Received 13 May 2026; revised 26 June 2026; accepted 21 July 2026. Date of publication 24 July 2026; date of current version 3 August 2026.

Digital Object Identifier 10.1109/OJCOMS.2026.3716850

RIS-Enabled Wireless Channel Equalization: Adaptive RIS Equalizer and Deep Reinforcement Learning

GAL BEN-ITZHAK (Graduate Student Member, IEEE) AND ENDER AYANOGLU (Fellow, IEEE)

Department of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA 92697 USA

CORRESPONDING AUTHOR: E. AYANOGLU (ayanoglu@uci.edu)

This work was supported in part by NSF under Grant 2030029.

ABSTRACT Reconfigurable Intelligent Surfaces (RISs) offer a promising means of reshaping the wireless propagation environment, yet practical methods for configuring large passive arrays to achieve reliable signal equalization remain limited. Equalization is essential in wideband links to counteract multipath-induced pulse distortion that otherwise degrades symbol recovery. This work investigates RIS-assisted pulse response equalization and signal boosting using both classical adaptive filtering and model-free deep reinforcement learning (DRL). We develop a steepest descent (SD) method that exploits cascaded base station-RIS-user equipment (UE) channel information to configure RIS coefficients for multipath mitigation and SNR enhancement, and we show that the tradeoffs between SD and DRL primarily arise from the extensive channel estimation required for accurate equalization with passive RIS hardware. Unlike traditional adaptive filtering, which updates delayed filter coefficients after signal reception, our approach uses the RIS positioned within the cascaded channel to perform equalization without delay elements, prior to reception at the UE. In this framework, the channel is estimated before equalization, forming the basis of what we term adaptive RIS equalization (ARISE). To overcome the reliance on channel estimation required for ARISE, we explore several DRL algorithms–DDPG, TD3, and SAC–that optimize RIS coefficients directly from the received pulse response without explicit channel estimation. Through extensive simulations across diverse channel conditions and RIS sizes, we show that SAC achieves fast, stable convergence and equalization performance comparable to ARISE while offering significantly lower implementation complexity. These results highlight the potential of DRL as a practical and scalable solution for real-time RIS control in future wireless systems.

INDEX TERMS Reconfigurable intelligent surface (RIS), steepest descent (SD), deep reinforcement learning (DRL), wireless channel equalization, deep deterministic policy gradient (DDPG), twin-delayed DDPG (TD3), soft actor critic (SAC).

I. INTRODUCTION

RECONFIGURABLE Intelligent Surfaces (RISs) have emerged as a promising technology for next-generation wireless networks, offering the ability to reshape the propagation environment through nearly passive, low-power hardware [1], [2]. By adjusting the phase of incident signals, RISs can enhance coverage, mitigate interference, and improve link reliability between the base station (BS) and user equipment (UE) without requiring active radio frequency (RF) chains. However, realizing these benefits in practice requires efficient methods for configuring large RIS arrays under realistic channel conditions. A central challenge is achieving reliable signal equalization and boosting without relying on full channel state information (CSI), which is difficult to obtain due to the cascaded BS-RIS-UE channel structure and the high dimensionality of RIS-assisted links [3]. These limitations become even more pronounced in dynamic or richly scattered environments, where rapid adaptation is essential for maintaining link quality [4]. These challenges motivate the need for approaches that can tune RIS configurations for optimal signal integrity without relying on full CSI acquisition or repeated pilot transmissions, particularly in dynamic environments.

A substantial body of work has explored RIS optimization using classical signal processing and model-based techniques, primarily focusing on maximizing the received

power, improving signal-to-noise ratio (SNR), maximizing sum rates, or enabling beamforming and focusing [3], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. While effective in controlled scenarios, these methods often assume access to accurate CSI and face scalability issues as RIS sizes grow and channel estimation overhead becomes prohibitive. Researchers have proposed reduced-complexity estimation schemes, compressive sensing techniques, and analytical optimization frameworks for CSI acquisition, which alleviate some system requirements and overhead but do not eliminate the channel estimation phase [18].

Despite these advances, RIS-based signal equalization remains relatively underexplored. Most existing RIS literature focuses on spatial beamforming or spatial-domain equalization, while the temporal shaping of the received signal–critical for mitigating multipath distortion and ensuring reliable symbol recovery–has received limited attention. Spatial shaping adjusts the angular distribution of multipath components to improve SNR or suppress interference, but it does not directly reduce delay spread or mitigate ISI. Equalization remains essential in high-speed links to mitigate intersymbol interference (ISI) caused by multipath distortion, making RIS-assisted temporal shaping particularly relevant. In wideband channels where the multipath delay spread becomes comparably larger than the symbol duration, conventional receiver-side equalizers become increasingly complex and noise-enhancing. In this work, we consider a symbol duration of T_s = 16.3 ns, where the signal is prone to distortion as practical wideband deployments (e.g., dense urban microcells or indoor industrial environments) often exhibit RMS delay spreads in the range of 50–150 ns [19].

For exponentially decaying power delay profiles, the maximum excess delay is proportional to the RMS delay spread, with the proportionality factor depending on the power threshold used to define the last resolvable path (typically on the order of 5–10$\times$ for thresholds between −20 dB and −40 dB) [20]. Such delay spreads span multiple symbol intervals and require long feed-forward equalizers (FFEs) or decision-feedback equalizers (DFEs) with many taps. Long feedforward filters amplify noise, and long feedback filters increase error-propagation risk, while requiring higher power consumption and processing complexity at the UE.

A few recent works have begun to address this gap. For example, [21] demonstrates that RISs can be configured to suppress or align multipath components prior to reception to minimize ISI. We note that [21] refers to ISI mitigation using RIS as "spatial equalization," but in our work we refer to this concept as "temporal equalization," since ISI reduction and multipath delay distortion fundamentally affects the delay-domain impulse response of the channel. Similarly, [22] shows that RISs can compensate for channel distortion with minimal receiver-side processing, enabling "over-the-air equalization" in constrained environments. However, these studies represent early steps, and the practical constraints of passive RIS hardware introduce additional challenges: effective equalization may require large or finely tuned coefficient sets, and there is an inherent tradeoff between signal gain and equalization accuracy.

In this work, the RIS simultaneously performs **both spatial and temporal equalization**. Spatially, the RIS aligns multipath components to enhance the main tap and increase the received SNR through constructive beamforming. Temporally, the RIS reshapes the delay-domain impulse response by suppressing or realigning delayed paths, thereby reducing channel memory and mitigating ISI. By placing the RIS at an intermediate physical layer node–the midpoint of the BS-RIS-UE wireless link–we directly manipulate the propagation environment to counteract multipath-induced distortion and focus the signal power towards the desired UE direction. By reducing the effective channel memory, our RIS-based over-the-air equalization framework enables the UE to operate with a shorter and less noise-enhancing equalizer. The RIS therefore complements, rather than replaces, digital equalization.

We present a comprehensive study of RIS-assisted signal equalization and boosting using both classical and learning-based methods. We first develop a steepest descent (SD)-based approach that leverages estimated cascaded BS-RIS-UE channels to configure RIS coefficients for pulse response equalization and SNR enhancement, thus achieving both temporal equalization and spatial beamforming using the RIS. The algorithm we develop, although based on SD, differs in implementation from conventional SD; for this reason, we refer to it as adaptive RIS equalization (ARISE). Although effective, ARISE depends on precise cascaded CSI for accurate equalization, which introduces substantial communication overhead and delays before RIS optimization can begin. These challenges, along with the hardware-related limitations of the physical RIS, motivate a comparison between classical adaptive filtering techniques and modern deep reinforcement learning (DRL) approaches for RIS-assisted equalization.

Learning-based methods–particularly DRL–have gained traction as a model-free alternative for RIS-aided spatial-domain optimization [23], [24], [25]. DRL enables RIS configuration directly from observed performance metrics, bypassing the need for explicit CSI and offering robustness in environments where analytical models are unreliable. Some DRL algorithms implemented in the literature include:

- **Deep Deterministic Policy Gradient (DDPG)** [23], [26]–used to maximize channel rates by optimizing the transmitter beamforming matrix and RIS reflection coefficients for multiple user scenarios, while including practical phase shifts;
- **Twin Delayed DDPG (TD3)** [27]–applied to maximize secrecy energy efficiency in uncrewed aerial vehicle (UAV)-assisted RIS systems;
- and **Soft Actor-Critic (SAC)** [24]–used to optimize the channel rate under realistic hardware impairments in RIS-based systems.

These DRL algorithms demonstrate strong potential for real-time deployment due to their scalability and support

for continuous action spaces. Moreover, practical wireless systems inevitably suffer from hardware impairments, such as amplifier nonlinearities, local oscillator phase noise, jitter, and quantization noise at the UE, as well as RIS phase shift errors–all of which can significantly degrade the performance of advanced beamforming architectures. Recent works have analyzed these effects in the context of reconfigurable holographic surfaces and proposed corresponding modeling and mitigation approaches in cell-free and satellite networks [28], [29], [30]. The model-free nature of DRL algorithms allows them to operate effectively even under these physical constraints, along with RIS hardware limitations which may exhibit nonlinearities, coupling effects, or frequency-dependent behaviors that are difficult to capture analytically.

In this work, we utilize three DRL algorithms towards temporal channel equalization using the RIS, specifically DDPG, TD3, and SAC. These algorithms optimize RIS coefficients directly from the received pulse response without requiring channel estimation overhead. While DRL has been applied to RIS beamforming and spatial optimization, its use for temporal equalization introduces a fundamentally different role for learning. In contrast to prior DRL-based RIS methods that optimize spatial gains or coverage, our formulation treats the RIS as a model-free spatial and temporal equalizer that learns to reshape the received pulse response purely through interaction with the environment. This removes the need for cascaded CSI, bypasses the analytical constraints of passive RIS models, and enables adaptation even when the RIS exhibits the hardware-impaired behaviors discussed previously. Thus, the DRL component is not merely an application of existing algorithms to a new cost function–it represents a shift from model-driven to model-free RIS equalization, allowing the RIS to learn temporal shaping policies that classical methods such as ARISE cannot realize without full channel knowledge. To the best of our knowledge, no prior work has investigated model-free temporal equalization using RISs in wideband channels.

Through extensive simulations across diverse channel conditions and RIS sizes, we demonstrate that SAC consistently achieves fast, stable convergence and performance comparable to ARISE, while offering significantly lower implementation complexity. In particular, SAC begins optimizing the RIS immediately from raw pulse response observations, avoiding the CSI acquisition delay inherent to ARISE and enabling faster adaptation in time-varying channels. This work makes the following key contributions:

- **RIS-based equalization and SNR-enhancement framework:** We formulate the problem of RIS-assisted pulse response equalization and signal boosting and analyze its challenges under realistic passive RIS constraints.
- **SD-based RIS optimization:** We derive an SD algorithm that uses cascaded channel estimates to configure RIS coefficients and characterize its performance, complexity, and tradeoffs. Because of its difference with respect to conventional SD, we call this algorithm ARISE.
- **Model-free DRL methods for RIS control:** We develop and evaluate DDPG, TD3, and SAC algorithms that optimize RIS coefficients directly from received signal observations, eliminating the need for channel estimation.
- **Comprehensive comparative analysis:** We compare the ARISE and DRL approaches in terms of convergence behavior, computational complexity, communication complexity, equalization performance, and robustness across channel types and RIS sizes.
- **Identification of SAC as a practical solution:** We show that SAC achieves rapid, stable convergence and performance on par with ARISE, positioning it as a strong candidate for real-time RIS control in future wireless systems.

The paper is organized as follows: Section II presents the system and channel models and defines the optimization problem. Section III derives the ARISE algorithm and provides simulation results to compare with baseline cases. Section IV introduces the three DRL algorithms and discusses their complexities. Section V shows comparative simulation results among the ARISE and DRL algorithms across diverse environmental models.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. SYSTEM MODEL

The RIS-assisted wireless communication network consists of a single-antenna BS, an M-element metasurface-based RIS, and a single-antenna UE. The RIS elements are metallic patches arranged in a single-row uniform linear array (ULA) structure and separated by a distance d_x along the x direction, electrically controlled by varactor diodes with continuous voltage biases to modulate the reflection phases. The reflection coefficient of each RIS element m is given by $\Gamma_m = |\Gamma_m| e^{j\phi_m}$, with phase shift ϕ_m. Assuming RIS element passivity, we set $|\Gamma_m| = 1$ for all $m = 1, 2, \ldots, M$.

In this study, we adopt an idealized RIS model commonly used in the literature, where each element applies a frequency-flat, lossless, and memoryless reflection coefficient. This allows us to isolate and evaluate the fundamental feasibility of RIS-based spatial and temporal equalization before incorporating hardware-specific effects such as element losses, frequency-dependent responses, nonlinearities, and mutual coupling. In our simulations, the signal bandwidth is less than 100 MHz and therefore the signal experiences multipath delay spread while the RIS elements are approximated as frequency-flat. This behavior is consistent among many RIS models from the literature [11], [31], [32], [33], [34], [35].

In the downlink scenario shown in Fig. 1, the UE receives two main rays of signals: the first coming directly from the BS and the second is the reflected signal from the BS through the RIS. All channels are assumed to be far-field and frequency-selective due to multipath delay spread. We

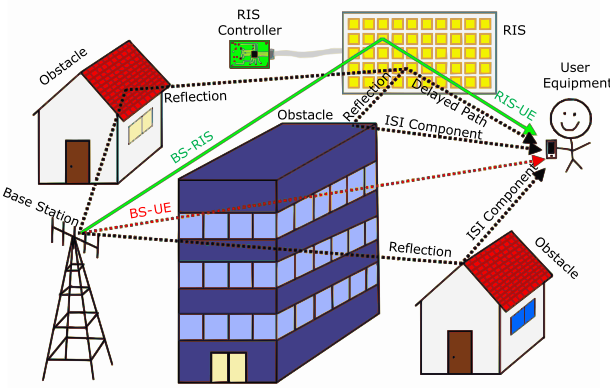


FIGURE 1. Illustration of the downlink RIS scenario. LoS paths are marked by solid lines, while NLoS paths are marked by dotted lines. Delayed paths and ISI components are created by reflections from obstacles.

assume that there is no dominant LoS path between the BS and the UE due to obstacles, thus the channel impulse response is mainly characterized by Rayleigh coefficients

$$h_{\mathrm{BU}}(t) = (\beta_{\mathrm{BU}})^{1/2} h_{\mathrm{NLoS}}(t), \tag{1}$$

where β_{BU} is the path loss related to the distance between the BS and the UE, $\beta_{\mathrm{BU}} = G d_{\mathrm{BU}}^{-\alpha}$. Here, G accounts for transmitter gains, receiver gains, and normalized path losses, d_{BU} is the distance between the BS and the UE, and α is the path loss exponent [21], [36]. The NLoS components are distributed as standard circularly symmetric complex Gaussian random variables $h_{\mathrm{NLoS}}(t) \in \mathcal{CN}(0, \sigma^2(t))$ whose variance $\sigma^2(t)$ depends on the arrival time of the signal at the receiver due to the delayed channel components from multipath effects, with $\sigma^2(0) = 1$ [37].

The channels characterizing the reflection links between the BS and UE through the RIS elements are modeled by the impulse response vector

$$\boldsymbol{h}_{\mathrm{BRU}}(t) = (\beta_{\mathrm{BRU}})^{1/2} \boldsymbol{R}^{1/2} \tilde{\boldsymbol{\Gamma}} \boldsymbol{h}(\theta_{\mathrm{BRU}}, t), \tag{2}$$

where $\beta_{\mathrm{BRU}} = G'(d_{\mathrm{BR}} d_{\mathrm{RU}})^{-\alpha}$ is the path loss related to the distances between the BS and the center of the RIS, and between the center of the RIS and the UE, denoted by d_{BR} and d_{RU}, respectively [38], [39]. We set $\tilde{\boldsymbol{\Gamma}} = \mathrm{diag}[\Gamma_1, \Gamma_2, \ldots, \Gamma_M]$. We define the steering vector by the RIS for the LoS channels as a function of the azimuth angle θ, $\boldsymbol{h}_{\mathrm{LoS}}(\theta) = \left[1, e^{j\omega(\theta)}, \ldots, e^{j(M-1)\omega(\theta)}\right]^T$ with spatial frequencies $\omega(\theta) = 2\pi d_x (f_c/c) \sin(\theta)$, where f_c is the carrier frequency and c is the speed of light [3]. The BS illuminates the RIS with normal incidence and the azimuth angle between the BS, RIS, and UE is θ_{BRU}.

