defined in Section II. A state at time t is represented as:

$$s_t = \begin{pmatrix} x_1^{(t)} \\ x_2^{(t)} \\ x_3^{(t)} \\ g^{(t)} \end{pmatrix}, \quad \begin{cases} x_i^{(t)} \in [0, L] \\ g^{(t)} = f(x_1^{(t)}, x_2^{(t)}, x_3^{(t)}) \end{cases} \tag{14}$$

where $x_i^{(t)}$ is the position of the i-th AP.

At each state transition step, each AP has three possible movement strategies. In time t, the precise positions of the three APs generate a cost function $g^{(t)}$. A deep neural network drives the transition from the current state t to the next state $t+1$ according to the state transition function. Each AP can move left by one step, move right by one step, or remain stationary, corresponding to the following mappings:

$$\text{Left move} \mapsto -1, \quad \text{Right move} \mapsto +1, \quad \text{No move} \mapsto 0.$$

Consequently, each state transition involves **27 possible actions**, represented by the action set:

$$\mathcal{A} = \{-1, 0, +1\}^3 = \{(a_1, a_2, a_3) \,|\, a_i \in \{-1, 0, +1\},\ i = 1, 2, 3\}. \tag{15}$$

To ensure compatibility with discrete action reinforcement learning algorithms, the 3D action vector is bijectively mapped to an integer index. This establishes a one-to-one correspondence between action vectors and indices, enabling seamless integration with policy networks. Each action is represented as a 3D vector $\mathbf{a} = (a_1, a_2, a_3)$, where $a_i \in \{-1, 0, +1\}$ denotes the direction of movement of the i th AP. The encoding function $\phi : \{-1, 0, +1\}^3 \rightarrow \{0, 1, \ldots, 26\}$ converts the vector into an integer index a_{idx} using:

$$\phi(a_1, a_2, a_3) = (a_1 + 1) \cdot 9 + (a_2 + 1) \cdot 3 + (a_3 + 1). \tag{16}$$

This transformation shifts each component from $\{-1, 0, +1\}$ to $\{0, 1, 2\}$, interprets them as digits in a base-3 numeral system, and computes their base-10 equivalent. The inverse mapping $\phi^{-1} : \{0, 1, \ldots, 26\} \rightarrow \{-1, 0, +1\}^3$ reconstructs the original action vector from the integer index via:

$$\begin{cases} a_1 = \left\lfloor \frac{a_{\text{idx}}}{9} \right\rfloor - 1, \\ a_2 = \left\lfloor \frac{a_{\text{idx}} \mod 9}{3} \right\rfloor - 1, \\ a_3 = (a_{\text{idx}} \mod 3) - 1, \end{cases} \tag{17}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation. This process reverses the base-3 decomposition to recover the original action components. The bijective nature of the mapping guarantees invertibility without information loss, while its reliance on arithmetic operations ensures computational efficiency. The integer indices align directly with the discrete action outputs of DQN policies, bridging the gap between multidimensional movement logic and reinforcement learning requirements.

The state transition function governs how the positions of the three Access Points (APs) evolve from time step t to $t+1$.

For each AP i ($i = 1, 2, 3$), its updated position $x_i^{(t+1)}$ is computed as:

$$x_i^{(t+1)} = \text{clip}\left(x_i^{(t)} + a_i \cdot \Delta, 0, L\right), \tag{18}$$

where:

- $x_i^{(t)}$ denotes the position of the i-th AP at time t,
- $a_i \in \{-1, 0, +1\}$ represents the discrete movement action for the i-th AP (left, stationary, or right),
- Δ is the predefined step size (hyperparameter in neural network),
- L is the tunnel length (e.g., 1500 meters),
- $\text{clip}(v, v_{\min}, v_{\max})$ enforces boundary constraints by clipping the value v to the range $[v_{\min}, v_{\max}]$:

$$\text{clip}(v, v_{\min}, v_{\max}) \triangleq \max(v_{\min}, \min(v, v_{\max})). \tag{19}$$

This function ensures that AP movements remain confined within the spatial limits of the tunnel ($0 \le x_i^{(t+1)} \le L$). The term $a_i \cdot \Delta$ adjusts the position by one step left ($-\Delta$), right ($+\Delta$), or no movement (0), depending on the action. The clipping operation guarantees physical feasibility, preventing APs from moving beyond the tunnel boundaries.

The immediate reward r_t is defined as the negative value of the cost function at the next state:

$$r_t = -g^{(t+1)} \tag{20}$$

where $g^{(t+1)}$ quantifies the coverage quality after transitioning to state $t+1$. This incentivizes the agent to minimize the cost function, as higher rewards correspond to lower coverage costs.

An episode terminates when the coverage quality meets the predefined threshold:

$$\text{Terminate if } g^{(t)} < 20 \tag{21}$$

This ensures the process stops once the AP placements achieve satisfactory coverage (cost below 20), concluding the optimization episode. Thus far, the AP placement optimization model for train tunnels and its DQN optimization algorithm have been fully presented. The simplified illustrative expressions are as follows:

$$\begin{pmatrix} x_1^{(t)} \\ x_2^{(t)} \\ x_3^{(t)} \\ g^{(t)} \end{pmatrix} \xrightarrow{\phi^{-1}(a_{\text{idx}})} \begin{pmatrix} x_1^{(t+1)} \\ x_2^{(t+1)} \\ x_3^{(t+1)} \\ g^{(t+1)} \end{pmatrix}$$

with transitions governed by $x_i^{(t+1)} = \text{clip}\left(x_i^{(t)} + a_i\Delta\right)$. The general procedure for finding optimal access point locations is shown in Figure 3. In our problem setting, we fix 3 access points for convenience along a 1500m-long curved train tunnel.

Parameter	Value
Learning rate	0.001
γ	0.995
ε (epsilon)	1
ε-decay (epsilon decay)	0.995
ε-min (minimal epsilon)	0.01
State size	4
Action size	27
Episodes	1500
Batch size	64

TABLE I: DRL algorithm parameters.