We model the channel fading parameters using Rician coefficients. For a given transmitted symbol, we define $t = 0$ as the time at which the main tap (the first sample with dominant signal power) is received. At this instant, we write

$$\boldsymbol{h}(\theta, t=0) = \sqrt{\frac{\kappa}{\kappa+1}} \boldsymbol{h}_{\mathrm{LoS}}(\theta) + \sqrt{\frac{1}{\kappa+1}} \boldsymbol{h}_{\mathrm{NLoS}}(t=0), \tag{3}$$

where κ, $0 \le \kappa < \infty$, is the Rician factor. For subsequent delayed components, where the LoS contribution is negligible, we assume purely NLoS propagation and model them as

$$\boldsymbol{h}(\theta, t) = \boldsymbol{h}_{\mathrm{NLoS}}(t), \quad t > 0. \tag{4}$$

This reflects a dominant early LoS component followed by scattered multipath. We adopt a quasi-static block-fading model and neglect small-scale common phase variations due to UE micro-mobility, as our focus is on assessing the feasibility of the proposed RIS-based spatial and temporal equalization framework.

Additionally, the channels are correlated by the positive semidefinite spatial correlation matrix $\boldsymbol{R}$ with elements

$$[\boldsymbol{R}]_{n,m} = \mathrm{sinc}\left(\frac{2\|u_n - u_m\|}{c/f_c}\right) \quad n, m = 1, \ldots, M, \tag{5}$$

where u_n and u_m are the positions along x of the RIS elements n and m, respectively, and $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$ is the sinc function [40], [41].

For a transmitted signal $s(t)$, the received baseband signal at the UE is expressed as

$$y(t) = \left(h_{\mathrm{BU}}(t) + \sum_{m=1}^{M} \Gamma_m h_{\mathrm{BRU},m}(t)\right) * s(t) + n(t), \tag{6}$$

where $n(t)$ is additive white Gaussian noise (AWGN) and $*$ is the convolution operator.

B. PROBLEM FORMULATION

Our objective is to configure the RIS so that the received pulse contains a strong, well-aligned main tap with maximal signal power while suppressing the delayed components that generate ISI. Because the RIS is passive and cannot introduce its own temporal delays, unlike traditional equalizers such as the FFE or the DFE, it must rely on the natural multipath structure of the channel: by selecting appropriate phase shifts, the surface can steer and combine the reflected paths in a way that reduces overlap between delayed replicas of the transmitted pulse. The RIS therefore acts simultaneously as a beamformer and as a temporal equalizer, shaping the composite channel to provide the UE a cleaner, more concentrated pulse.

Mathematically, we define the ISI as the signal energy arriving at integer multiples of the symbol period T other than the main tap of the pulse response $y(t)$. Assuming the dominant component of the pulse arrives at the receiver at time $t = 0$, the total ISI is written as

$$I = \sum_{k \neq 0} y(t - kT)|_{t=0}, \tag{7}$$

where the transmitted discrete-time baseband signal is the pulse given by

$$s_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0. \end{cases} \tag{8}$$

The RIS configuration is chosen by optimizing over the reflection coefficient phase-shift vector

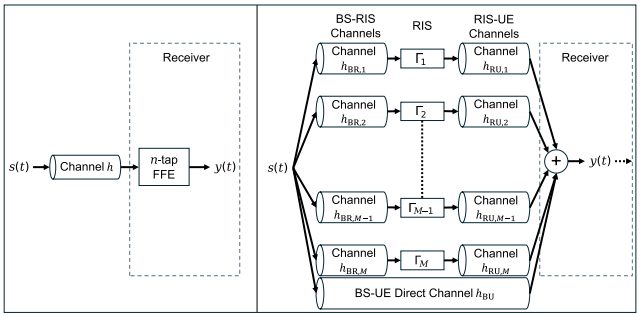


FIGURE 2. Left: conventional feedforward equalizer (FFE) at the receiver vs. right: our proposed RIS-based equalizer. The conventional equalizer processes the baseband signal after reception using n delay taps, whereas the RIS-based equalizer is at the midpoint of the cascaded BS-RIS-UE link and equalizes the signal before reception by using each one of its M reflection coefficients and their corresponding channels.

$\mathbf{\Phi} = [\phi_1, \phi_2, \ldots, \phi_M]^T$. The optimization problem is defined as

$$(P1): \quad \max_{\mathbf{\Phi}} \quad \eta, \tag{9}$$

where

$$\eta = \mathrm{sgn}\big(\Re\{y(0)\}\big) \cdot \big[\Re\{y(0)\}\big]^2 - \sum_{k\neq 0} |y(t-kT)|^2 . \tag{10}$$

The metric η is designed to explicitly reward a strong, correctly phased main tap of the received pulse response while penalizing residual ISI energy. The first term, $\mathrm{sgn}\big(\Re\{y(0)\}\big) \cdot \big[\Re\{y(0)\}\big]^2$, promotes a large real-valued peak at $t = 0$ (consistent with the real-valued transmitted pulse) and enforces phase alignment through the sign factor. Otherwise, the received symbols may be rotated over the real and imaginary axes, as demonstrated in some of the constellations in Figs. 7 and 11. This accounts for the **spatial equalization** task of the RIS. The second term, $-\sum_{k\neq 0} |y(t-kT)|^2$, suppresses the power of all the ISI components and will be minimized for an equalized pulse, thus representing the **temporal equalization** part of the metric.

III. ADAPTIVE RIS EQUALIZATION

A. THE ARISE ALGORITHM

A popular approach to minimize the ISI of a signal is to use the SD algorithm and adaptively adjust the equalizer coefficients to produce a desired target signal [42]. This is done either by sending a pilot pattern, such as a pseudo-random binary sequence (PRBS), or with single-bit pulses to capture the pulse response of the channel for the given data rate [43]. In the case of the RIS-based equalizer, the equalizer coefficients would be the RIS reflection coefficients, as demonstrated in Fig. 2. Note the important distinction in Fig. 2 that while the conventional SD algorithm on the left-hand side achieves channel equalization via the memory elements in the n-tap FFE, the structure we propose on the right-hand side of Fig. 2 does not have any discrete memory elements in the RIS and uses the RIS reflection coefficients $\Gamma_1, \Gamma_2, \ldots, \Gamma_M$ to achieve signal equalization. It is important to note that the RIS does not contain physical delay lines or discrete memory elements; it is a passive, memoryless surface that instantaneously modifies the reflection coefficient of the incident wave. Here, the multipath channels contain the delay elements required for temporal equalization. In the conventional equalizer, an SD algorithm would only need to process the input signal at the receiver, $x(t) = s(t) * h(t)$, where $h(t)$ is the channel from the transmitter to the receiver. Assuming $s(t)$ is known, then the filter coefficients can be tuned directly to minimize the squared magnitude of the error signal $e(t) = s(t) - y(t)$ at the desired sampling instances $t = kT$, where $y(t) = x(t) * c(t)$ and $c(t)$ is the impulse response of the equalizer.

In the case of the RIS-based equalizer, the RIS reflection coefficients $\mathbf{\Gamma}$ serve as the equalizer coefficients; when combined with the cascaded channels $\boldsymbol{h}_{\mathrm{BRU}}$, they result in phase- and magnitude-shifted versions of the transmitted signal $s(t)$, each given by $x_m(t) = \Gamma_m h_{\mathrm{BRU},m}(t) * s(t)$ for $m = 1, 2, \ldots, M$. The resulting signal at the receiver is a combination of the RIS-aided propagation and the direct channel from the BS to UE and can be written as $y(t) = s(t) * h_{\mathrm{BU}}(t) + \sum_{m=1}^{M} x_m(t)$, similar to (6) but with the noise term omitted. By adjusting the coefficients correctly and combining all of these signals at the receiver, we may eliminate (or at least minimize) the signal distortion and ISI caused by multipath scattering associated with the direct BS-UE link and with the BS-RIS-UE links themselves. Because we have M such signals due to the M RIS elements, we will need to estimate M cascaded BS-RIS-UE channels in order to compute the gradients correctly and minimize the resulting error signal, as demonstrated by the SD update rule derived and discussed in this section. This channel estimation introduces both additional computational complexity and communication complexity, as this SD algorithm, called ARISE, can only begin tuning the RIS **after** all the CSI for the $\boldsymbol{h}_{\mathrm{BRU}}$ channel matrix have been obtained. This discussion is elaborated further in Section IV-E.

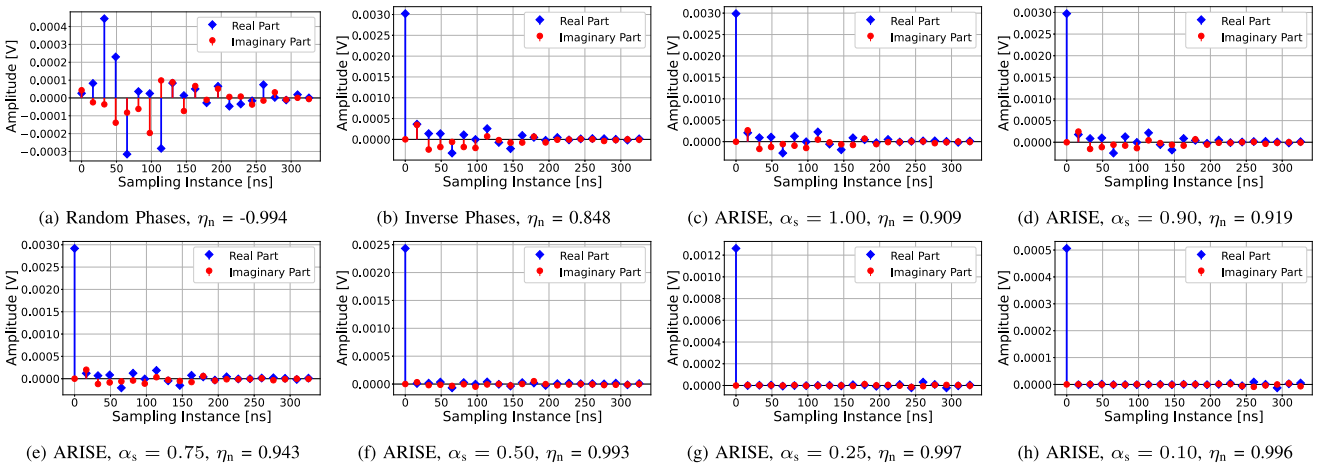
We assume the receiver samples $y(t)$ with period T at integer time indices k. Going forward, we use the discrete-time signal notation y_k as the equivalent to $y(kT)$ for all k, and the array notation $\boldsymbol{y}$ as $[y_0, y_1, \ldots, y_L]$ where L is the number of ISI components in the received pulse. For a given received pulse response $\boldsymbol{y}$, the ARISE algorithm seeks to match it to the transmitted signal $\boldsymbol{s}$, given by the $1 \times (L+1)$ array $\boldsymbol{s} = [1, 0, \ldots, 0]$, to solve the optimization problem (P1). We define a cost function J as the mean of the squared error magnitudes

$$J = E[\epsilon_k \epsilon_k^*], \tag{11}$$

with the error signal ϵ_k at each discrete time index k, defined by

$$\epsilon_k = s_k - y_k. \tag{12}$$

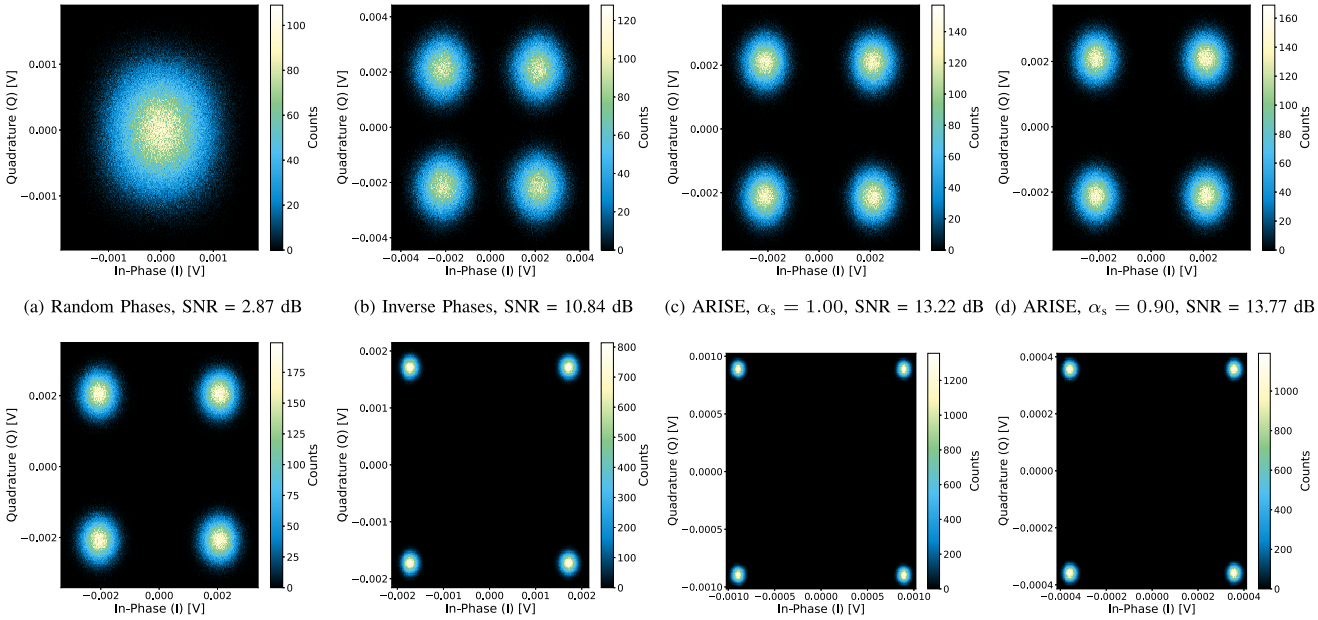
By observation of the optimization problem (P1), we note that the RIS acts as both an equalizer and a signal booster, since it has the capability to beamform and increase the SNR of the signal while also reducing the ISI. If the magnitude



(a) Random Phases, η_n = -0.994 (b) Inverse Phases, η_n = 0.848 (c) ARISE, $\alpha_s = 1.00$, η_n = 0.909 (d) ARISE, $\alpha_s = 0.90$, η_n = 0.919

(e) ARISE, $\alpha_s = 0.75$, η_n = 0.943 (f) ARISE, $\alpha_s = 0.50$, η_n = 0.993 (g) ARISE, $\alpha_s = 0.25$, η_n = 0.997 (h) ARISE, $\alpha_s = 0.10$, η_n = 0.996

FIGURE 3. Representative converged pulse responses real and imaginary parts at UE location (−20, −20), M = 100, κ = 10, n_r = 10.



(a) Random Phases, SNR = 2.87 dB (b) Inverse Phases, SNR = 10.84 dB (c) ARISE, $\alpha_s = 1.00$, SNR = 13.22 dB (d) ARISE, $\alpha_s = 0.90$, SNR = 13.77 dB

(e) ARISE, $\alpha_s = 0.75$, SNR = 15.32 dB (f) ARISE, $\alpha_s = 0.50$, SNR = 24.36 dB (g) ARISE, $\alpha_s = 0.25$, SNR = 27.57 dB (h) ARISE, $\alpha_s = 0.10$, SNR = 26.89 dB

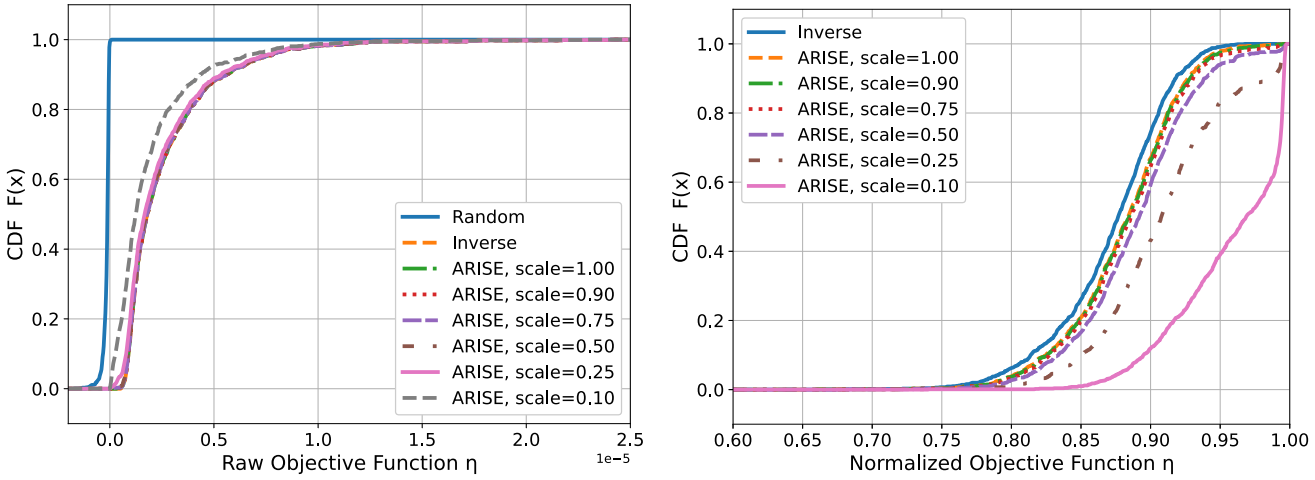
FIGURE 4. Representative converged QPSK constellations at UE location (−20, −20), M = 100, κ = 10, n_r = 10.

of s_0 is significantly larger than the tap magnitudes of $\boldsymbol{y}$, the algorithm will see most of the error contribution at the main tap and thus will focus mostly on power maximization and less on ISI minimization. On the contrary, if s_0 is relatively small, the algorithm will attempt to minimize the ISI while keeping the main tap small, because the magnitude of the ISI error in ϵ_k, $k > 0$, will become larger relative to the main tap error in ϵ_0. Note that the received pulse at sampling time $t = 0$ will go through a phase shift due to the channels and will include imaginary components and its real part may become negative. For these reasons, we normalize $\boldsymbol{y}$ based on its magnitude at $k = 0$ to match the desired pulse $\boldsymbol{s}$. This normalization accounts for the loss in signal power as the signal passes through the channel, while ensuring the ISI components are weighed correctly with respect to the main tap. Furthermore, to maximize the signal power by using the RIS as a beamformer, we scale the signal by the number of RIS elements to account for the maximum achievable gain of the passive RIS. We define a scaling variable α_s and the scaling factor

$$y_s = \alpha_s M |y_0|, \tag{13}$$

using the main tap y_0 of the initially received pulse, to scale $\boldsymbol{y}$ during iterations of ARISE. At high SNR, noise is not the dominant impairment, so more aggressive ISI suppression is beneficial and a smaller α_s can be used.