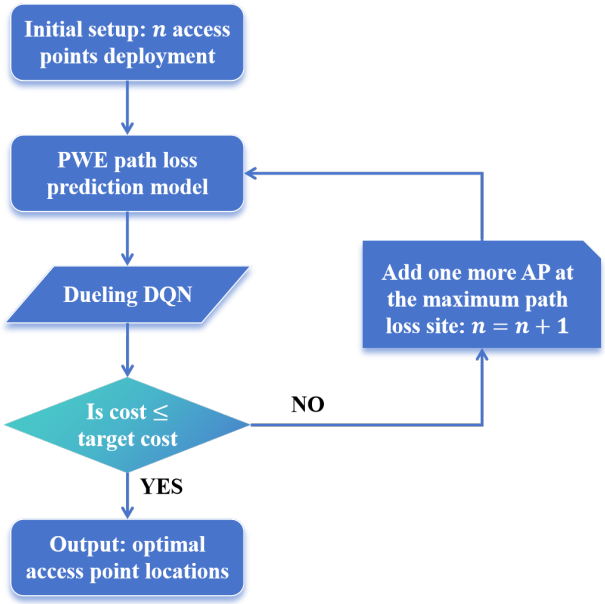

Fig. 3: Diagram of the optimization procedure.

IV. Proposed AP Placement Optimization Approach for CBTC Systems

In our tunnel AP deployment optimization problem, the standard DQN faces two core challenges: First, the traditional Q-network couples the inherent state value with relative action advantages, making it difficult to accurately distinguish subtle differences between actions, especially during fine-tuning stages when AP positions approach optimality. When the spacing between three APs is close to the optimal configuration, the differences in the improvement of the cost function g among 27 possible actions may be minor. Under such conditions, standard DQN exhibits oscillatory Q-DRL algorithms. Second, in sparse reward environments (where reward signals flatten as g approaches optimal values), traditional architectures struggle to capture the intrinsic value of states, leading to ambiguous policy updates. Based on the potential risks of the standard DQN mentioned above, we propose an improved method, Dueling DQN, to improve our neural network. The framework of the proposed approach of AP placement strategy is shown in Figure 4.

As illustrated in Figure 5, the innovation of Dueling DQN lies in decoupling the Q-value function into a linear combination of a state value function $V(s)$ and an action advantage function $A(s, a)$:

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|\mathcal{A}|}\sum_{a'} A(s, a')\right) \tag{22}$$

This decomposition carries profound physical implications: $V(s)$ characterizes the inherent communication potential of

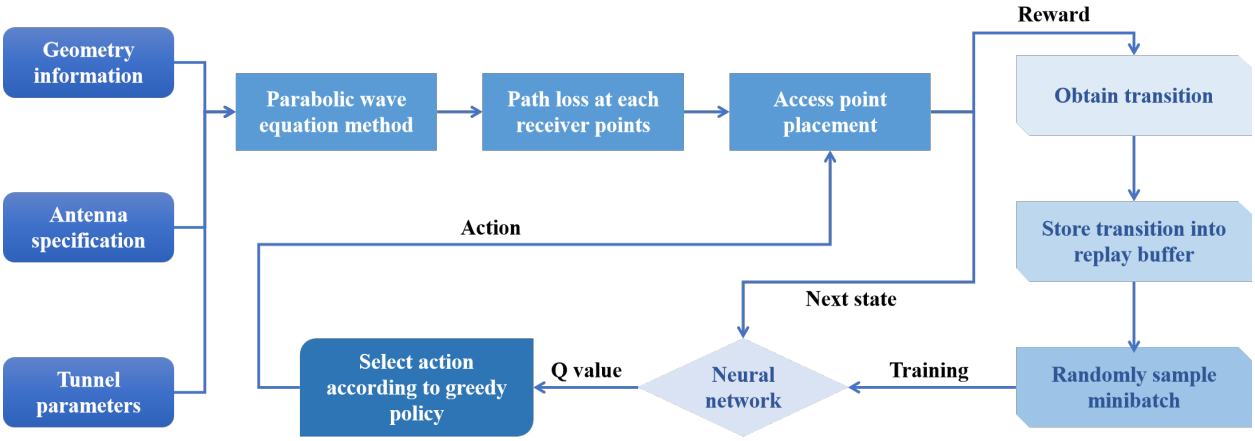

Fig. 4: Framework of the proposed approach of AP placement for CBTC systems.

Optimization Methods	Cost Function Value for Initial AP Positions	Optimized Cost Function Values	Percentage Improvement	Final AP Positions
HJ Optimization	276.83	90.02	67.5%	$\mathbf{X} = [0, 483, 979]$
Traditional DQN		83.27	69.9%	$\mathbf{X} = [0, 238, 714]$
Dueling DQN		54.24	80.4%	$\mathbf{X} = [0, 155, 658]$

TABLE II: Optimization results comparison.

Weight Parameter α	Optimized AP Positions	Optimized Cost Function Value	Unoptimized Cost Function Value	Percentage Improvement
0	$\mathbf{X} = [0, 130, 626]$	80.23	526.51	84.8%
0.5	$\mathbf{X} = [0, 155, 658]$	54.24	276.83	80.4%
1	$\mathbf{X} = [1, 192, 713]$	19.70	27.15	27.4%

TABLE III: Parameter adjustment comparison.

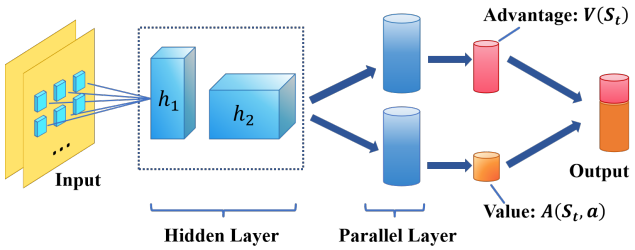

Fig. 5: Diagram of the dueling DQN for the AP optimization problem.

the current AP layout in the tunnel, while $A(s, a)$ reflects the relative improvement from specific movement commands. For example, when three APs are nearly equidistant (where $V(s)$ peaks), any action that disrupts this balance generates negative advantages in $A(s, a)$, even if such actions temporarily enhance local signal strength. This decoupling enables for clearer identification of critical state features.

The network architecture of dueling DQN achieves refined modeling of AP deployment strategies by decoupling state value estimation and action advantage learning. As shown in Figure 5, the network consists of a shared feature extraction layer, parallel value advantage evaluation layers, and a normalized output layer. Its design strictly follows the progressive principle of "feature sharing, value decoupling, and advantage normalization." The input layer receives a 4-dimensional state vector $s_t = [x_1, x_2, x_3, g_t]$, where $x_i \in [0, 1500]$ represents the AP location coordinates, and g_t is the current cost function. The forward propagation process begins with nonlinear feature abstraction through two fully connected layers:

$$h_1 = \text{ReLU}(W_1 s_t + b_1)$$
$$h_2 = \text{ReLU}(W_2 h_1 + b_2)$$

Here, $W_1 \in \mathbb{R}^{64\times4}$ and $W_2 \in \mathbb{R}^{64\times64}$ are shared weight matrices. The ReLU activation function ensures that the network captures non-linear spatial relationships between AP locations, such as the compound effects between adjacent APs on signal attenuation. The 64-dimensional vector h_2 from the shared feature layer is then decoupled into two parallel branches: the value stream $V(s_t)$, which evaluates the inherent quality of the current AP layout, and the advantage stream $A(s_t, a)$, which quantifies the improvement potential of each action relative to the average. This decoupling mechanism allows the network to independently learn two critical types of information—the value stream focuses on identifying high-quality layout patterns aligned with communication engineering principles (e.g., equidistant triangular distributions), while the advantage stream refines the subtle utility differences among 27 possible actions.

The output layer generates the final Q-value through an aggregation formula demonstrated in Equation 22. The core

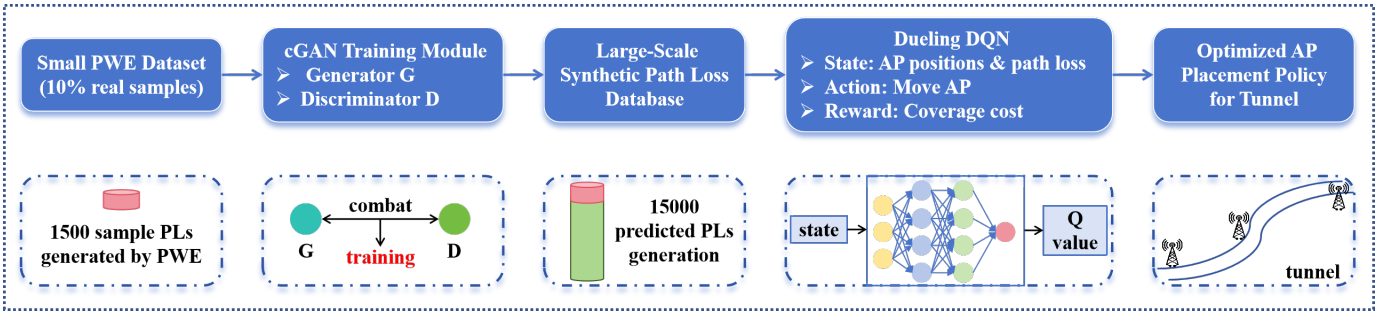

Fig. 6: Overall pipeline of cGAN-augmented DRL for AP placement: (i) limited PWE simulations provide high-fidelity PL maps, (ii) cGAN learns to map coarse PL inputs (and AP mask) to fine-resolution PL, (iii) the generated database fuels a Dueling DQN agent for placement optimization.

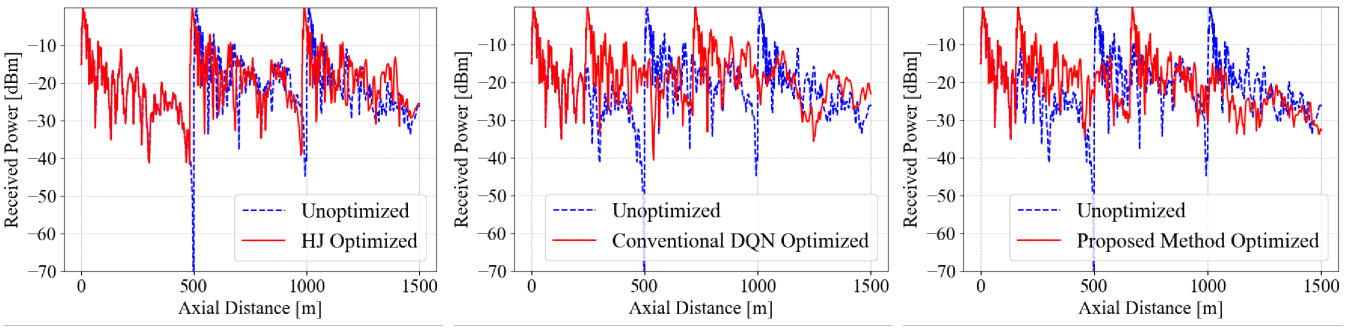

Fig. 7: Optimization results of three different methods for AP placement in tunnels: (a) Hooke and Jeeves (HJ), (b) DQN, and (c) Dueling DQN.

innovation lies in the normalization of the advantage term. By subtracting the mean of action advantages $\frac{1}{|\mathcal{A}|}\sum_{a'} A(s_t, a')$, the network enforces the constraint:

$$\sum_{a'} [Q(s_t, a) - V(s_t)] = 0 \tag{23}$$

This ensures strict alignment between the Q-value baseline and the inherent state value. In problem practice, this constraint effectively suppresses the oscillation of Q values observed in traditional DQN during the fine-tuning of actions: When the AP positions approach optimality, the advantage values of actions converge and the Q values are governed primarily by $V(s_t)$. This prevents policy updates from deviating from stable regions due to noise perturbations.

Dueling DQN enhances AP placement optimization by decoupling the Q-value into a state value $V(s)$ and action advantages $A(s, a)$. This separation allows $V(s)$ to holistically evaluate the intrinsic quality of AP configurations (e.g., uniform spacing), while $A(s, a)$ isolates the relative impact of individual movements. Crucially, normalizing advantages by subtracting their mean $(A(s, a) - \mathbb{E}[A(s, a)])$ stabilizes training by anchoring Q-values to $V(s)$, suppressing noise during fine-tuning phases. In sparse-reward scenarios (for example, near-optimal g), $V(s)$ maintains gradient signals to guide exploration, while $A(s, a)$ amplifies subtle action differences. The architecture thus achieves superior policy precision and faster convergence compared to standard DQN, particularly in identifying spatially optimal AP layouts under complex signal propagation constraints. Algorithm 1 provides detailed procedure of our proposed method for AP placement.