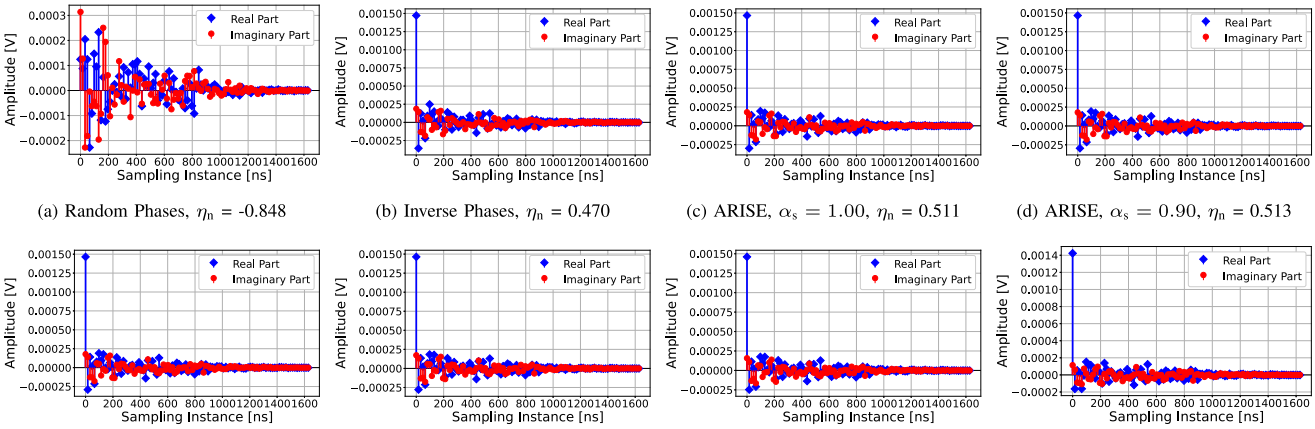
At low SNR, where noise dominates, a larger α_s prevents ARISE from over-equalizing and preserves signal power (e.g., α_s = 0.25 in Fig. 4 achieves the highest SNR, whereas α_s = 0.1 in Fig. 7 achieves the highest SNR under the same



(a) Converged η values

(b) Converged normalized η_{n} values

FIGURE 5. Converged CDFs over 1000 random channel realizations under the random walk scenario, $M = 100, \kappa = 10, n_{\text{r}} = 10$.



(a) Random Phases, η_{n} = -0.848

(b) Inverse Phases, η_{n} = 0.470

(c) ARISE, $\alpha_{\text{s}} = 1.00$, η_{n} = 0.511

(d) ARISE, $\alpha_{\text{s}} = 0.90$, η_{n} = 0.513

(e) ARISE, $\alpha_{\text{s}} = 0.75$, η_{n} = 0.518

(f) ARISE, $\alpha_{\text{s}} = 0.50$, η_{n} = 0.532

(g) ARISE, $\alpha_{\text{s}} = 0.25$, η_{n} = 0.568

(h) ARISE, $\alpha_{\text{s}} = 0.10$, η_{n} = 0.680

FIGURE 6. Representative converged pulse responses real and imaginary parts at UE location (−20, −20), $M = 100, \kappa = 0, n_{\text{r}} = 50$.

noise power but with different main tap amplitudes due to channel losses and RIS configurations). This heuristic can be implemented by allowing the receiver to estimate the noise variance relative to the signal power and provide ARISE with a coarse SNR estimate, from which α_{s} can be selected accordingly. Importantly, this guideline does not modify the optimization metric η used in the paper, as the ARISE cost function remains based on the difference between main tap power and ISI power. The scaling factor simply determines the relative emphasis placed on main tap preservation within this metric, and the SNR-based heuristic provides a principled way to choose it without altering the underlying objective.

To minimize the cost function J, the ARISE algorithm will directly modify the RIS reflection coefficients Γ_m to obtain stable gradients, rather than the reflection phases ϕ_m which affect the received signal in a nonlinear manner. The objective function of ARISE thus becomes

$$\min_{\boldsymbol{\Gamma}} E[\epsilon_k \epsilon_k^*], \tag{14}$$

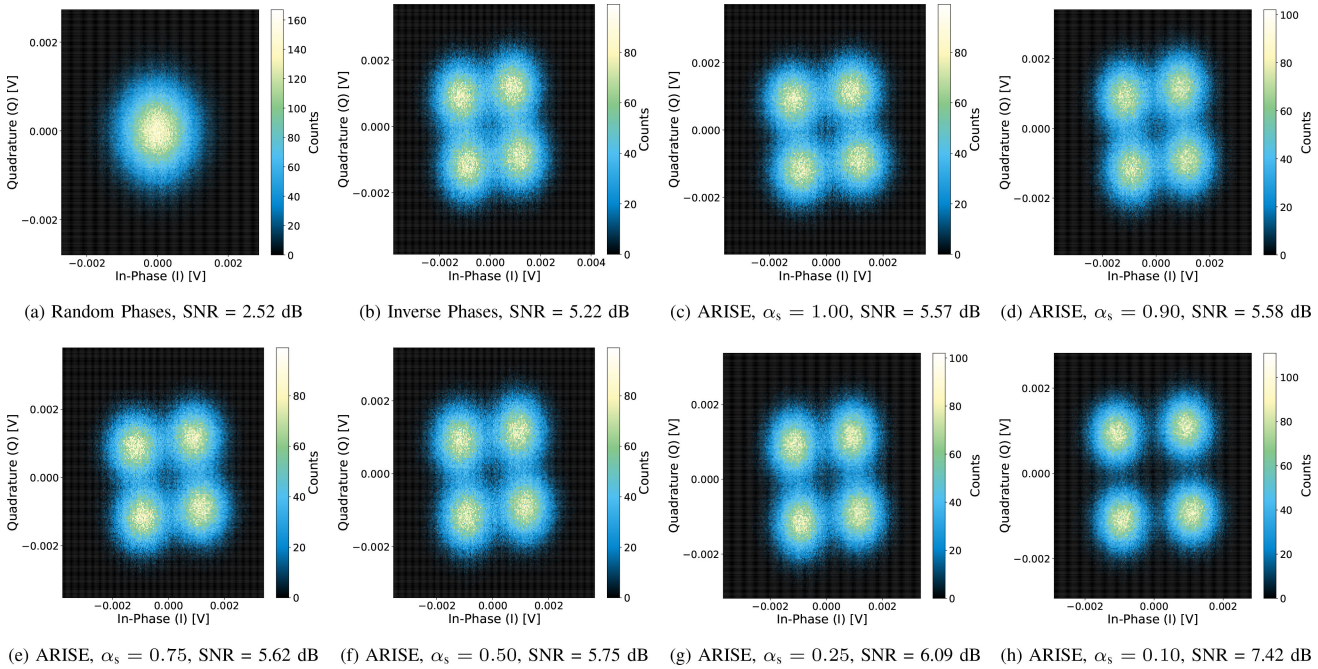
with the gradient descent update rule of the reflection coefficients $\boldsymbol{\Gamma} = [\Gamma_1, \Gamma_2, \ldots, \Gamma_M]^T$ as

$$\boldsymbol{\Gamma}_{t+1} = \boldsymbol{\Gamma}_t + \mu E\left[\sum_{l=0}^{L} \boldsymbol{h}_{\text{BRU},l}^* s_{k-l}^* \epsilon_k\right], \tag{15}$$

following the derivation in Appendix A, where $\boldsymbol{\Gamma}_t = [\Gamma_{t,1}, \Gamma_{t,2}, \ldots, \Gamma_{t,M}]^T$ is the reflection coefficient vector $\boldsymbol{\Gamma}$ at time t. We take the expectation over $L + 1$ signal taps–essentially as a sliding window across the pulse response–to calculate the ensemble average and therefore the "true gradient" during each time step [42]. The expectation is carried out using time averages under the standard assumption of ergodicity. To enforce the passivity of the RIS, we normalize the magnitude of each reflection coefficient after every update using

$$\Gamma_{t+1,m} \leftarrow \frac{\Gamma_{t+1,m}}{|\Gamma_{t+1,m}|} \tag{16}$$

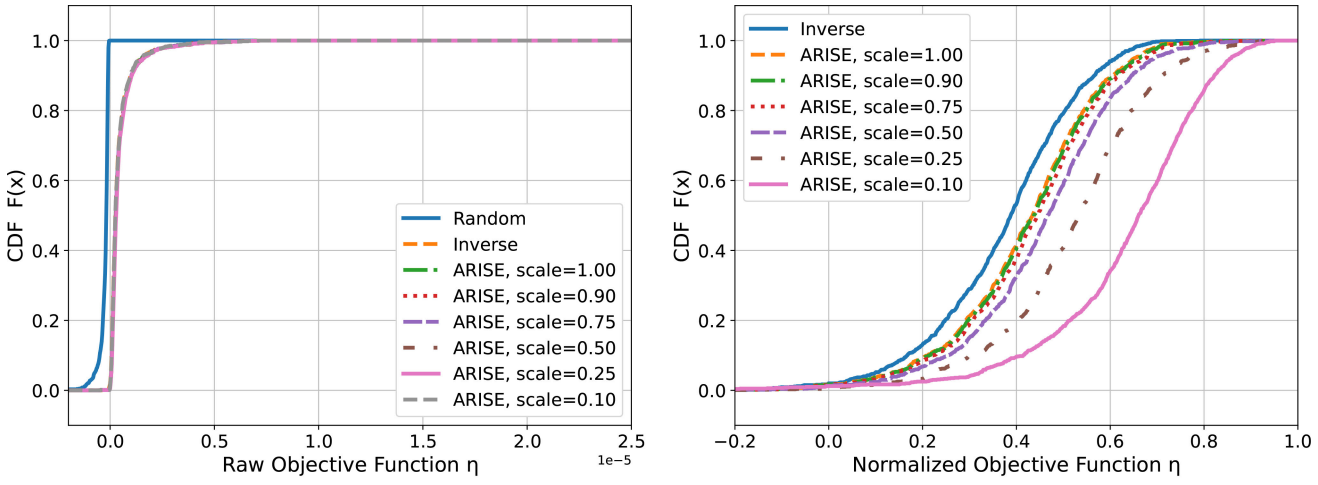
for $m = 1, 2, \ldots, M$.



(a) Random Phases, SNR = 2.52 dB (b) Inverse Phases, SNR = 5.22 dB (c) ARISE, $\alpha_s = 1.00$, SNR = 5.57 dB (d) ARISE, $\alpha_s = 0.90$, SNR = 5.58 dB

(e) ARISE, $\alpha_s = 0.75$, SNR = 5.62 dB (f) ARISE, $\alpha_s = 0.50$, SNR = 5.75 dB (g) ARISE, $\alpha_s = 0.25$, SNR = 6.09 dB (h) ARISE, $\alpha_s = 0.10$, SNR = 7.42 dB

FIGURE 7. **Representative converged QPSK constellations at UE location (−20, −20), $M = 100, \kappa = 0, n_r = 50$.**



(a) Converged η values (b) Converged normalized η_n values

FIGURE 8. **Converged CDFs over 1000 random channel realizations under the random walk scenario, $M = 100, \kappa = 0, n_r = 50$.**

In practice, the pulse responses after each RIS gradient update can be obtained by either:

- Updating the RIS configuration during each step and measuring the new pulse response from the receiver.
- Calculating the expected pulse response offline assuming the RIS is updated, using the expression in (6).

Note the tradeoff between the two approaches above: the first requires live communication with the receiver while the RIS is being optimized, whereas the second approach requires channel estimation for the direct $\boldsymbol{h}_{\mathrm{BU}}$ vector for accurate computation of the expected pulse responses.

The full ARISE algorithm is outlined in Algorithm 1. The step size for gradient descent is given by

$$\mu = \frac{\alpha_\mu (L+1)}{\sum_k |y_k|}, \tag{17}$$

with an arbitrary scaling factor α_μ and signal components $|y_k|$ taken from the initially received pulse response. The stopping condition for convergence is based on comparing the absolute difference between normalized versions of η, defined as

$$\eta_n = \frac{\eta}{\sum_k |y_k|^2}, \tag{18}$$

over ten consecutive steps against a predetermined threshold η_{th}. For an additional safety measure, we define $\eta_{n,\max}$,

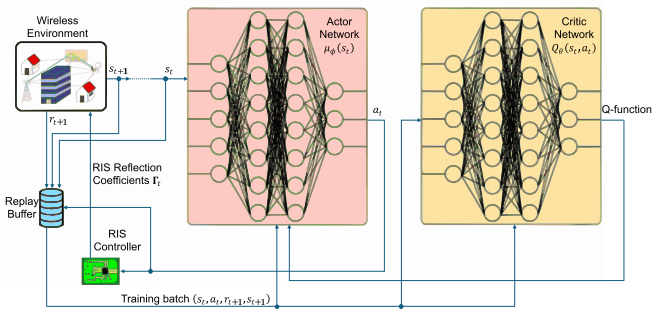


FIGURE 9. Illustration of the DRL-based RIS equalizer implementation. The actor (policy) DNN takes the state from the environment and outputs the action corresponding to RIS reflection coefficients that should improve the signal integrity. The RIS controller updates the RIS reflection coefficients and the environment changes accordingly. The environmental states (pulse responses), actions (RIS reflection coefficients), and rewards (normalized η_{n} metrics) are stored in a replay buffer, from which training batches are extracted to optimize the DNN hyperparameters. The actor is also trained using the output of the critic (Q-function).

Algorithm 1 Adaptive RIS Equalization (ARISE)

Initialize $\mathbf{\Gamma}, \boldsymbol{h}_{\text{BRU}}, \boldsymbol{s}, \eta_{\text{n,max}}, \eta_{\text{d}}, \eta_{\text{th}}, \alpha_\mu, \alpha_{\text{s}}$.
Initialize $\boldsymbol{y}$ by transmitting $\boldsymbol{s}$ and recording the received signal from (6).
Initialize $\mu \leftarrow \frac{\alpha_\mu (L+1)}{\sum_k |y_k|}$.
Initialize $y_s \leftarrow \alpha_{\text{s}} M |y_0|$.
Initialize $i \leftarrow 0$.
repeat
  $\boldsymbol{\epsilon} \leftarrow \boldsymbol{s} - \boldsymbol{y}/y_s$.
  Calculate η using (10) and η_{n} using (18).
  if $\eta_{\text{n}} > \eta_{\text{n,max}}$ **then**
    $\eta_{\text{n,max}} \leftarrow \eta_{\text{n}}$.
  end if
  Update $\mathbf{\Gamma}$ using (15).
  Normalize $\mathbf{\Gamma}$ using (16).
  Obtain new received pulse $\boldsymbol{y}'$ from (6).
  Calculate new objective functions η' using (10) and η'_{n} using (18).
  if $\eta'_{\text{n}} < \eta_{\text{n,max}} - \eta_{\text{d}}$ **then**
    $\mu \leftarrow \frac{\mu}{2}$.
    $\eta_{\text{n,max}} \leftarrow -1.0$.
  end if
  if $|\eta'_{\text{n}} - \eta_{\text{n}}| < \eta_{\text{th}}$ **then**
    $i \leftarrow i + 1$.
  else
    $i \leftarrow 0$.
  end if
  if $i \geq 10$ **then**
    break
  end if
until convergence

initially set to -1, to track the highest achieved objective function. We also introduce an auxiliary parameter η_{d} to monitor the difference between the current objective and the best recorded objective, and reduce the step size in case the algorithm overshoots and begins to diverge.