V. cGAN-based Path Loss Data Augmentation

To alleviate the prohibitive computational cost of running parabolic wave equation (PWE) simulations for every possible access point (AP) position, we introduce a conditional Generative Adversarial Network (cGAN) as a data augmentation module. The cGAN is trained on a limited set of high-fidelity PWE-generated path loss (PL) maps and learns to synthesize realistic PL distributions for unseen AP positions. The overall workflow of the cGAN-augmented optimization framework is illustrated in Figure 6.

A. Principle of cGAN

The cGAN consists of a generator G and a discriminator D, both conditioned on auxiliary information describing the AP position and tunnel geometry. Let $\mathbf{x} \in \mathbb{R}^{n_x}$ denote a coarse path loss (PL) profile obtained from sparse PWE simulations or low-resolution field measurements, and let $\mathbf{c} \in \mathbb{R}^{n_c}$ represent a condition vector encoding the AP position, tunnel geometry parameters, and other scenario-specific metadata.

The generator learns a conditional mapping from $(\mathbf{x}, \mathbf{c})$ to a high-resolution PL prediction $\hat{\mathbf{y}} \in \mathbb{R}^{n_y}$:

$$G : (\mathbf{x}, \mathbf{c}) \longrightarrow \hat{\mathbf{y}}, \tag{24}$$

where $\hat{\mathbf{y}}$ denotes the generated full-resolution PL map covering the valid forward region of the tunnel.

The discriminator D receives either the real PWE-generated PL map $\mathbf{y}$ or the generated output $\hat{\mathbf{y}}$, along with $(\mathbf{x}, \mathbf{c})$, and estimates the probability of the map being real. The adversarial training objective is:

$$\begin{aligned}\mathcal{L}_{\text{cGAN}}(G, D) = &\mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{c}}\big[\log D(\mathbf{x}, \mathbf{y}, \mathbf{c})\big] \\ &+ \mathbb{E}_{\mathbf{x},\mathbf{c}}\big[\log\big(1 - D(\mathbf{x}, G(\mathbf{x}, \mathbf{c}), \mathbf{c})\big)\big] \\ &+ \lambda\, \mathcal{L}_{\text{rec}}(G),\end{aligned} \tag{25}$$

where $\mathcal{L}_{\text{rec}}(G)$ is the reconstruction loss:

$$\mathcal{L}_{\text{rec}}(G) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{c}}\left[\|\mathbf{y} - G(\mathbf{x}, \mathbf{c})\|_1\right], \tag{26}$$

and λ balances perceptual realism and numerical accuracy.

In our implementation, G adopts an encoder-decoder architecture with residual connections to capture both large-scale attenuation patterns and fine-grained fading details, while D employs a PatchGAN structure that enforces local realism. Conditioning is injected at multiple layers to ensure physical consistency with the tunnel environment and AP placement.

B. Role in AP Placement Optimization

Within the AP placement optimization framework, the cGAN serves as a physics-aware surrogate model for the PWE simulator. Once trained, it can generate high-resolution PL maps for any candidate AP position almost instantly, eliminating the need to re-run costly PWE simulations. This capability significantly expands the pool of available PL data, enabling the optimizer to evaluate a much larger number of placement configurations under the same computational budget.

By learning the statistical and physical characteristics of PWE-generated data, the cGAN preserves key propagation features such as waveguide effects, path-dependent attenuation, and localized fading not captured by empirical models. Consequently, the optimizer can make placement decisions based on realistic coverage predictions, while drastically reducing simulation time. This integration effectively bridges the gap between computational electromagnetics and large-scale optimization, providing a scalable and accurate approach to AP deployment in tunnel-based CBTC systems.

VI. Results and Performance Evaluation

In this section, we evaluate the effectiveness of our proposed approach by optimizing the positions of three APs in a 1500m railway tunnel. Figure 8 illustrates the geometry of the test tunnel. The tunnel curves right in the first half and left in the second half. The radius of curvature for both is 477.5 m. The electrical parameters of the tunnel walls are $\epsilon_r = 5$ and $\sigma_0 = 0.01 S/m$.

To align with the environmental model described in Section III, we specify the height of the AP (defined as the z axis)

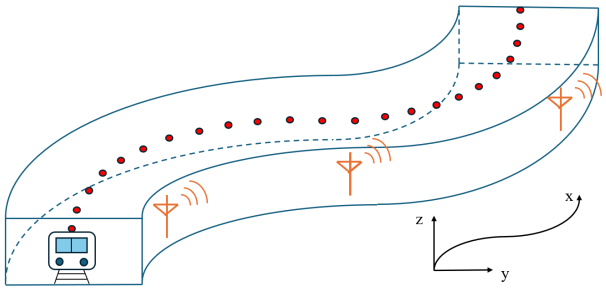

Fig. 8: The geometry of the test tunnel.

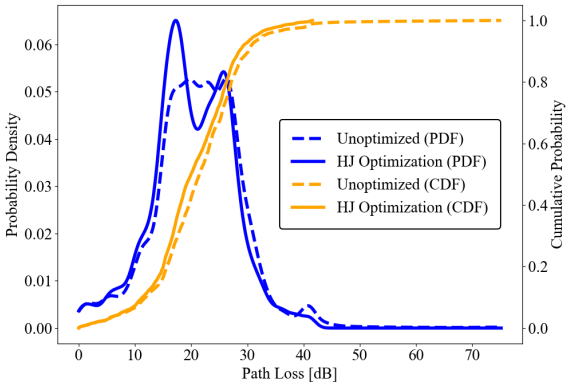

Fig. 9: PDF and CDF of HJ method.

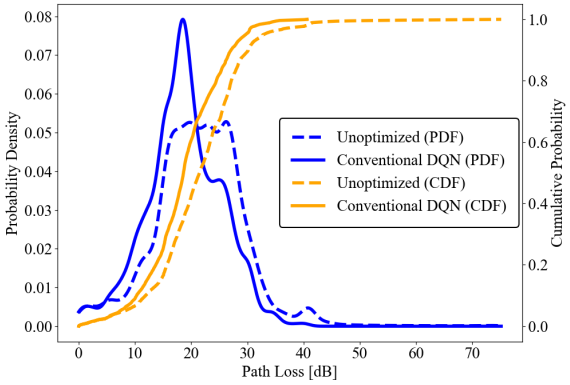

Fig. 10: PDF and CDF of DQN method.

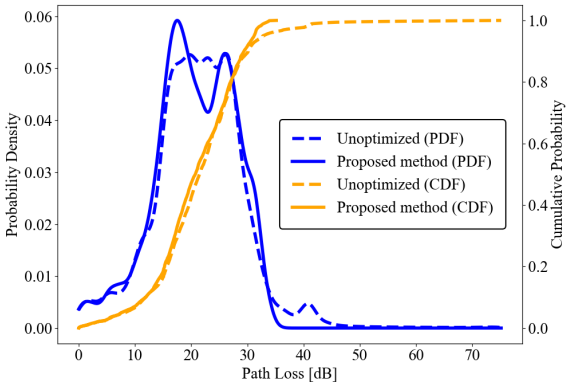

Fig. 11: PDF and CDF of Dueling DQN method.

Algorithm 1 Dueling DQN for AP Deployment Optimization in Tunnels

Initialize: Tunnel length $L = 1500$, AP initial positions $\mathbf{X}_0 = [0, 500, 1000]$, step size Δ, state $s_t = [x_1^{(t)}, x_2^{(t)}, x_3^{(t)}, g^{(t)}]$
Initialize: Dueling DQN parameters: shared layers $W_1 \in \mathbb{R}^{64\times4}$, $W_2 \in \mathbb{R}^{64\times64}$, value stream $V(s;\theta_v)$, advantage stream $A(s,a;\theta_a)$
Initialize: Replay memory D (capacity $C = 2000$), batch size $B = 64$, $\gamma = 0.995$, $\epsilon = 1.0$, $\epsilon_{\text{decay}} = 0.995$, $\epsilon_{\min} = 0.01$
Initialize: Dueling DQN network with coefficients θ, target network $\theta^- \leftarrow \theta$
for episode $= 1$ **to** N_{episodes} **do**
  Reset environment: $\mathbf{X} \leftarrow \mathbf{X}_0$, compute $g_0 = f(\mathbf{X})$, state $s_0 \leftarrow [\mathbf{X}, g_0]$
  while $g^{(t)} \geq 20$ **and** step $< \overline{N}_{\text{step}}$ **do**
    Generate $r \sim \mathcal{U}(0,1)$
    if $r < \epsilon$ **then**
      Sample action index $a_{\text{idx}} \sim \mathcal{U}(0, 26)$
    else
      Compute Q-values via Dueling DQN: $Q(s_t,a) = V(s_t;\theta_v) + (A(s_t,a;\theta_a) - \frac{1}{27}\sum_{a'} A(s_t,a';\theta_a))$
    end if
    Decode $a_{\text{idx}} = \underset{a}{\text{argmax}}\ Q(s_t,a)$
    Decode action vector: $(a_1,a_2,a_3) = \phi^{-1}(a_{\text{idx}})$
    Update AP positions: $x_i^{(t+1)} = \text{clip}\left(x_i^{(t)} + a_i \cdot \Delta,\ 0,\ L\right) \forall i = 1,2,3$
    Compute $g^{(t+1)} = f(\mathbf{X}^{(t+1)})$, reward $R = -g^{(t+1)}$
    Store transition $(s_t, a_{\text{idx}}, R, s_{t+1})$ in D
    if $|D| \geq B$ **then**
      Sample minibatch (s_j, a_j, R_j, s_{j+1})
      Compute TD targets:
      if $g^{(j+1)} < 20$ **then**
        $y_j = R_j$
      else
        $y_j = R_j + \gamma \cdot \max_{a'} \left[Q(s_{j+1}, a'; \theta^-)\right]$
      end if
      Update θ via gradient descent on $\frac{1}{B}\sum(y_j - Q(s_j,a_j;\theta))^2$
    end if
    if $\text{mod}(step, 100) == 0$ **then**
      Update target network: $\theta^- \leftarrow \theta$
    end if
    Decay $\epsilon \leftarrow \max(\epsilon \cdot \epsilon_{\text{decay}}, \epsilon_{\min})$
    $t \leftarrow t + 1$
  end while
end for

and the lateral distance from the tunnel wall (defined as the y axis) as 3 meters and 0.5 meters, respectively, thus restricting AP movement to the tunnel axis designated as the x axis. In addition, since the receivers employed on the train have a fixed height and distance from the tunnel wall as well, all receivers move on a one-dimensional curve along the x-axis. In that case, we only consider the path loss values on the curve of receivers and set APs on the one-dimensional curve along the x-axis, 3m high, and 0.5m from the tunnel wall. We discretize the curve where the receivers are located at intervals of 0.1 meters. Then, we use the PWE method to generate the path loss values of each discrete point when an AP is at different positions. Using the function $\text{PL}_i(\mathbf{X})$ mentioned in Section II, the path loss values can be calculated for scenarios with three APs deployed in the tunnel. For APs, we use a vertically polarized horn antenna with 7 dBi gain and 2.4 GHz operating frequency. For receivers, we use vertically polarized dipole-receiving antennas. The APs are placed at integer-meter positions on the curve where they are located. This means that during the training of deep reinforcement learning, the action step of an AP is also set to 1 meter. The three APs are initially placed at uniformly spaced intervals of 0m, 500m, and 1000m along the x-axis.

Additionally, a series of parameters of the functions in section II need to be set. The th_i defined in Equation 5 is assigned a value of 30 to account for the path loss values generated by PWE. The λ_i defined in Equation 6 and Equation 7 is set to 10 to increase the intensity of penalties. The α defined in Equation 8 is set at 0.5 to balance the impact of f_1 and f_2. In addition, we adjust α to 0 and 1 to optimize when the cost function is only influenced by f_1 or f_2. Furthermore, the parameters of the DRL algorithm must be specified. Table I illustrates the significant parameters.