Although we are using a steepest descent approach by applying the expectation operator as in the update rule in (15), we note that a stochastic gradient descent-based implementation of ARISE is also achievable, similar to the least mean squares (LMS) algorithm [44]. The update rule becomes

$$\mathbf{\Gamma}_{t+1} = \mathbf{\Gamma}_t + \mu \sum_{l=0}^{L} \boldsymbol{h}^*_{\text{BRU},l} s^*_{k-l} \epsilon_k. \tag{19}$$

Then, instead of processing the pulse response via a sliding window and averaging the errors before each RIS update, we use the stochastic gradient obtained from processing the pulse (or any other desired pilot signal) during only a single window per time step. This approach would have less computational complexity per RIS update step, but would likely require more steps until convergence and result in noisy gradients. In the simulation results discussed in this work, we use the update rule with the expectation operator as in (15) to take complete advantage of every sample in the received pulse response, thereby equalizing the signal using a smooth gradient.

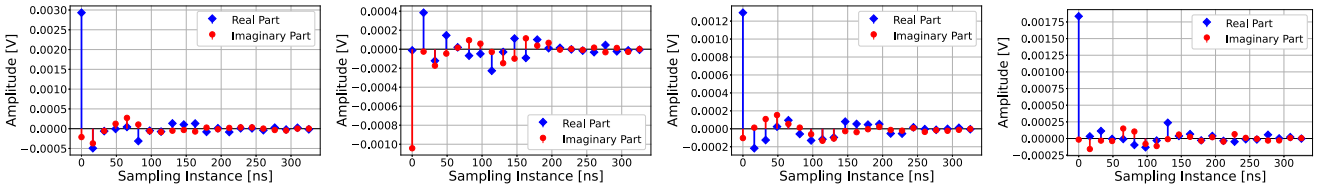
B. ARISE SIMULATION RESULTS

The RIS environment consists of a single-antenna BS, single-antenna UE, and by default, an $M = 100$-element RIS. The spacing between the RIS elements is $d_x = 20$ mm and the wireless signal carrier frequency is $f_c = 2.45$ GHz. We use a symbol duration $T = 16.3$ ns with a modulated signal bandwidth of $B = 61.44$ MHz, in accordance with the 3GPP standard for 5G New Radio (NR) [45]. The normalized gain factors are $G = G' = -43$ dBm, as in [21], and the noise power is $P_n = -96$ dBm in the AWGN signal $n(t)$. We define a 2-dimensional coordinate system with the BS at the center with coordinates (in meters) $(0, 0)$. The BS directly illuminates the RIS, located at $(10, 0)$, at normal incidence. The starting location for the UE is at $(-20, -20)$ and the UE is allowed to move within the rectangular area bounded between the points $(-100, -100)$ and $(-10, 100)$. At the start of each channel coherence block, the NLoS channels are randomly generated and the UE moves in a random walk fashion, each time in a random direction and a random distance up to 20 m away from the previous point, both uniformly distributed. Considering the ISI paths between the BS and the RIS may also be scattered and delayed in the RIS-UE link, we define the relationship between the number of delayed paths n_{r} and number of ISI terms L as $L = 2n_{\text{r}}$.

We define two baseline RIS reflection coefficient schemes for comparison:

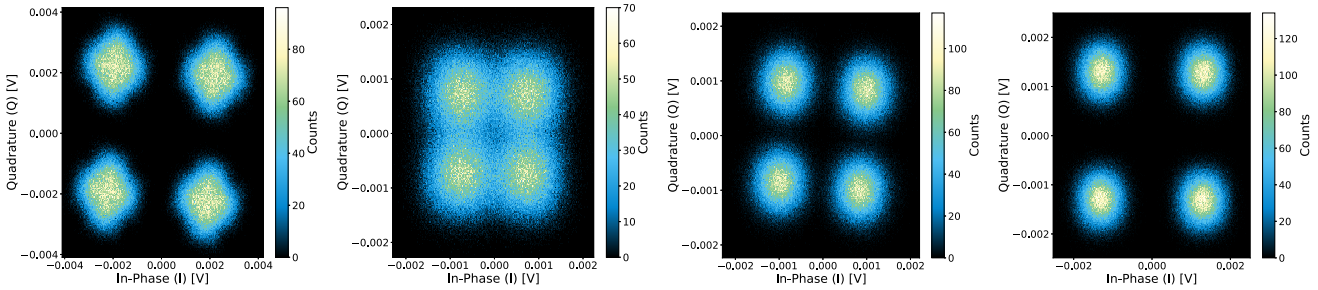
- **Random Phases**–the RIS reflection phases are sampled uniformly between $[-\pi, \pi]$.
- **Inverse Phases**–the RIS reflection coefficients are the complex conjugates of the M channel coefficients in $\boldsymbol{h}_{\text{BRU},k}$, evaluated at the sampling instance $k = 0$ and normalized to unity magnitude, e.g.,

$$\mathbf{\Gamma} = \exp(-j \angle \boldsymbol{h}_{\text{BRU},0}), \tag{20}$$



(a) ARISE, η_n = 0.848 (b) DDPG, η_n = -0.258 (c) TD3, η_n = 0.770 (d) SAC, η_n = 0.883

FIGURE 10. Representative converged pulse responses real and imaginary parts at UE location (−20, −20), $M = 100, \kappa = 10, n_r = 10$.



(a) ARISE, SNR = 11.01 dB (b) DDPG, SNR = 5.20 dB (c) TD3, SNR = 9.00 dB (d) SAC, SNR = 12.11 dB

FIGURE 11. Representative converged QPSK constellations at UE location (−20, −20), $M = 100, \kappa = 10, n_r = 10$.

intending to produce a coherent sum at the receiver for y_0 [11].

For the ARISE algorithm, we set the threshold parameter $\eta_{th} = 10^{-5}$ and the overshoot parameter $\eta_d = 0.5$. We scale the step size μ by $\alpha_\mu = 10$ for rapid convergence.

The first simulation compares the baseline schemes with the converged results from ARISE using various scaling factors α_s for a single representative scenario. In this case, the Rician factor for the BS-RIS-UE link is $\kappa = 10$ and the link between the BS and UE is characterized by Rayleigh fading due to obstacles in the path–modeling pre-existing signal distortion and highlighting the dependency of the wireless environment on the RIS. The number of delayed paths is $n_r = 10$, thus the number of ISI terms is $L = 20$. The converged pulse responses for each algorithm are plotted in Fig. 3, assuming the unit interval (UI) $T = 16.3$ ns. The corresponding QPSK constellations, obtained from PRBS signals using the same channels and converged RIS configurations, are presented in Fig. 4. The simulation was then repeated using the random walk method over 1000 random channel coherence blocks. At the beginning of each channel realization, the RIS reflection coefficient matrix $\tilde{\mathbf{\Gamma}}$ is initialized with random reflection phases and unity magnitudes. The CDF plots of the converged η and normalized η_n metrics across those 1000 simulations are presented in Fig. 5.

The pulse responses reveal that under the Random Phases configuration, the channel exhibits a long ISI tail with substantial real and imaginary components, while the main tap is significantly weaker than many of the interfering terms. A conventional receiver would struggle to recover the signal data under such conditions to achieve low bit error rate (BER). Under the Inverse Phases configuration, the main tap amplitude is restored, but the ISI tail remains largely intact. The advantage of ARISE becomes evident when comparing the subsequent configurations: for a scaling factor $\alpha_s = 1$, ARISE achieves a main tap amplitude near-equal to the Inverse Phases case while simultaneously reducing the ISI power. As α_s decreases and temporal equalization is prioritized over spatial beamforming, the ISI components are progressively suppressed and eventually become nearly negligible. The main tap amplitude remains stable down to $\alpha_s = 0.5$, after which it decreases slightly–yet this reduction is accompanied by a substantial suppression of ISI, yielding a cleaner overall pulse response. This behavior is further reflected in the QPSK constellations in Fig. 4, where ARISE yields both an SNR boost and a noticeable reduction in symbol variance. The CDF results in Fig. 5 reinforce this trend: although ARISE and Inverse Phases achieve near-equal η values, ARISE consistently attains higher normalized η_n, highlighting its superior equalization capability. A similar pattern appears in a more challenging setting where the main tap path is entirely NLoS ($\kappa = 0$), and the number of ISI terms increase to $L = 100$ as $n_r = 50$. As shown in the representative single-run scenarios in Figs. 6 and 7, and in the corresponding CDF results for the same initial configuration using the random walk method over 1000 random channel realizations (Fig. 8), ARISE continues to suppress ISI effectively and maintain a strong main tap component even under these harsher propagation conditions.

The observed convergence behavior of ARISE can be understood through the structure of the underlying optimization problem. Unlike classical equalization, where the optimization variables enter linearly and lead to a quadratic (convex) objective, RIS-based equalization involves phase variables that affect the channel multiplicatively through complex exponentials, resulting in a nonconvex optimization

landscape. Therefore, ARISE is a projected steepest-descent method applied to a smooth but generally nonconvex objective. Under standard step-size rules (e.g., fixed small step or backtracking), the algorithm ensures monotonic decrease of the objective and convergence to a first-order stationary point, with iterates remaining bounded due to the unit-modulus phase constraint; these properties follow from standard results in smooth nonconvex optimization. In practice, convergence speed depends on channel conditioning, but stable behavior is observed across channel realizations in our simulation results, and warm starts across coherence intervals further accelerate convergence. While global optimality is not guaranteed due to nonconvexity, multiple initializations or warm starts can be used to mitigate local minima, with negligible additional overhead in slowly varying channels.

The feasibility of ARISE critically depends on the availability of accurate CSI, since its update rule in (15) requires explicit knowledge of the cascaded channel matrix $\boldsymbol{h}_{\mathrm{BRU}}$ to suppress the dominant ISI components. Estimating this high-dimensional channel matrix becomes increasingly burdensome as the RIS scales, and eliminating L ISI taps in principle requires $M(L+1)$ coefficients associated with the M RIS elements and the $(L+1)$ signal components (main tap and ISI). Although practical estimators may jointly recover the main and delayed channels using fewer than $M(L+1)$ pilot configurations, the design of such estimators for RIS-aided wideband channels with ISI lies outside the scope of this work [3], [34], [46]. Importantly, ARISE cannot begin optimizing the RIS until channel estimation is completed, introducing additional delay and communication overhead due to the need for repeated pilot transmissions across multiple RIS configurations. Moreover, ARISE implicitly assumes a linear and frequency-flat RIS response, whereas realistic RIS hardware exhibits element coupling, amplitude losses, phase errors, and frequency-dependent reflection coefficients–effects that complicate accurate modeling, especially in wideband scenarios [31], [47], [48], [49]. These issues are further discussed in Section IV-E. We address the above problems by using the DRL-based methods discussed in the following sections, which avoid explicit channel estimation altogether and are inherently model-free. These algorithms only require a single pilot symbol transmission to obtain the pulse response at each time step, and the agent can begin optimizing the RIS as soon as a minimal number of samples are stored in its memory.

IV. DEEP REINFORCEMENT LEARNING

In this section, we will review general concepts behind DRL, relate it to the RIS equalizer scenario, and introduce three different algorithms to solve the optimization problem (P1).

A. BACKGROUND

In the most general sense, reinforcement learning (RL) involves an agent that takes actions in a specific environment and receives different rewards for different actions. The goal is to learn a policy that guides the agent to choose actions that maximize long-term rewards through trial and error. In the case of DRL, we employ deep neural networks (DNNs) to model the environment and the agent. The key elements in RL, applied to our RIS case, are described as follows [23], [50]:

1) **State:** A set of parameters characterizing the environment. We define the state at time t as $s_t \in \mathcal{S}$ as a function of the received pulse response $\boldsymbol{y}$ at time t, denoted by $\boldsymbol{y}_t$, where $\mathcal{S}$ is the continuous state space. We define the notation $y_{t,k}$ to identify the sample at index k from the received pulse response $\boldsymbol{y}_t$. Since the pulse response is complex due to reflection phase shifts in the channel, we expand the pulse of length $L+1$ to an array of length $2(L+1)$ which contains the real and imaginary parts of each sampled point in the pulse, to be used as inputs to the DNNs. The state is defined as the resulting array $s_t = \boldsymbol{y}'_t$ where $\boldsymbol{y}'_t = [\Re\{y_{t,0}\}, \Im\{y_{t,0}\}, \ldots, \Re\{y_{t,L}\}, \Im\{y_{t,L}\}]$.
2) **Action:** The choice $a_t \in \mathcal{A}$ of the agent at each time step to move from the current state of the environment s_t to the next state s_{t+1}, sampled from the continuous action space $\mathcal{A}$. We define the action as the RIS reflection coefficients $\boldsymbol{\Gamma}$ at time t that will aim to improve the quality of the pulse response, i.e., $a_t = \boldsymbol{\Gamma}'_t$. Here, we also split the real and imaginary parts of the M-dimensional reflection coefficient vector $\boldsymbol{\Gamma}_t$ into an array of length $2M$ as $\boldsymbol{\Gamma}'_t = [\Re\{\Gamma_{t,1}\}, \Im\{\Gamma_{t,1}\}, \ldots, \Re\{\Gamma_{t,M}\}, \Im\{\Gamma_{t,M}\}]^T$.
3) **Reward:** An instantaneous performance metric provided by the environment to inform the agent of the quality of its action a_t at state s_t as it moves to the next state s_{t+1}. Here, the reward r_{t+1} is defined as the normalized version of η according to (18), η_{n}, expressed by

$$r_{t+1} = \frac{\eta}{\sum_k |y_{t+1,k}|^2}. \tag{21}$$

This normalization ensures the rewards are always between -1 at the worst case and $+1$ at the best case, to provide stable gradients when the DNNs are trained. Note that per the definition of η in (10) and due to the normalization here, the reward function is penalized when the received signal $\boldsymbol{y}_t$ is dominated by ISI, when the real part of $y_{t,0}$ is small, and when the imaginary part of $y_{t,0}$ is significant, by accounting for the power at each sampled index of $\boldsymbol{y}_t$. This also indicates that there will exist different local optima in RIS configurations, similar to how the performance of ARISE depends on the scaling of the received signal. This problem invokes the tradeoffs between *exploration* and *exploitation* of the DRL algorithms, which will be discussed later.
4) **Experience:** Previous observations and interactions of the agent with the environment stored in memory as $(s_t, a_t, r_{t+1}, s_{t+1})$.

5) **Policy:** Determines the probability of choosing a specific action based on the current state, denoted by $\pi(a_t|s_t)$ and satisfying $\sum_{a_t\in\mathcal{A}}\pi(a_t|s_t)=1$. Our goal is to determine a policy that will quickly find a set of stable RIS reflection coefficients to reduce the ISI and improve the SNR of the received wireless signal.
6) **State-action value function:** The quality of taking action a_t in state s_t based on the potential future rewards.

To ensure a fair comparison between ARISE and the DRL-based methods, all algorithms are designed to optimize the same objective. ARISE minimizes the mean-squared error between a scaled version of the received pulse and a desired target pulse, and its convergence is monitored through the normalized equalization metric η_{n}. The DRL algorithms directly maximize this same normalized metric, but do so through stochastic policy-gradient updates rather than deterministic gradient descent. Thus, although the optimization procedures differ–ARISE relying on full cascaded CSI and deterministic updates, and DRL relying on exploration and stochastic training–the underlying objective definition remains consistent across all methods.

Amplification effects are also implicitly reflected in the DRL reward: because η_{n} measures the difference between main tap power and ISI power, normalized by the total received power, any RIS configuration that increases the main tap amplitude improves the reward, while configurations that reduce the main tap or increase ISI penalize it. The normalized form of the metric is used in the DRL setting to avoid reward-scale variations across different channel realizations, since the receiver does not know the absolute path loss. This normalization ensures stable learning behavior across coherence blocks while preserving the same equalization objective used by ARISE.

We define V_t as the future cumulative discounted reward

$$V_t=\sum_{\tau=0}^{\infty}\gamma^{\tau}r_{t+\tau+1}, \tag{22}$$

where τ takes values from nonnegative integers and $0<\gamma<1$ is the discount rate. The state-action value function is given by the Q-function, which satisfies the Bellman equation

$$\begin{aligned}Q_\pi(s_t,a_t)&=E_\pi[V_t|s_t=s,a_t=a]\\&=E_\pi[r_{t+1}|s_t=s,a_t=a]\\&\quad+\gamma\sum_{s'\in\mathcal{S}}P^a_{ss'}\left(\sum_{a'\in\mathcal{A}}\pi(a'|s')Q_\pi(s',a')\right)\end{aligned} \tag{23}$$

and can be learned recursively to find the optimal policy by utilizing the Q-learning algorithm [51]. In the above, $P^a_{ss'}$ is the transition probability from state s to state s' given action a in the Markov decision process (MDP). A target Q-function is initialized and then learned to achieve the optimal Q-function, with the expectation taken over all possible next states s_{t+1},

$$Q^*(s_t,a_t)=E\left[r_{t+1}+\gamma\max_{a_{t+1}}Q^*(s_{t+1},a_{t+1})\right], \tag{24}$$

using the update rule

$$\begin{aligned}Q^*(s_t,a_t)\leftarrow(1-\alpha)Q^*(s_t,a_t)\\+\alpha\left(r_{t+1}+\gamma\max_{a_{t+1}}Q_\pi(s_{t+1},a_{t+1})\right),\end{aligned} \tag{25}$$

where α is the learning rate. By definition, the Q-function determines the future cumulative discounted rewards based on a given policy, the current state, and the choice of action. The optimal Q-function–the one that would yield the highest future rewards for the current state and action–would follow the desired optimal policy, π^*.

Considering the context of this work, the action space is high-dimensional and scales with the size of the RIS, prompting the use of machine learning to estimate the Q-function and optimal policy using two DNNs. The *critic* DNN estimates the Q-function based on the states, actions, and rewards as the agent acts in the environment. Using the symbol θ to represent the critic DNN parameters (neuron weights, activation functions, number of layers, etc.), we have $Q(s_t,a_t)=Q_\theta(s_t,a_t)$. The *actor* DNN seeks the optimal policy, taking the current state as the input and estimating an action at the output. Depending on the algorithm choice, each of the two DNNs may also split into training and target DNNs [52]:

- **Training critic** $Q_\theta(s_t,a_t)$ learns the Q-function via gradient descent.
- **Target critic** $Q_{\theta_{\mathrm{targ}}}(s_t,a_t)$ provides a slowly updated copy of the critic used to provide stable time-difference (TD) targets.
- **Training actor** $\mu_\phi(s_t)$ is the policy network and is optimized using the deterministic policy gradient.
- **Target actor** $\mu_{\phi_{\mathrm{targ}}}(s_t)$ is a slowly updated copy of the actor used to generate stable next-state actions for loss function computations during training.

Note that the DNNs can now be trained by updating their weights using stochastic optimization algorithms and backpropagation as

$$\theta_{t+1}=\theta_t-\mu\nabla_\theta l(\theta), \tag{26}$$

where μ is the learning rate and $\nabla_\theta l(\theta)$ is the gradient of the loss function $l(\theta)$ [53]. Modern machine learning frameworks like TensorFlow [54] automatically compute gradients through backpropagation, so we do not need to derive these expressions manually.

Another key component of DRL is the *experience replay buffer*, which stores the interactions of the agent with the environment. Training the DNNs on randomly sampled batches from this buffer, rather than on consecutive states, reduces the strong temporal correlations present in the data. This increased stochasticity helps prevent the networks from overfitting to recent experiences and becoming trapped in local optima, thereby improving overall learning performance. This mechanism also interacts with the broader concept of *exploration versus exploitation*: exploration seeks out new states, often through noisy or randomized actions,

while exploitation leverages previously collected experiences, such as those stored in the replay buffer, to refine the policy.

The principles discussed in this subsection provide the foundation for the DRL algorithms presented next: DDPG, TD3, and SAC, where these ideas are applied in concrete algorithmic frameworks.

B. DDPG

Deep Deterministic Policy Gradient (DDPG) is an off-policy (learns a target policy from data generated by a different behavior policy [55]), model-free actor-critic algorithm designed for continuous control tasks. It combines the deterministic policy gradient framework with techniques from DRL, using separate actor and critic networks along with target networks and a replay buffer. The critic learns an action-value function, while the actor is updated by back-propagating the gradient of the critic with respect to the action, enabling the learning of a deterministic policy in high-dimensional continuous action spaces [52], [55]. The actor DNN takes the state as its input and outputs the continuous action based on its learned policy, and the critic takes the state and the action, and outputs the estimated Q-function $Q(s_t, a_t)$ according to (23) [23]. Referring to (25), the role of the actor is to find the best action that will result in the $\max_{a_{t+1}} Q_\pi(s_{t+1}, a_{t+1})$ term, thus eliminating the need for exhaustive search and non-convex optimization. DDPG uses a deterministic policy $\mu_\phi(s_t)$, which serves the same role as a stochastic policy $\pi(a_t|s_t)$ but outputs a single action rather than a distribution over actions.

The top-level algorithm is outlined as follows. We have an RIS-aided environment as described in the previous section, where the current state s_t is defined as the received sampled pulse response at time t, $\boldsymbol{y}'_t$, and the current action $a_t = \mu_\phi(s_t)$ is the update to the reflection coefficients of the RIS by passing $\boldsymbol{y}'_t$ into the policy DNN, yielding $\boldsymbol{\Gamma}'_t$. We normalize the $\boldsymbol{y}'_t$ array to the limits $[-1, 1]$ to produce stable gradients when training the DNNs using backpropagation [53]. We define the current "step" as the current state of the environment with the current RIS configuration, and the current "episode" as the time frame during which the wireless channels remain coherent. At each state, we induce exploration by adding a random noise vector $\boldsymbol{\epsilon}$ to the RIS reflection coefficients, clip the reflection coefficients to $[-1, 1]$, and assign unity magnitudes after combining the separated real and imaginary parts from $\boldsymbol{\Gamma}'_t$ into $\boldsymbol{\Gamma}_t$ and using (16). After updating the RIS reflection coefficients, we obtain the next (normalized) state $s_{t+1} = \boldsymbol{y}'_{t+1}$ and next reward r_{t+1} from (21), and store $(s_t, a_t, r_{t+1}, s_{t+1})$ in the replay buffer $\mathcal{D}$.

To update the DNNs, we randomly sample a batch $\mathcal{B}$ of size B from the replay buffer, compute and apply the gradients to all DNNs as shown in Algorithm 2. The loss function used to train the critic with gradient descent is given by the mean-squared Bellman error (MSBE) function, used to determine how close $Q_\theta(s, a)$ comes to satisfying the Bellman equation from (24) [56],

$$l(\theta) = E\Big[\big(Q_\theta(s,a) - (r' + \gamma Q_{\theta_{\text{targ}}}(s', \mu_{\phi_{\text{targ}}}(s')))\big)^2\Big], \tag{27}$$

Algorithm 2 Deep Deterministic Policy Gradient (DDPG) for RIS Equalizer

Initialize both training and target DNNs for both actor and critic, assign $\phi_{\text{targ}} \leftarrow \phi, \quad \theta_{\text{targ}} \leftarrow \theta$.
for episode in N_{ep} **do**
  for t in N_{steps} **do**
    Observe state $s_t \leftarrow \boldsymbol{y}'_t$ and set RIS coefficients

$$\boldsymbol{\Gamma}'_t \leftarrow \text{clip}(\mu_{\phi_{\text{targ}}}(\boldsymbol{y}'_t) + \boldsymbol{\epsilon}, -1, 1), \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_{\text{a}}^2)^{2M\times 1}.$$

    Set $a_t \leftarrow \boldsymbol{\Gamma}'_t$.
    Update RIS reflection coefficients as

$$\boldsymbol{\Gamma}_t \leftarrow [\Gamma'_{t,1} + j\Gamma'_{t,2}, \ldots, \Gamma'_{t,2M-1} + j\Gamma'_{t,2M}]^T$$

    and normalize using (16).
    Observe new state $s_{t+1} \leftarrow \boldsymbol{y}'_{t+1}$.
    Calculate rewards r_{t+1} using (21).
    Store $(s_t, a_t, r_{t+1}, s_{t+1})$ in the replay buffer $\mathcal{D}$.
    if $|\mathcal{D}| \geq B$ **then**
      for i in N_{train} **do**
        Sample batch $\mathcal{B} = \{(s, a, r', s')\}$ from $\mathcal{D}$.
        Calculate $l(\theta)$ using (27).
        Update critic:

$$\theta \leftarrow \theta - \mu_{\text{c}} \nabla_\theta l(\theta).$$

        Calculate $l(\phi)$ using (28).
        Update actor:

$$\phi \leftarrow \phi - \mu_{\text{a}} \nabla_\phi l(\phi).$$

        Soft-update target critic:

$$\theta_{\text{targ}} \leftarrow \tau\theta + (1-\tau)\theta_{\text{targ}}.$$

        Soft-update target actor:

$$\phi_{\text{targ}} \leftarrow \tau\phi + (1-\tau)\phi_{\text{targ}}.$$

      end for
    end if
  end for
end for

where the expectation is taken over a randomly selected batch of (s, a, r', s') from the replay buffer $\mathcal{D}$. The actor is trained using gradient ascent with the loss function

$$l(\phi) = E\Big[Q_\theta\big(s, \mu_\phi(s)\big)\Big], \tag{28}$$

to learn a policy that maximizes the Q-function. The target DNNs are updated using Polyak averaging [57] with a weighting factor τ,

$$\theta_{\text{targ}} \leftarrow \tau\theta + (1-\tau)\theta_{\text{targ}}. \tag{29}$$

Note that during each time step t, the actor and critic DNNs may be trained N_{train} times by sampling random batches from the replay buffer. However, they will only start training once

the replay buffer contains sufficiently many samples to form a batch.

C. TD3

Although DDPG can achieve strong performance, it often suffers from overestimation bias in the learned Q-function, which can destabilize training and increase sensitivity to hyperparameters [58]. Twin Delayed DDPG (TD3) is an algorithm that builds on the foundation of DDPG and seeks to address the above issues. It introduces the concept of *clipped double Q-learning*, where two "twin" critic networks are learned (each with its own training and target networks), Q_{θ_1} and Q_{θ_2}. During the gradient update, we modify the loss function in (27) to take the minimum between the two target critic Q-function estimators

$$l(\theta_i) = E\Big[\Big(Q_{\theta_i}(s,a) - (r' + \gamma \min_{i=1,2} Q_{\theta_{\text{targ},i}}(s', \mu_{\phi_{\text{targ}}}(s')))\Big)^2\Big], \quad (30)$$

thus accounting for the overestimation bias.

Additionally, TD3 introduces *delayed policy updates* to reduce the error in the value function estimates and decrease the risk of divergence, by updating the actor network and target networks every d_{TD3} steps. This allows the critic to make more progress among policy updates, reducing the propagation of critic estimation errors into the policy.

Finally, TD3 uses *target policy smoothing regularization* during training. When computing the target value to train the critic, TD3 adds clipped noise to the output of the target actor $\mu_{\phi_{\text{targ}}}(s)$, encouraging the critic to learn a smoother Q-function and preventing exploitation of sharp value peaks.

The pseudocode for the TD3 algorithm is given in Appendix B as Algorithm 3.

D. SAC

Soft Actor-Critic (SAC) extends the actor-critic framework by introducing *maximum entropy reinforcement learning*, which augments the standard objective with an entropy term that encourages exploration and robustness [59]. Unlike DDPG and TD3, which rely on deterministic policies, SAC learns a stochastic policy $\pi_\phi(a|s)$ that maximizes both expected return and policy entropy. The resulting objective is

$$J(\pi) = E_\pi\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H(\pi(\cdot|s_t))\right)\right], \quad (31)$$

where α is the temperature parameter that controls the tradeoff between reward maximization and the entropy $H(\cdot)$, and γ is the discount rate from (22). Note that the exploration in DDPG and TD3 solely depends on adding noise to the RIS reflection coefficients that are selected from the policy DNN. In the case of SAC, the stochastic policy yields different actions by producing Gaussian distributions for each RIS reflection coefficient and randomly sampling from them.

SAC employs three main DNNs: a stochastic policy network π_ϕ as the actor, and two Q-function approximators Q_{θ_1} and Q_{θ_2} as the critics (with their training and target networks). The two critics resemble the double Q-learning introduced in TD3 to mitigate overestimation bias by taking the minimum of their outputs when forming the target for the loss function. However, SAC differs in that the loss function incorporates both the next-state value and the entropy of the policy. The new MSBE loss function to train the critics thus becomes

$$l(\theta_i) = E\Big[\Big(Q_{\theta_i}(s,a) - \Big(r' + \gamma\big(\min_{i=1,2} Q_{\theta_{\text{targ},i}}(s', \mu_{\phi_{\text{targ}}}(s')) - \alpha \log \pi_\phi(a'|s')\big)\Big)\Big)^2\Big]. \quad (32)$$

The actor aims to maximize both the expected future rewards and the expected future entropy. The actor loss function is

$$l(\phi) = E[\min_{i=1,2} Q_{\theta_i}(s,a) - \alpha \log \pi_\phi(a|s)]. \quad (33)$$

A key feature of SAC is the *automatic temperature tuning mechanism*, which adjusts α during training to maintain a target entropy level [60]. This allows the algorithm to adaptively balance exploration and exploitation without manual tuning of the entropy weight, using the loss function for gradient descent

$$l(\log \alpha) = E[-\log \alpha(\log \pi_\phi(a|s) - \mathcal{H})], \quad (34)$$

where $\mathcal{H} = -\dim \mathcal{A} = -2M$ is the target entropy and $\mathcal{A}$ is the action space corresponding to the $2M$ real and imaginary parts of RIS reflection coefficients. We update the log value of α to avoid accidental negative α values and ensure smooth and stable gradients. The pseudocode for SAC is given in Appendix B as Algorithm 4.

Overall, SAC provides several advantages over deterministic methods such as DDPG and TD3, mainly due to its improved exploration through stochasticity, reduced sensitivity to hyperparameters, and enhanced stability due to entropy regularization and double Q-learning. All the above algorithms are effective contestants for high-dimensional continuous control tasks such as the RIS equalization problem discussed in this work. In the following subsection, we discuss the computational and communication costs of each of these algorithms to provide a fair comparison between them and the ARISE algorithm.

TABLE 1. Per-step computational complexities of the proposed algorithms.

Algorithm	Per-Step Complexity
ARISE	$\mathcal{O}(M(L+1)^2)$
ARISE-SGD	$\mathcal{O}(M(L+1))$
DDPG	$\mathcal{O}(N_{\text{train}}(C_{\text{critic}} + C_{\text{actor}}))$
TD3	$\mathcal{O}(N_{\text{train}}(2C_{\text{critic}} + \frac{1}{d}C_{\text{actor}}))$
SAC	$\mathcal{O}(N_{\text{train}}(2C_{\text{critic}} + C_{\text{actor}}))$

E. COMPLEXITY ANALYSIS

Following the structure of the update rules in DDPG, TD3, and SAC, we derive the per-update computational cost in

terms of the number of forward/backward passes through the actor and critic networks. To our knowledge, no prior work provides a formal asymptotic complexity analysis of these algorithms. We define the cost of a single forward/backward pass per batch through a single DNN as C, which depends on the architecture and hyperparameters of each DNN and the batch size. We assume all batch sizes are the same and the DNN architectures across the three different algorithms are similar such that C does not vary significantly among the algorithms. Note that the overall computational complexity until convergence of each algorithm would vary depending on its performance for the RIS equalization problem–whether training is required over a long period of time, or if the reflection coefficients can be optimized quickly. The per-step complexities of all the algorithms are summarized in Table 1.

For DDPG, we update the actor and critic DNNs N_{train} times per step, yielding $T_{\text{DDPG}} = \mathcal{O}(N_{\text{train}}(C_{\text{critic}} + C_{\text{actor}}))$, where C_{critic} is the cost per critic update, and C_{actor} is the cost per actor update. We leave the target DNN updates out of this expression, since the complexity of Polyak averaging is insignificant compared to that of a forward/backward pass through a DNN.

In the case of TD3, we have two critic DNNs and a single actor. The actor updates every d_{TD3} iterations, yielding $T_{\text{TD3}} = \mathcal{O}\left(N_{\text{train}}\left(2C_{\text{critic}} + \frac{1}{d_{\text{TD3}}}C_{\text{actor}}\right)\right)$. SAC also has two critic DNNs and a single actor, with an actor update every iteration. Assuming the temperature and target updates are negligible compared to the backpropagation required for DNN training, we obtain $T_{\text{SAC}} = \mathcal{O}(N_{\text{train}}(2C_{\text{critic}} + C_{\text{actor}}))$.

To derive the per-update complexity of ARISE, we refer to the update rule in (15). We perform vector multiplications on M-dimensional vectors $L+1$ times due to the length of the captured pulse response. Under the assumption that each time step introduces a new pulse response with $L+1$ taps to equalize, we take the expectation over $L+1$ signal taps to create the ensemble average, yielding a final per-step cost of $T_{\text{ARISE}} = \mathcal{O}(M(L+1)^2)$ [42]. Note that for the stochastic gradient descent (SGD) version of ARISE in (19), the cost becomes $T_{\text{ARISE-SGD}} = \mathcal{O}(M(L+1))$ as the pulse is processed only a single time per RIS update, without averaging an ensemble.

The per-step complexity ultimately depends on the size of the DNNs and their activation functions that yield C_{critic} and C_{actor}. Depending on the available hardware at the RIS controller, the training of the DNNs may be accelerated to reduce the amount of time required per step.