A. Performance Comparison of HJ, DQN, and Dueling DQN

Table II illustrates the optimization results of the three algorithms. The final AP placements and the final cost function have been listed. The cost function has been defined as Equation 8 in Section II. The final AP positions in Table II show only the x-axis due to the fixed height configurations and the tunnel wall distance. In addition, the table shows the enhancements for the three methods compared to the initial AP positions. We can find that all of them successfully optimize the AP positions, since the cost function is reduced by them, but the cost function is maximally optimized by the Dueling DQN method while traditional DQN methods rank second. The HJ method performs the worst. Compared to HJ optimization, the traditional DQN method achieves a 7.5% improvement. Meanwhile, the dueling DQN algorithm outperforms the traditional DQN with a 34.9% improvement. Figure 7 shows the receive power of each receiver points under three distinguished methods. Analysis shows that the received power generally increases after optimization. In Figure 9-Figure 11, the probability density functions (PDF) and cumulative distribution functions (CDF) of path loss before and after optimization of the three methods comparison are shown. We can observe that all the statistical distribution profiles of the path loss move to the left after optimizations, which means that the path loss values are generally reduced.

B. Results Comparison of Cost Function Weight Adjustments in Dueling DQN Algorithm

We change the weight parameter α in the cost function defined as Equation 8 to find its impact on the Dueling DQN optimization results. The results are displayed in Table III. We

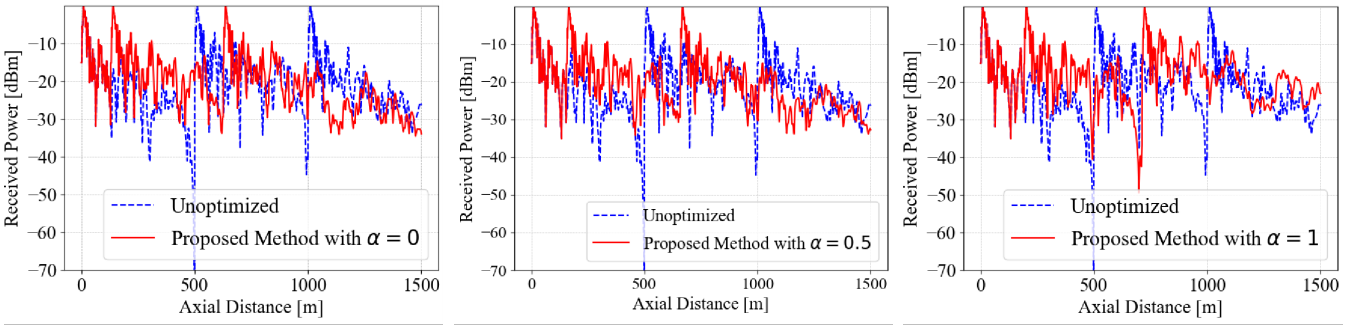

Fig. 12: Optimization results of the proposed Dueling DQN method under different values of α: (a) $\alpha = 0$; (b) $\alpha = 0.5$; (c) $\alpha = 1$.

also present Figure 12 to demonstrate the receive power values at each receiver point in different α. When α is set to 0, the optimization focuses more on the receiver points with high path loss values. When α is set to 1, the optimization only considers the impact of the average path loss. When α is set to 0.5, both the average path loss and the point with the worst path loss are considered. In that case, the weight parameter α can be chosen based on different user requirements.

C. Performance of cGAN-based Path Loss Generation

To evaluate the quality of the generated path loss (PL) maps, we use two standard error metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE). Both are computed over the physically valid, forward-direction region of the tunnel (after the AP location), defined by the binary mask m:

$$\text{MSE} = \frac{1}{N} \sum_i m_i (y_i - \hat{y}_i)^2 \tag{27}$$

$$\text{MAE} = \frac{1}{N} \sum_i m_i |y_i - \hat{y}_i| \tag{28}$$

where y_i is the PWE-simulated reference path loss and $\hat{y}_i$ is the cGAN-generated prediction.

Table IV summarizes the aggregated error metrics over all AP positions in the tunnel. The results demonstrate that the proposed cGAN model achieves low mean and 90th-percentile errors across the entire tunnel length.

TABLE IV: Overall Error Metrics (Grouped by Meter)

Metric	Mean ± Std	90th Percentile
MSE (dB^2)	0.08080 ± 2.74177	0.02325
MAE (dB)	0.05676 ± 0.26662	0.09459

These values indicate that even when generating high-resolution PL maps for unseen AP positions, the model maintains low average prediction error. The 90th-percentile results further confirm the reliability of the model under most deployment scenarios.

Figure 13 and Figure 14 illustrate qualitative results for APs deployed at 100 m and 500 m, respectively. Each plot compares the PWE-simulated reference PL curve with the cGAN-generated prediction. In both cases, the cGAN successfully captures the complex fading patterns and overall signal attenuation trends, closely matching the reference data in the valid forward region.

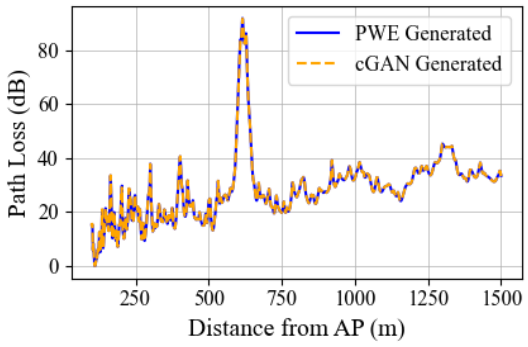

Fig. 13: Path loss at AP position 100 m: PWE-simulated vs cGAN-generated.

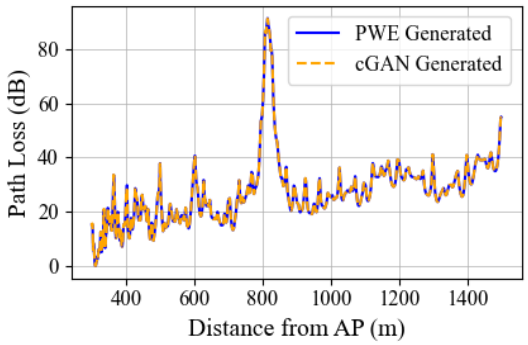

Fig. 14: Path loss at AP position 500 m: PWE-simulated vs cGAN-generated.

Figure 15 shows the distribution of MSE and MAE as a function of AP position along the tunnel. Each point represents the grouped error (by meter), providing insight into how prediction quality varies with AP location. Errors generally

remain low for most AP positions, with a slight increase toward the far end of the tunnel. This is likely due to reduced training data density in those regions, highlighting potential benefits from targeted data augmentation.