The computational and signaling complexity of the proposed approach scales linearly with the number of RIS elements M. In the ARISE algorithm, each iteration updates M phase variables, resulting in per-iteration complexity on the order of $\mathcal{O}(M)$, while the total cost scales as $\mathcal{O}(N_I M)$ with the number of iterations N_I. For DRL-based methods, at deployment, the inference complexity scales approximately linearly with M for standard network architectures, as the output dimension of the policy DNN scales linearly with M in our implementation. The control signaling overhead is also $\mathcal{O}(M)$, as M quantized phase values must be conveyed to the RIS per update, which occurs at the channel coherence timescale rather than the symbol rate. These complexities remain manageable even for large RIS arrays due to the relatively slow update rate and the absence of per-symbol adaptation. This linear scaling is comparable to that of conventional large-array beamforming systems.

Beyond computational and signaling complexity, the proposed approach benefits from a natural separation of timescales between fast symbol-level dynamics and slower channel variations. In representative deployment scenarios such as indoor industrial environments or fixed backhaul links, channel coherence times are typically on the order of milliseconds, compared to symbol durations in the nanosecond range, allowing RIS configurations to be updated infrequently relative to the symbol rate. Consequently, control latency requirements are moderate and compatible with current RIS implementations, while still enabling effective over-the-air equalization without the need for rapid per-symbol adaptation.

Furthermore, the communication complexities associated with ARISE and DRL are significantly different from each other. We note that the success of our ARISE algorithm primarily depends on available CSI of the $\boldsymbol{h}_{\text{BRU}}$ cascaded channel matrix for the gradient descent rule to work correctly. As documented in prior RIS channel estimation works, CSI acquisition with an M-element RIS typically requires configuring the RIS into at least M orthogonal patterns (e.g., DFT-based or ON/OFF patterns) so that the cascaded channel can be uniquely identified. Although feasible, this procedure additionally requires computationally intensive channel estimation algorithms that scale with the RIS size, as documented in recent works such as [3], [18], [61], [62], [63], and [64].

To quantify the overhead associated with ARISE, estimating the full BS-RIS-UE channel $\boldsymbol{h}_{\text{BRU}} \in \mathcal{C}^{M\times(L+1)}$ requires $M(L+1)$ pilot symbols. Furthermore, to capture the pulse responses after each RIS update in practice, we either need to calculate the expected pulse offline (which requires channel estimation for the BS-UE link $\boldsymbol{h}_{\text{BU}}$), or reconfigure the RIS and obtain the pulse response from the receiver. For ARISE-SGD, more time steps would be required until convergence due to noisy gradients. In our baseline simulation setup with a 100-element RIS and a 21-tap pulse response, this corresponds to $100 \times 21 = 2100$ pilots that must be collected **before** ARISE can perform its first gradient update. For an episode consisting of 2000 RIS updates, ARISE therefore incurs a front-loaded pilot block that is comparable in size to the entire optimization horizon and must fit within the channel coherence time.

In contrast, the DRL algorithms require only a single measured pulse response per RIS update–2000 measurements in total if optimization lasts the entire episode–which can be conveyed via a low-rate out-of-band (OOB) link and eliminate the need for explicit CSI [65]. Crucially, DRL does not require any channel estimation before optimization

TABLE 2. Communicational overheads of the proposed algorithms.

Method	Required information per RIS update	Pilot overhead	Notes
ARISE	Full cascaded CSI $\boldsymbol{h}_{\mathrm{BRU}}$	$M(L+1)$ pilots	Must be reacquired at the start of each episode; overhead grows with RIS size and delay spread
DRL (DDPG/TD3/SAC)	Measured pulse response after RIS update	1 pulse measurement	Can be sent via low-rate OOB link; independent of RIS size

begins: it can start adapting the RIS immediately, using only the received pulse responses as feedback.

While the absolute overheads are similar in this particular configuration, the structural difference is significant: ARISE concentrates its overhead into a large pilot block that must be reacquired whenever the cascaded channel changes, whereas DRL distributes its measurements over time and remains agnostic to the underlying channel model. Moreover, the scaling behavior strongly favors DRL. For example, with a larger RIS (e.g., $M = 128$) and a shorter episode (e.g., 100 updates), ARISE would require $128 \times 21 = 2688$ pilots before the first gradient update, versus only 100 pulse measurements for DRL while optimizing the RIS, yielding more than an order-of-magnitude reduction in overhead. These examples illustrate that the reliance of ARISE on full cascaded CSI becomes increasingly impractical as RIS size grows or coherence time shortens, whereas DRL maintains consistent and scalable communication efficiency. Furthermore, the model-free operation enabled by DRL allows it to naturally accommodate RIS nonlinearities, coupling effects, and frequency-dependent behavior, making it a more scalable and robust approach for wideband RIS-aided equalization.

In summary, although ARISE is computationally lightweight in terms of its per-update gradient operations, its reliance on full cascaded CSI makes it increasingly impractical as RIS size grows or coherence time shortens. On the contrary, DRL maintains scalable, model-free operation with consistent communication efficiency. A summary of the communication overheads is provided in Table 2.

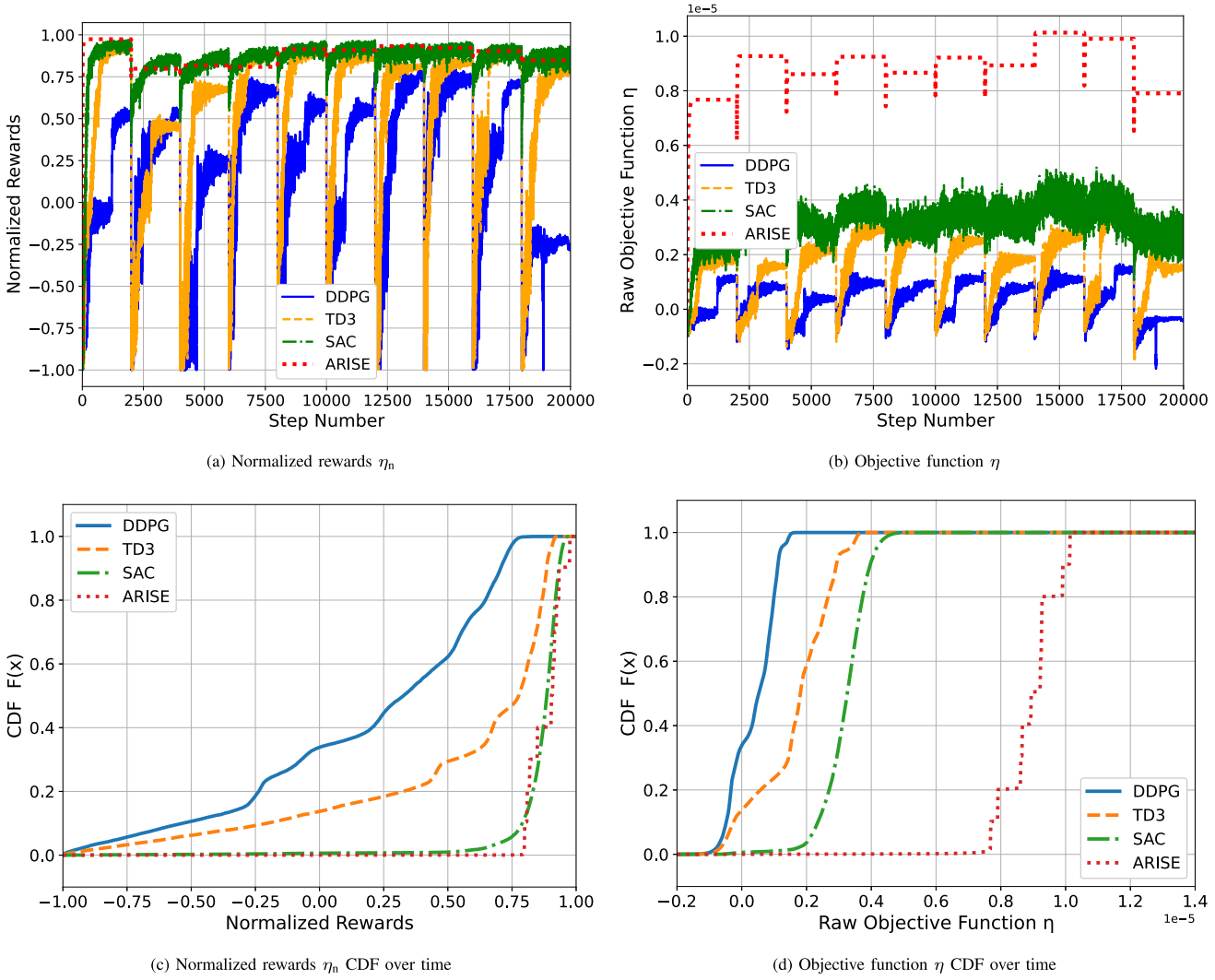
V. COMPARATIVE SIMULATION RESULTS

We use the same simulation setup as in Section III-B to compare the performance of the DRL algorithms with that of ARISE. Based on the results in Section III-B, we use a scaling factor $\alpha_{\mathrm{s}} = 0.10$ for the ARISE algorithm to enhance its equalization capability and provide a strong baseline to compare against the DRL algorithms. Here, we define an "episode" as a channel coherence block. At the beginning of each episode, the UE moves according to a random walk and the NLoS components are generated randomly, ensuring randomized channel realizations across episodes. A "time step" corresponds to a change in the RIS reflection coefficients, after which the resulting pulse response and the corresponding reward are obtained. Each episode consists of 2000 time steps. We use reward scaling by multiplying each r_{t+1} by 100 before saving it in the replay buffer to boost the learning rates of the networks. The replay buffers are updated in a first-in-first-out (FIFO) policy with a maximum buffer size $D_{\max}$. At the beginning of each episode, we reset the replay buffers and learning rates of all DNNs, as well as the SAC temperature coefficient α. Exclusively for DDPG, we also randomize the weights of the actor and critic networks to avoid remaining stuck at local optima.

TABLE 3. DRL implementation parameters.

Variable Name	Description	Value
N_{train}	Initial actor and critic updates per step	4
B	Batch size	4
$D_{\max}$	Max replay buffer size	400
d_{TD3}	TD3 actor update period	2
σ_{a}	DDPG and TD3 initial action noise standard deviation	0.3
τ_{d}	Noise decay rate	0.999
σ_{t}^2	TD3 training action standard deviation	0.3
c_{e}	TD3 training action noise bound	0.5
μ_{c}	Critic learning rate	10^{-3}
μ_{a}	Actor learning rate	DDPG: 10^{-3}; TD3 and SAC: 2×10^{-3}
μ_{α}	SAC α learning rate	10^{-3}
τ	Target network Polyak averaging factor	0.001

The noise added to the actions for DDPG and TD3 is scaled each time step by $10^{-\tau_{\mathrm{d}} t}$ to provide a decaying noise schedule which promotes exploration at the beginning of each episode, and more exploitation at the end of each episode to promote convergence stability, where t is the current time step index, $t = 0, 1, \ldots, 1999$, and resets at the beginning of each episode. Furthermore, we reduce the number of trainings per time step and actor learning rates depending on the average of the latest five consecutive rewards achieved by each algorithm. For instance, we start with $\mu_{\mathrm{a}} = 10^{-3}$ and $N_{\mathrm{train}} = 4$. When the average of five consecutive rewards are over 0.75, we set $\mu_{\mathrm{a}} \leftarrow 0.8\mu_{\mathrm{a}}$, $N_{\mathrm{train}} \leftarrow N_{\mathrm{train}} - 1$, and increase the reward target by 0.05. We repeat the same for rewards being over 0.8, then 0.85, etc., while setting the minimum bound for N_{train} to 1, to ensure the networks continue learning



(a) Normalized rewards η_{n}

(b) Objective function η

(c) Normalized rewards η_{n} CDF over time

(d) Objective function η CDF over time

FIGURE 12. Objective functions over time from a representative scenario, $M = 100, \kappa = 10, n_{\text{r}} = 10$.

while promoting stability and reducing training iterations at higher reward values. All the other relevant implementation parameters are shown in Table 3.

All actor and critic DNNs are implemented as shown in Fig. 9 with two hidden layers, with 512 nodes per hidden layer and ReLu activation functions [66]. Specifically for DDPG, we employ layer normalization after each hidden layer of the actor and critic networks to improve the training stability of the algorithm. All layers are initialized using the Random Normal initializer with zero mean and 0.1 standard deviation [67]. The networks are trained using TensorFlow Keras [54] in Python, using the Adam optimizer [68].

We first analyze the performance of the DRL algorithms compared with the ARISE algorithm for a stationary UE located at $(-20, -20)$, with the assumption of Rayleigh fading for the BS-UE link, Rician factor $\kappa = 10$ for the BS-RIS-UE link, and $n_{\text{r}} = 10$ delayed paths (such that $L = 20$ ISI components). The simulated scenario has 10 episodes where at the beginning of each episode, the NLoS channels change randomly. For fairness, we assume ARISE has access to the full cascaded CSI at the start of each episode, even though in practice this would require a dedicated channel estimation phase involving multiple RIS configurations before optimization could begin. The converged pulse responses at the end of the tenth episode of a single-run representative scenario are plotted in Fig. 10 and their corresponding QPSK constellations in Fig. 11. Additionally, we show the adaptation over time of the normalized rewards r_{t+1} and the objective function η from this same single-run representative trajectory in Fig. 12. The CDFs quantifying the reward values over the entire 10-episode simulation are also included in the same family of plots. We observe that the DRL algorithms all perform very differently from each other. SAC achieves equalization performance comparable and at times exceeding that of ARISE for the same channel coefficients, with very rapid convergence, taking less than 1000 steps at the beginning to converge to high reward values and staying at consistently high rewards over time afterwards,

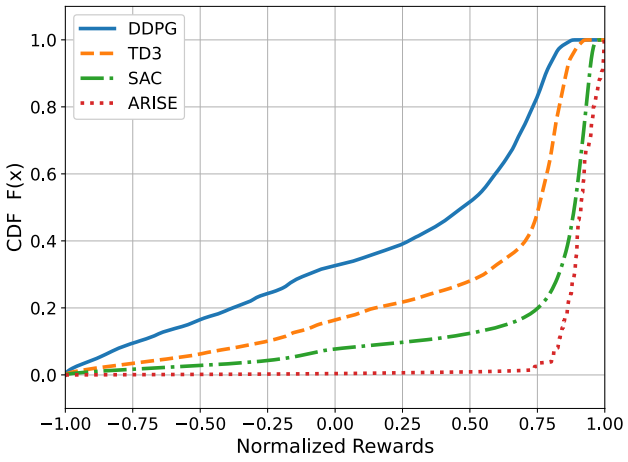


FIGURE 13. Normalized rewards η_n for each step over the duration of 100 episodes, $M = 100$, $\kappa = 10$, $n_r = 10$.

even when the channels are changing. Next, TD3 takes longer to converge than SAC, but seems to converge to high reward values after a few episodes and remain stable. DDPG is trailing behind and gets stuck at local optima, often unable to achieve the performance of TD3. Comparing the objective function η realized by each algorithm, ARISE consistently attains the highest objective mainly due to its amplification capabilities when attempting to match an attenuated signal with an amplified and equalized version of itself. The rewards defined for the DRL algorithms did not explicitly require signal amplification because they are normalized with respect to the sum of the absolute pulse response samples. On average, the RIS optimized using ARISE has around 3 dB more power gain than the RIS optimized using SAC, even though both achieve near-identical equalization capabilities for the given scenario. This is particularly notable because SAC attains ARISE-level temporal equalization without requiring the extensive pilot transmissions and full cascaded channel estimation that ARISE depends on, thereby reducing the communication overhead associated with RIS optimization.

Next, we employ the random walk model and compare the performance of each algorithm over more rigorous cases to evaluate their applicability to different RIS-based scenarios. In Fig. 13, we use the random walk model of the UE for the duration of 100 episodes and compare the normalized rewards η_n over time, still using the same RIS size and environment as in the previous case. Even though both the LoS and NLoS components change entirely at the beginning of each episode, SAC consistently maintains high and rapid equalization capabilities with normalized rewards higher than 0.75 for more than 80% of the time, approaching the performance of ARISE. Unlike ARISE, which would require repeated channel re-estimation as the UE moves, SAC adapts directly from received pulse responses, avoiding this additional communication burden. TD3 converges slower than SAC but maintains high rewards most of the time, while DDPG trails behind.

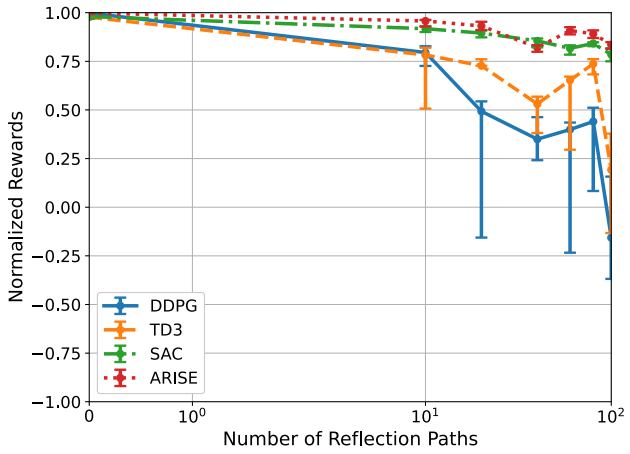


FIGURE 14. Converged normalized rewards η_n versus number of delayed paths n_r, $M = 100$, $\kappa = 10$.

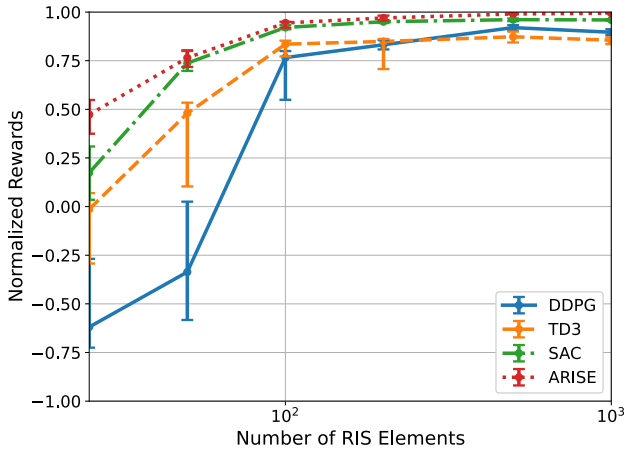


FIGURE 15. Converged normalized rewards η_n versus number of RIS elements M, $n_r = 10$, $\kappa = 10$.

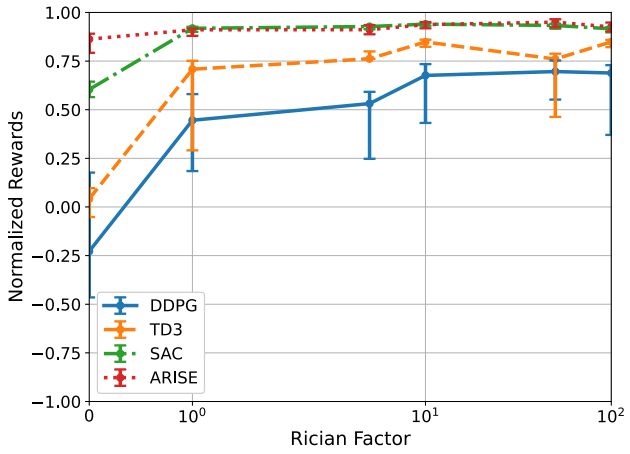


FIGURE 16. Converged normalized rewards η_n versus Rician factor κ, $M = 100$, $n_r = 10$.

In Figs. 14, 15, and 16, each data point represents the average normalized rewards from the last 10 steps of 10 consecutive channel realizations (episodes), starting with

random seeds for both the DNNs and the channel coefficients. The error bars show the standard deviations of the normalized rewards above and below the mean value. In Fig. 14, we compare the converged normalized rewards versus the number of delayed paths n_r, with a total number of ISI components given by $L = 2n_r$. SAC and ARISE still achieve similar performance with high stability, shown by the small standard deviations and high average normalized rewards, while TD3 and SAC have smaller rewards with SAC having more variance among episodes. Importantly, SAC maintains this performance without the need to estimate the increasing number of cascaded channel coefficients required by ARISE as the number of ISI components grows.

The overall performance of the 100-element RIS as an equalizer remains consistent as more ISI terms are introduced to the signal. We evaluate the equalization capability of the DRL algorithms as the RIS scales in Fig. 15. We note that modifying the number of RIS elements M would alter the output dimension of the actor DNN, while changing the number of delayed paths n_r may alter the input dimensions of the actor and critic DNNs (depending on the choice of the RIS operator to equalize a specific number of ISI terms). In our experiments, we modify the input and output dimensions of the actor and critic DNNs to correspond with each variable change, while keeping their hidden layers fixed. In the scenario in Fig. 15, we observe that when the RIS size is small, its equalization capability is considerably lower than the previous simulated cases, across all algorithms, with ARISE leading. As the RIS size grows, the algorithms systematically achieve greater performance and increasing stability. Additionally, ARISE continues to improve marginally as the number of RIS elements increases above 200, whereas the DRL algorithms seem to plateau. This is most likely a consequence of the hidden layers of the DNNs maintaining the same sizes, struggling to adhere to the increasing number of random channels while being able to generalize enough to eliminate most of the ISI.

Finally, we assess the performance of each algorithm with respect to the Rician factor of the BS-RIS-UE main tap link, κ, in Fig. 16. For the Rayleigh case, ARISE attains a considerably better performance than the DRL algorithms with $\eta_n \approx 0.85$, with DDPG being completely unable to equalize the signal, TD3 slightly better, and SAC consistently achieving close to an average value of $\eta_n \approx 0.6$. As the Rician factor increases and the DNNs are able to learn the geometry of the LoS components of the wireless channels, their performance improves significantly, with SAC achieving the same equalization performance as ARISE. The performance parity is achieved despite SAC operating without explicit CSI acquisition and RIS model, further underscoring its advantage in communication efficiency.

We overall demonstrate the ability of the DRL algorithms to successfully configure the RIS as an equalizer, with SAC achieving the same performance as ARISE in most cases and a comparable performance in others. Though requiring a higher per-step computational cost than TD3 and DDPG, it provides a means to quickly and efficiently boost the signal integrity of the link without requiring excessive channel estimation. In SAC, the incorporation of an entropy-regularization term fundamentally alters the optimization dynamics by explicitly encouraging high-entropy policies to induce exploration. This mechanism mitigates premature convergence and reduces the susceptibility to suboptimal deterministic behaviors commonly observed in TD3 and DDPG, both of which rely on the external addition of noise to learn the action space. The resulting exploration incentive, coupled with the reward signal from the environment, enables SAC to more effectively discover advantageous regions of the continuous action space during training. Moreover, entropy regularization smooths the underlying optimization landscape, promoting policies that represent distributions over effective actions rather than collapsing to narrow optima. Combining the stochastic policy formulation and dual-critic architecture in SAC yields improved robustness, increased stability, enhanced sample efficiency, and superior performance in complex continuous-control environments. Taken together, these results show that SAC quickly achieves ARISE-level equalization accuracy while incurring substantially lower communication complexity and only moderate computational overhead, making it a more scalable solution for practical RIS deployments.

VI. CONCLUSION

In this work, we demonstrated the utility of RIS technology as both a signal booster (spatial beamformer) and a temporal equalizer across diverse channel conditions and wireless environments. We developed a steepest descent-based approach–ARISE–that leverages the estimated cascaded BS-RIS-UE channels and showed its ability to configure the RIS coefficients to equalize the received pulse response and enhance the SNR. We also examined the challenges of applying ARISE to passive RISs, particularly the extensive channel estimation required for accurate equalization and the resulting tradeoff between signal gain and equalization performance–an effect that may necessitate iterative optimization or deeper SNR analysis.

As an alternative, we explored several DRL methods–DDPG, TD3, and SAC–that operate without channel estimation and rely solely on the received pulse response to optimize the RIS coefficients, enabling faster adaptation and reduced latency in dynamic environments. Through comparative analysis of computational complexity and achievable performance, we found SAC to be the most effective, converging rapidly to stable solutions across varying channels and RIS sizes, and achieving performance comparable to ARISE. Crucially, SAC attains this performance without the heavy channel estimation requirements of ARISE by reconfiguring the RIS immediately from raw pulse response observations, resulting in a significantly reduced communication complexity and a more practical overall optimization pipeline. Its model-free nature makes SAC inherently robust to RIS nonlinearities,

element coupling, and frequency-dependent reflection behavior, which are difficult to capture in analytical channel models. Overall, DRL emerges as a promising and potentially integral component of future RIS-assisted wireless systems due to its implementation simplicity, robustness, and strong performance.

The proposed RIS-based equalization is particularly well-suited to indoor industrial environments and high-frequency (mmWave/THz) backhaul links, where large bandwidths and multipath propagation result in pronounced channel frequency selectivity and ISI. In these settings, implementing equalization solely at the receiver can be costly and noise-enhancing, while transmitter-side precoding may be limited by CSI acquisition overhead. By contrast, an RIS can reshape the propagation channel itself through passive phase control, enabling over-the-air equalization that reduces receiver complexity and improves robustness under practical constraints. These are *representative* scenarios where delay spreads on the order of tens of nanoseconds over multi-GHz bandwidths lead to significant ISI and thus where benefits are clear.

Finally, we note that the proposed RIS-based over-the-air equalization does not alter the fundamental information-theoretic capacity of the channel. Because the RIS is a passive structure, it introduces no additional power or degrees of freedom; rather, it reshapes the existing propagation environment. By coherently combining multipath components and suppressing detrimental delayed paths, the RIS modifies the effective channel impulse response, enabling improved achievable performance under practical receiver and CSI constraints without altering the underlying capacity laws.

Following the contributions of this study, future work should analyze the performance of ARISE and the DRL-based optimization in the presence of multiple users, micro-scale UE movements and Doppler spread effects, and multiple-antenna base stations, and extend the RIS architecture from a ULA to a uniform planar array (UPA). Incorporating physical RIS constraints with hardware restrictions such as frequency-dependent reflection coefficients, mutual coupling between elements, and nonlinearities will further refine the concept of RIS-based temporal and spatial equalization, and broaden its applicability in practical multiple-input multiple-output (MIMO) systems with realistic hardware and propagation conditions.

APPENDIX A
ARISE ALGORITHM DERIVATION

We wish to minimize the mean squared error (MSE) between a desired signal $\boldsymbol{s}$ and the received signal $\boldsymbol{y}$

$$\boldsymbol{y} = \left(\boldsymbol{h}_{\mathrm{BU}} + \sum_{m=1}^{M} \Gamma_m \boldsymbol{h}_{\mathrm{BRU},m}\right) * \boldsymbol{s} \tag{35}$$

via gradient descent on the RIS reflection coefficients Γ_m for $m = 1, 2, \ldots, M$. The cost function is defined by $J = E[\epsilon_k \epsilon_k^*]$ with error signal $\boldsymbol{\epsilon} = \boldsymbol{s} - \boldsymbol{y}$, where the expectation is taken over all desired sample indices k under the assumption of ergodicity. Here, $\boldsymbol{s} = [1, 0, \ldots, 0]$ is the transmitted discrete-time pulse. We can express Γ_m as the composition of its real and imaginary parts

$$\Gamma_m = a_m + jb_m, \tag{36}$$

thus expressing $\boldsymbol{y}$ as

$$\boldsymbol{y} = \left(\boldsymbol{h}_{\mathrm{BU}} + \sum_{m=1}^{M} (a_m + jb_m)\boldsymbol{h}_{\mathrm{BRU},m}\right) * \boldsymbol{s}. \tag{37}$$

Algorithm 3 Twin Delayed DDPG (TD3) for RIS Equalizer

Initialize both training and target DNNs for both actor and critic, assign $\phi_{\mathrm{targ}} \leftarrow \phi, \quad \theta_{\mathrm{targ}} \leftarrow \theta$.
for episode in N_{ep} **do**
  for t in N_{steps} **do**
    Observe state $s_t \leftarrow \boldsymbol{y}'_t$ and set RIS coefficients

$$\boldsymbol{\Gamma}'_t \leftarrow \mathrm{clip}(\mu_{\phi_{\mathrm{targ}}}(\boldsymbol{y}'_t) + \boldsymbol{\epsilon}, -1, 1),$$
$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_{\mathrm{a}}^2)^{2M\times 1}.$$

    Set $a_t \leftarrow \boldsymbol{\Gamma}'_t$.
    Update RIS reflection coefficients as

$$\boldsymbol{\Gamma}_t \leftarrow [\Gamma'_{t,1} + j\Gamma'_{t,2}, \ldots, \Gamma'_{t,2M-1} + j\Gamma'_{t,2M}]^T$$

    and normalize using (16).
    Observe new state $s_{t+1} \leftarrow \boldsymbol{y}'_{t+1}$.
    Calculate rewards r_{t+1} using (21).
    Store $(s_t, a_t, r_{t+1}, s_{t+1})$ in the replay buffer $\mathcal{D}$.
    if $|\mathcal{D}| \geq B$ **then**
      for i in N_{train} **do**
        Sample batch $\mathcal{B} = \{(s, a, r', s')\}$ from $\mathcal{D}$.
        Add noise to a:

$$a \leftarrow \mathrm{clip}(a + \mathrm{clip}(\epsilon, -c_e, c_e), -1, 1)$$
$$\epsilon \sim \mathcal{N}(0, \sigma_{\mathrm{t}}^2).$$

        Calculate $l(\theta_i)$ using (30) for $i = 1, 2$.
        Update critics:

$$\theta_i \leftarrow \theta_i - \mu_{\mathrm{c}} \nabla_{\theta_i} l(\theta_i).$$

        Calculate $l(\phi)$ using (28).
        if $i \mod d_{\mathrm{p}} == 0$ **then**
        Update actor:

$$\phi \leftarrow \phi - \mu_{\mathrm{a}} \nabla_{\phi} l(\phi).$$

        Soft-update target critics:

$$\theta_{\mathrm{targ},i} \leftarrow \tau\theta_i + (1-\tau)\theta_{\mathrm{targ},i}, \quad i = 1, 2.$$

        Soft-update target actor:

$$\phi_{\mathrm{targ}} \leftarrow \tau\phi + (1-\tau)\phi_{\mathrm{targ}}.$$

      end if
    end for
  end if
end for
end for

Algorithm 4 Soft Actor-Critic (SAC) for RIS Equalizer

Initialize actor DNN hyperparameters ϕ, training and target critic DNNs, assign $\theta_{\text{targ}} \leftarrow \theta$.
Initialize temperature parameter α.
for episode in N_{ep} **do**
  for t in N_{steps} **do**
    Observe state $s_t \leftarrow \boldsymbol{y}'_t$.
    Sample action from stochastic policy:

$$a_t \sim \pi_\phi(a_t \mid s_t).$$

    Clip RIS coefficients: $\boldsymbol{\Gamma}'_t \leftarrow \text{clip}(a_t, -1, 1)$.
    Update RIS reflection coefficients as

$$\boldsymbol{\Gamma}_t \leftarrow [\Gamma'_{t,1} + j\Gamma'_{t,2}, \ldots, \Gamma'_{t,2M-1} + j\Gamma'_{t,2M}]^T$$

    and normalize using (16).
    Observe new state $s_{t+1} \leftarrow \boldsymbol{y}'_{t+1}$.
    Calculate rewards r_{t+1} using (21).
    Store $(s_t, a_t, r_{t+1}, s_{t+1})$ in replay buffer $\mathcal{D}$.
    if $|\mathcal{D}| \geq B$ **then**
      for i in N_{train} **do**
        Sample batch $\mathcal{B} = \{(s, a, r', s')\}$ from $\mathcal{D}$.
        Sample next actions and log-probabilities:

$$a' \sim \pi_\phi(a' \mid s'), \quad \log \pi_\phi(a' \mid s').$$

        Calculate $l(\theta_i)$ using (32) for $i = 1, 2$.
        Update critics:

$$\theta_i \leftarrow \theta_i - \mu_{\text{c}} \nabla_{\theta_i} l(\theta_i).$$

        Sample actions for actor update:

$$a \sim \pi_\phi(a \mid s), \quad \log \pi_\phi(a \mid s).$$

        Calculate $l(\phi)$ using (33).
        Update actor:

$$\phi \leftarrow \phi - \mu_{\text{a}} \nabla_\phi l(\phi).$$

        Calculate $l(\log \alpha)$ using (34).
        Update temperature:

$$\log \alpha \leftarrow \log \alpha - \mu_\alpha \nabla_{\log \alpha} l(\log \alpha).$$

        $\alpha \leftarrow \exp(\log \alpha)$.
        Soft-update target critics:

$$\theta_{\text{targ},i} \leftarrow \tau\theta_i + (1 - \tau)\theta_{\text{targ},i}, \quad i = 1, 2.$$

      end for
    end if
  end for
end for

The error gradient with respect to Γ_m is defined as

$$\frac{\partial \boldsymbol{\epsilon}}{\partial \Gamma_m} = \frac{\partial \boldsymbol{\epsilon}}{\partial a_m} + j\frac{\partial \boldsymbol{\epsilon}}{\partial b_m} \tag{38}$$

and is related to $\boldsymbol{y}$ by

$$\frac{\partial \boldsymbol{\epsilon}}{\partial a_m} = -\frac{\partial \boldsymbol{y}}{\partial a_m}; \quad \frac{\partial \boldsymbol{\epsilon}}{\partial b_m} = -\frac{\partial \boldsymbol{y}}{\partial b_m}. \tag{39}$$