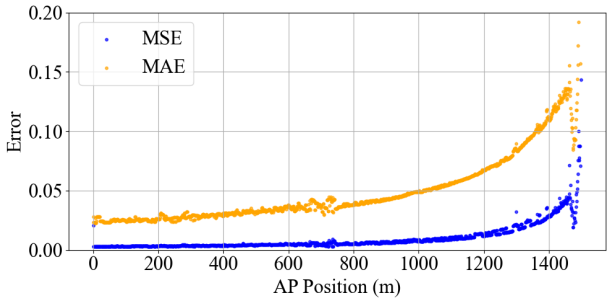

Fig. 15: Scatter plot of MSE and MAE vs AP position along the tunnel.

Overall, the cGAN model demonstrates strong capability in reproducing high-resolution PL distributions from limited PWE simulation data. Its ability to preserve fine-scale fading structures and attenuation trends ensures that the generated PL maps can serve as reliable surrogates for computationally expensive full-wave simulations in subsequent optimization tasks.

VII. Conclusion

This paper proposes a novel physics-driven and data-augmented framework for optimal access point deployment in railway tunnels within the communication-based train control system. Unlike existing AP placement optimization methods, which often rely solely on heuristic search or purely data-driven models and thus suffer from local optima or limited generalization, the proposed approach integrates high-fidelity channel modeling using the parabolic wave equation with a conditional generative adversarial network for data augmentation, and a dueling deep Q-network for policy learning.

In the proposed framework, a limited number of PWE simulations are first conducted to generate accurate path loss maps under different AP placements. These high-fidelity samples are then used to train the cGAN, which learns to synthesize realistic PL maps for untested AP positions, substantially reducing the computational cost of exhaustive electromagnetic simulations. The augmented PL dataset is subsequently used to define a CBTC-specific reinforcement learning environment, enabling the Dueling DQN agent to explore and optimize AP configurations efficiently. Comparative experiments with Hooke and Jeeves optimization and standard DQN confirm that our method achieves superior average and worst-case received power performance, with the additional benefit of reduced simulation requirements.

The framework also supports flexible optimization objectives by tuning the parameter α in the cost function: a higher α emphasizes minimizing the average PL, whereas a lower value prioritizes worst-case PL improvement. This adaptability allows the system to be customized for different CBTC operational requirements while maintaining superior performance over baseline methods.

Future research will focus on integrating more advanced deep reinforcement learning algorithms, such as rainbow DQN, which combines enhancements including double DQN, Prioritized Experience Replay, Noisy Nets, and categorical DQN. These techniques could further improve exploration-exploitation balance, adapt policies dynamically to spatially varying tunnel conditions, and accelerate convergence by prioritizing high-impact samples (e.g., from severe attenuation zones). Such improvements would make the proposed hybrid framework even more scalable, robust, and deployment-ready for complex tunnel environments.

References

[1] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Design and performance enhancements in communication-based train control systems with coordinated multipoint transmission and reception," *EEE Trans. Intell. Transp. Syst*, vol. 15, no. 3, pp. 1258–1272, 2014.

[2] H. Qin and X. Zhang, "Physics-based wave propagation model assisted vehicle localisation in tunnels," *IET Microw. Antennas Propag.*, vol. 17, no. 10, pp. 786–796, 2023.

[3] H. Song, W. Wu, H. Dong, and E. Schnieder, "Propagation and safety analysis of the train-to-train communication system," *IET Microw. Antennas Propag.*, vol. 13, no. 13, pp. 2324–2329, 2019.

[4] H. Wang, F. R. Yu, L. Zhu, T. Tang, and B. Ning, "A cognitive control approach to communication-based train control systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1676–1689, 2015.

[5] H. Qin, Z. Wu, Y. Liu, X. Zhang, and X. Zhang, "Physics-based trajectory design for cellular-connected uav in rainy environments based on deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, pp. 1–16, 2025.

[6] T. Wen, C. Constantinou, L. Chen, Z. Tian, and C. Roberts, "Access point deployment optimization in cbtc data communication system," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1985–1995, 2018.

[7] M. Saki, M. Abolhasan, J. Lipman, and A. Jamalipour, "A comprehensive access point placement for iot data transmission through train-wayside communications in multi-environment based rail networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11 937–11 949, 2020.

[8] H. Qin and X. Zhang, "Comparative analysis of finite-difference and split-step based parabolic equation methods for tunnel propagation modelling," *IET Microw. Antennas Propag.*, vol. 18, no. 2, pp. 59–72, 2024.

[9] X. Guan, L. Guo, and Q. Li, "A vector parabolic equation method for propagation predictions over flat terrains," in *2016 11th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, 2016, pp. 365–368.

[10] H. Qin, X. Zhang, W. Hou, and X. Zhang, "Fast parabolic wave equation-based time-of-arrival estimation exploiting sparse matrices," *IEEE Antennas and Wireless Propag. Lett.*, vol. 24, no. 3, pp. 661–665, 2025.

[11] Z. Zhao, H. Qin, J. Wang, W. Hou, and X. Zhang, "Embedding antennas with tilted beam patterns into parabolic wave equation-based models for tunnel propagation," *IET Microw. Antennas Propag.*, vol. 17, no. 9, pp. 685–693, 2023.

[12] O. Ozgun, G. Apaydin, M. Kuzuoglu, and L. Sevgi, "PETOOL: MATLAB-based one-way and two-way split-step parabolic equation tool for radiowave propagation over variable terrain," *Comput. Phys. Commun.*, vol. 182, no. 12, pp. 2638–2654, 2011.

[13] H. Qin and X. Zhang, "Efficient radio wave propagation modeling in tunnels with a sparse fourier transform-based split-step parabolic equation method," *IEEE Antennas Wireless Propag. Lett.*, vol. 22, no. 10, pp. 2442–2446, 2023.

[14] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[15] F. AlMahamid and K. Grolinger, "Reinforcement learning algorithms: An overview and classification," in *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, Sep. 2021, pp. 1–7.

[16] F. Jiao and H. Liang, "A 5G enabled next generation train control data communication system based on train-to-train communication," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 2022, pp. 3064–3068.