Expanding the convolution operator and considering each term in $\boldsymbol{y}$, y_k, we obtain

$$\frac{\partial y_k}{\partial a_m} = \sum_{l=0}^{L} h_{\text{BRU},m,l} s_{k-l} \tag{40}$$

and

$$\frac{\partial y_k}{\partial b_m} = j\sum_{l=0}^{L} h_{\text{BRU},m,l} s_{k-l}. \tag{41}$$

Differentiating the cost function yields

$$\begin{aligned}\frac{\partial J}{\partial a_m} &= E\left[\frac{\partial \epsilon_k}{\partial a_m}\epsilon_k^* + \frac{\partial \epsilon_k^*}{\partial a_m}\epsilon_k\right] \\ &= E\left[-\sum_{l=0}^{L} h_{\text{BRU},m,l} s_{k-l}\epsilon_k^* - \sum_{l=0}^{L} h_{\text{BRU},m,l}^* s_{k-l}^* \epsilon_k\right]\end{aligned} \tag{42}$$

and

$$\begin{aligned}\frac{\partial J}{\partial b_m} &= E\left[\frac{\partial \epsilon_k}{\partial b_m}\epsilon_k^* + \frac{\partial \epsilon_k^*}{\partial b_m}\epsilon_k\right] \\ &= E\left[-j\sum_{l=0}^{L} h_{\text{BRU},m,l} s_{k-l}\epsilon_k^* + j\sum_{l=0}^{L} h_{\text{BRU},m,l}^* s_{k-l}^* \epsilon_k\right].\end{aligned} \tag{43}$$

Combining, we get

$$\frac{\partial J}{\partial \Gamma_m} = E\left[-2\sum_{l=0}^{L} h_{\text{BRU},m,l}^* s_{k-l}^* \epsilon_k\right] \tag{44}$$

and the final gradient is

$$\nabla_{\boldsymbol{\Gamma}} = -2E\begin{bmatrix} \sum_{l=0}^{L} h_{\text{BRU},1,l}^* s_{k-l}^* \epsilon_k \\ \sum_{l=0}^{L} h_{\text{BRU},2,l}^* s_{k-l}^* \epsilon_k \\ \vdots \\ \sum_{l=0}^{L} h_{\text{BRU},M,l}^* s_{k-l}^* \epsilon_k \end{bmatrix}. \tag{45}$$

Finally, we have the update rule for $\boldsymbol{\Gamma}$ given by gradient descent

$$\begin{aligned}\boldsymbol{\Gamma}_{t+1} &= \boldsymbol{\Gamma}_t - \frac{1}{2}\mu\nabla_{\boldsymbol{\Gamma}_t} \\ &= \boldsymbol{\Gamma}_t + \mu E\left[\sum_{l=0}^{L} \boldsymbol{h}_{\text{BRU},l}^* s_{k-l}^* \epsilon_k\right],\end{aligned} \tag{46}$$

where μ is the step size. To accommodate the passivity requirement of the RIS, we normalize the magnitudes of the reflection coefficients after every update by

$$\Gamma_{t+1,m} \leftarrow \frac{\Gamma_{t+1,m}}{|\Gamma_{t+1,m}|} \tag{47}$$

for all $m = 1, 2, \ldots, M$.

APPENDIX B
TD3 AND SAC ALGORITHMS PSEUDOCODES

The pseudocode for the TD3 algorithm is given as Algorithm 3 and the pseudocode for the SAC algorithm is given as Algorithm 4 in this appendix.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive and insightful comments, which have significantly improved the quality and clarity of this article.

REFERENCES

[1] C. Pan et al., "Reconfigurable intelligent surfaces for 6G systems: Principles, applications, and research directions," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 14–20, Jun. 2021.

[2] M. A. ElMossallamy, H. Zhang, L. Song, K. G. Seddik, Z. Han, and G. Y. Li, "Reconfigurable intelligent surfaces for wireless communications: Principles, challenges, and opportunities," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 3, pp. 990–1002, Sep. 2020.

[3] C. Pan et al., "An overview of signal processing techniques for RIS/IRS-aided wireless systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 883–917, Aug. 2022.

[4] X. Wei, D. Shen, and L. Dai, "Channel estimation for RIS assisted wireless communications—Part I: Fundamentals, solutions, and future opportunities," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1398–1402, May 2021.

[5] H. Zhou, M. Erol-Kantarci, Y. Liu, and H. V. Poor, "A survey on model-based, heuristic, and machine learning optimization approaches in RIS-aided wireless networks," 2023, *arXiv:2303.14320*.

[6] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 8, pp. 4157–4170, Aug. 2019.

[7] M. Di Renzo et al., "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020.

[8] G. Zhou, C. Pan, H. Ren, K. Wang, and A. Nallanathan, "A framework of robust transmission design for IRS-aided MISO communications with imperfect cascaded channels," *IEEE Trans. Signal Process.*, vol. 68, pp. 5092–5106, 2020.

[9] X. Pei et al., "RIS-aided wireless communications: Prototyping, adaptive beamforming, and indoor/outdoor field trials," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8627–8640, Dec. 2021.

[10] R. Liu, M. Li, Y. Liu, Q. Wu, and Q. Liu, "Joint transmit waveform and passive beamforming design for RIS-aided DFRC systems," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 995–1010, Aug. 2022.

[11] G. Ben-Itzhak, M. Saavedra-Melo, B. Bradshaw, E. Ayanoglu, F. Capolino, and A. Lee Swindlehurst, "Design and operation principles of a wave-controlled reconfigurable intelligent surface," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7730–7751, 2024.

[12] Y. Liu et al., "Reconfigurable intelligent surfaces: Principles and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1546–1577, 3rd Quart., 2021.

[13] H. Guo, Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3064–3076, May 2020.

[14] X. Mu, Y. Liu, L. Guo, J. Lin, and R. Schober, "Simultaneously transmitting and reflecting (STAR) RIS aided wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 5, pp. 3083–3098, May 2022.

[15] Z. Yu et al., "Active RIS-aided ISAC systems: Beamforming design and performance analysis," *IEEE Trans. Commun.*, vol. 72, no. 3, pp. 1578–1595, Mar. 2024.

[16] B. Di, H. Zhang, L. Song, Y. Li, Z. Han, and H. V. Poor, "Hybrid beamforming for reconfigurable intelligent surface based multi-user communications: Achievable rates with limited discrete phase shifts," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1809–1822, Aug. 2020.

[17] J. He, H. Wymeersch, and M. Juntti, "Channel estimation for RIS-aided mmWave MIMO systems via atomic norm minimization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5786–5797, Sep. 2021.

[18] X. Wei, D. Shen, and L. Dai, "Channel estimation for RIS assisted wireless communications—Part II: An improved solution based on double-structured sparsity," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1403–1407, May 2021.

[19] Y. Xing and T. S. Rappaport, "Millimeter wave and terahertz urban microcell propagation measurements and models," *IEEE Commun. Lett.*, vol. 25, no. 12, pp. 3755–3759, Dec. 2021.

[20] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[21] H. Zhang, L. Song, Z. Han, and H. V. Poor, "Spatial equalization before reception: Reconfigurable intelligent surfaces for multi-path mitigation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 8062–8066.

[22] H. Prod'homme, M. F. Imani, S. Abadal, and P. del Hougne, "RIS-based over-the-air channel equalization in resource-constrained wireless networks," in *Proc. 18th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2024, pp. 1–5.

[23] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1839–1850, Aug. 2020.

[24] B. Saglam, D. Gurgunoglu, and S. S. Kozat, "Deep reinforcement learning based joint downlink beamforming and RIS configuration in RIS-aided MU-MISO systems under hardware impairments and imperfect CSI," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2023, pp. 66–72.

[25] K. Kim, Y. K. Tun, M. S. Munir, W. Saad, and C. S. Hong, "Deep reinforcement learning for channel estimation in RIS-aided wireless networks," *IEEE Commun. Lett.*, vol. 27, no. 8, pp. 2053–2057, Aug. 2023.

[26] P.-H. Chou, B.-R. Zheng, W.-J. Huang, W. Saad, Y. Tsao, and R. Y. Chang, "Deep reinforcement learning-based precoding for multi-RIS-aided multiuser downlink systems with practical phase shift," *IEEE Wireless Commun. Lett.*, vol. 14, no. 1, pp. 23–27, Jan. 2025.

[27] M.-L. Tham, Y. J. Wong, A. Iqbal, N. B. Ramli, Y. Zhu, and T. Dagiuklas, "Deep reinforcement learning for secrecy energy- efficient UAV communication with reconfigurable intelligent surface," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2023, pp. 1–6.

[28] Q. Li, M. El-Hajjar, C. Xu, J. An, C. Yuen, and L. Hanzo, "Stacked intelligent metasurfaces for holographic MIMO-aided cell-free networks," *IEEE Trans. Commun.*, vol. 72, no. 11, pp. 7139–7151, Nov. 2024.

[29] Q. Li, M. El-Hajjar, K. Cao, C. Xu, H. Haas, and L. Hanzo, "Holographic metasurface-based beamforming for multi-altitude LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 24, no. 4, pp. 3103–3116, Apr. 2025.

[30] Q. Li, M. El-Hajjar, Y. Sun, and L. Hanzo, "Performance analysis of reconfigurable holographic surfaces in the near-field scenario of cell-free networks under hardware impairments," *IEEE Trans. Wireless Commun.*, vol. 23, no. 9, pp. 11972–11984, Sep. 2024.

[31] R. Wang, Y. Yang, B. Makki, and A. Shamim, "A wideband reconfigurable intelligent surface for 5G millimeter-wave applications," *IEEE Trans. Antennas Propag.*, vol. 72, no. 3, pp. 2399–2410, Mar. 2024.

[32] E. K. Hidir, E. Basar, and H. A. Cirpan, "On practical RIS-aided OFDM with index modulation," *IEEE Access*, vol. 11, pp. 13113–13120, 2023.

[33] K. D. Katsanos, N. Shlezinger, M. F. Imani, and G. C. Alexandropoulos, "Wideband multi-user MIMO communications with frequency selective RISs: Element response modeling and sum-rate maximization," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2022, pp. 151–156.

[34] A. L. Swindlehurst, G. Zhou, R. Liu, C. Pan, and M. Li, "Channel estimation with reconfigurable intelligent surfaces—A general framework," *Proc. IEEE*, vol. 110, no. 9, pp. 1312–1338, Sep. 2022.

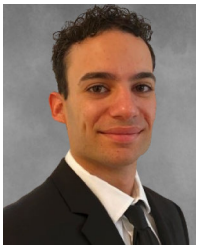
[35] J. Tapie, H. Prod'homme, M. F. Imani, and P. D. Hougne, "Systematic physics-compliant analysis of over-the-air channel equalization in RIS-parametrized wireless networks-on-chip," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 8, pp. 2026–2038, Aug. 2024.

[36] T. S. Rappaport, G. R. MacCartney, M. K. Samimi, and S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," *IEEE Trans. Commun.*, vol. 63, no. 9, pp. 3029–3056, Sep. 2015.

[37] A. A. M. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Sel. Areas Commun.*, vol. SAC-5, no. 2, pp. 128–137, Feb. 1987.

[38] S. Zeng, H. Zhang, B. Di, Z. Han, and L. Song, "Reconfigurable intelligent surface (RIS) assisted wireless coverage extension: RIS orientation and location optimization," *IEEE Commun. Lett.*, vol. 25, no. 1, pp. 269–273, Jan. 2021.

[39] H. Zhang, B. Di, L. Song, and Z. Han, "Reconfigurable intelligent surfaces assisted communications with limited phase shifts: How many phase shifts are enough?" *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4498–4502, Apr. 2020.
[40] E. Björnson and L. Sanguinetti, "Rayleigh fading modeling and channel hardening for reconfigurable intelligent surfaces," *IEEE Wireless Commun. Lett.*, vol. 10, no. 4, pp. 830–834, Apr. 2021.
[41] Z. Abdullah, A. Papazafeiropoulos, S. Kisseleff, S. Chatzinotas, and B. Ottersten, "Impact of phase-noise and spatial correlation on double-RIS-assisted multiuser MISO networks," *IEEE Wireless Commun. Lett.*, vol. 11, no. 7, pp. 1473–1477, Jul. 2022.
[42] A. H. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley, 2011.
[43] S. Qureshi, "Adaptive Equalization," *Proc. IEEE*, vol. 73, no. 9, pp. 1349–1387, 1985.
[44] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, 1976.
[45] *NR; Physical Channels and Modulation*, document TS 38.211, 3GPP, 2024.
[46] M. Jian et al., "Reconfigurable intelligent surfaces for wireless communications: Overview of hardware designs, channel models, and estimation techniques," *Intell. Converged Netw.*, vol. 3, no. 1, pp. 1–32, Mar. 2022.
[47] G. Ben-Itzhak, M. Saavedra-Melo, E. Ayanoglu, F. Capolino, and A. L. Swindlehurst, "AI-driven optimization of wave-controlled reconfigurable intelligent surfaces," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 6650–6665, 2025.
[48] K. Keykhosravi, M. F. Keskin, G. Seco-Granados, P. Popovski, and H. Wymeersch, "RIS-enabled SISO localization under user mobility and spatial-wideband effects," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 1125–1140, Aug. 2022.
[49] X. Zhao, M. Jian, Y. Chen, Y. Zhao, and L. Mu, "Reconfigurable intelligent surfaces for 6G: Engineering challenges and the road ahead," *Intell. Converged Netw.*, vol. 6, no. 1, pp. 53–81, Mar. 2025.
[50] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.
[51] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[52] D. Silver et al., "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 387–395.
[53] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
[54] M. Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: https://www.tensorflow.org/
[55] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[56] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.
[57] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, Jul. 1992.
[58] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018, *arXiv:1802.09477*.
[59] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018, *arXiv:1801.01290*.
[60] T. Haarnoja et al., "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.
[61] H. Liu, X. Yuan, and Y.-J.-A. Zhang, "Matrix-calibration-based cascaded channel estimation for reconfigurable intelligent surface assisted multiuser MIMO," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2621–2636, Nov. 2020.
[62] N. K. Kundu and M. R. McKay, "Channel estimation for reconfigurable intelligent surface aided MISO communications: From LMMSE to deep learning solutions," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 471–487, 2021.
[63] J. An, C. Xu, L. Gan, and L. Hanzo, "Low-complexity channel estimation and passive beamforming for RIS-assisted MIMO systems relying on discrete phase shifts," *IEEE Trans. Commun.*, vol. 70, no. 2, pp. 1245–1260, Feb. 2022.
[64] H. Guo and V. K. N. Lau, "Uplink cascaded channel estimation for intelligent reflecting surface assisted multiuser MISO systems," *IEEE Trans. Signal Process.*, vol. 70, pp. 3964–3977, 2022.
[65] A. Ali, N. González-Prelcic, and R. W. Heath Jr., "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1038–1052, Feb. 2018.
[66] B. Ding, H. Qian, and J. Zhou, "Activation functions and their characteristics in deep neural networks," in *Proc. Chin. Control Decis. Conf. (CCDC)*, 2018, pp. 1836–1841.
[67] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone, "A review on weight initialization strategies for neural networks," *Artif. Intell. Rev.*, vol. 55, no. 1, pp. 291–322, Jan. 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-10033-z
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.



GAL BEN-ITZHAK (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of California, Irvine (UCI), Irvine, CA, USA, in 2023 and 2024, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research and professional interests include high-speed communications, optimization, digital signal processing, high-speed circuit design, and signal integrity.

ENDER AYANOGLU (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1986. He was with the Communications Systems Research Laboratory, Bell Laboratories, Holmdel, NJ, USA. From 1999 to 2002, he was a Systems Architect with Cisco Systems Inc., San Jose, CA, USA. Since 2002, he has been a Professor with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA, USA, where he was the Director of the Center for Pervasive Communications and Computing and the Conexant-Broadcom Endowed Chair from 2002 to 2010. He was a recipient of the IEEE Communications Society Stephen O. Rice Prize Paper Award in 1995, the IEEE Communications Society Best Tutorial Paper Award in 1997, and the IEEE Communications Society Communication Theory Technical Committee Outstanding Service Award in 2014. From 2000 to 2001, he was the Founding Chair of the IEEE-ISTO Broadband Wireless Internet Forum, an industry standards organization. He served on the Executive Committee for the IEEE Communications Society Communication Theory Committee from 1990 to 2002 and the Chair from 1999 to 2002. From 1993 to 2014, he was an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He was the Editor-in-Chief of IEEE TRANSACTIONS ON COMMUNICATIONS from 2004 to 2008 and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS-Series on Green Communications and Networking from 2014 to 2016. He was the Founding Editor-in-Chief of IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING from 2016 to 2020. He served as an IEEE Communications Society Distinguished Lecturer for two terms in 2022 and 2025 and serving a third term in 2026 and 2027. In 2023, he received the IEEE Communications Society Joseph L. LoCicero Award for outstanding contributions to IEEE Communications Society journals as an Editor, the Editor-in-Chief (EiC), and the Founding EiC.