[17] X. Zhang, A. Ludwig, N. Sood, and C. D. Sarris, "Physics-based optimization of access point placement for train communication systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 3028–3038, 2018.
[18] H. Sherali, C. Pendyala, and T. Rappaport, "Optimal location of transmitters for micro-cellular radio communication system design," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 4, pp. 662–673, 1996.
[19] F. Aguado-Agelet, A. Varela, L. Alvarez-Vazquez, J. Hernando, and A. Formella, "Optimization methods for optimal transmitter locations in a mobile wireless system," *IEEE Trans. Veh. Technol.*, vol. 51, no. 6, pp. 1316–1321, 2002.
[20] L. Zhou, J. Zhang, J. Zhang, and K. Qiu, "An environment-adaptive radio propagation path loss model with ray-based validation," *IEEE Antennas Wireless Propag. Lett.*, vol. 23, no. 10, pp. 3217–3221, 2024.
[21] B. Myagmardulam, N. Tadachika, K. Takahashi, R. Miura, F. Ono, T. Kagawa, L. Shan, and F. Kojima, "Path loss prediction model development in a mountainous forest environment," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 2494–2501, 2021.
[22] S. I. Popoola, A. A. Atayero, and O. A. Popoola, "Comparative assessment of data obtained using empirical models for path loss predictions in a university campus environment," *Data Brief*, vol. 18, pp. 380–393, 2018.
[23] J. Li, X. Xu, H. Cao, Q. Ren, and K. Wang, "Large-area complex terrain radio wave propagation loss prediction using bidirectional three-dimensional parabolic equations," in *2024 14th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*, 2024, pp. 1–4.
[24] N. Noori, S. Safavi-Naeini, and H. Oraizi, "A new three-dimensional vector parabolic equation approach for modeling radio wave propagation in tunnels," in *2005 IEEE Antennas and Propagation Society International Symposium*, vol. 4B, 2005, pp. 314–317 vol. 4B.
[25] S. Sun, T. S. Rappaport, T. A. Thomas, A. Ghosh, H. C. Nguyen, I. Z. Kovacs, I. Rodriguez, O. Koymen, and A. Partyka, "Investigation of prediction accuracy, sensitivity, and parameter stability of large-scale propagation path loss models for 5g wireless communications," *IEEE transactions on vehicular technology*, vol. 65, no. 5, pp. 2843–2860, 2016.
[26] Z. Qiang, Y. Lixia, G. Xiangming, and L. Qingliang, "Prediction of electromagnetic wave propagation loss in evaporation duct based on 3d vector parabolic equation," in *2021 13th International Symposium on Antennas, Propagation and EM Theory (ISAPE)*. IEEE, 2021, pp. 1–3.
[27] L. M. M. Saravia, A. R. M. Saravia, M. M. Silva, and L. H. Gonsioroski, "Comparing exponential decay models for path loss prediction at 5.8 ghz using machine learning techniques," in *2024 IEEE 1st Latin American Conference on Antennas and Propagation (LACAP)*. IEEE, 2024, pp. 1–2.
[28] I. F. M. Rafie, S. Y. Lim, and M. J. H. Chung, "Path loss prediction in urban areas using convolutional neural networks," in *2022 IEEE International RF and Microwave Conference (RFM)*. IEEE, 2022, pp. 1–4.
[29] A. Marey, M. Bal, H. F. Ates, and B. K. Gunturk, "PL-GAN: Path loss prediction using generative adversarial networks," *IEEE Access*, vol. 10, pp. 90 474–90 480, 2022.
[30] D. K. Vishwakarma *et al.*, "Comparative analysis of deep convolutional generative adversarial network and conditional generative adversarial network using hand written digits," in *2020 4th international conference on intelligent computing and control systems (ICICCS)*. IEEE, 2020, pp. 1072–1075.
[31] S. Zhang, A. Wijesinghe, and Z. Ding, "RME-GAN: A learning framework for radio map estimation based on conditional generative adversarial network," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18 016–18 027, 2023.
[32] C. T. Cisse, O. Baala, V. Guillet, F. Spies, and A. Caminada, "Irgan: cgan-based indoor radio map prediction," in *2023 IFIP Networking Conference (IFIP Networking)*. IEEE, 2023, pp. 1–9.
[33] A. E. Forooshani, A. A. Lotfi-Neyestanak, and D. G. Michelson, "Optimization of antenna placement in distributed mimo systems for underground mines," *IEEE Trans. Wireless Commun.*, vol. 13, no. 9, pp. 4685–4692, 2014.
[34] K. Maksuriwong, V. Varavithya, and N. Chaiyaratana, "Wireless lan access point placement using a multi-objective genetic algorithm," in *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, vol. 2, 2003, pp. 1944–1949 vol.2.
[35] A. Seretis and C. D. Sarris, "An overview of machine learning techniques for radiowave propagation modeling," *IEEE Trans. Antennas Propag.*, vol. 70, no. 6, pp. 3970–3985, 2022.
[36] S. Huang, H. Qin, W. Hou, X. Zhang, and X. Zhang, "A generalizable physics-guided convolutional neural network for irregular terrain propagation," *IEEE Trans. Antennas Propag.*, vol. 73, no. 6, pp. 3975–3985, 2025.
[37] S. Huang, H. Qin, and X. Zhang, "Efficient parabolic equation-driven CNN propagation model in tunnels based on frequency conversion," *IEEE Trans. Antennas Propag.*, vol. 72, no. 7, pp. 6024–6031, 2024.
[38] H. Qin, S. Huang, and X. Zhang, "A high-accuracy deep back-projection CNN-based propagation model for tunnels," *IEEE Antennas and Wireless Propag. Lett.*, vol. 23, no. 3, pp. 1015–1019, 2023.
[39] D. Zhou, H. Chen, and G. Cheng, "A security containers placement algorithm based on dqn for microservices to defend against co-resident threat," in *2023 8th International Conference on Computer and Communication Systems (ICCCS)*, 2023, pp. 683–688.
[40] J. Liu, M. Song, H. Shu, W. Peng, L. Wei, and K. Wang, "A novel FBG placement optimization method for tunnel monitoring based on woa and deep q-network," *Symmetry*, vol. 16, no. 10, 2024.
[41] J. Kim, J. Lee, T. Kim, and S. Pack, "Deep Q-network-based cloud-native network function placement in edge cloud-enabled non-public networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 2, pp. 1804–1816, 2023.