Actor-Critic Methods for IRS Design in Correlated Channel Environments: A Closer Look Into the Neural Tangent Kernel of the Critic

Spilios Evmorfos, Athina P. Petropulu, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*—The article studies the design of an Intelligent Reflecting Surface (IRS) in order to support a Multiple-Input-Single-Output (MISO) communication system operating in a mobile, spatiotemporally correlated channel environment. The design objective is to maximize the expected sum of Signal-to-Noise Ratio (SNR) at the receiver over an infinite time horizon. The problem formulation gives rise to a Markov Decision Process (MDP). We propose an actor-critic algorithm for continuous control that accounts for both channel correlations and destination motion by constructing the state of the Reinforcement Learning algorithm to include history of destination positions and IRS phases. To account for the variability of the underlying value function, arising due to the channel variability, we propose to pre-process the input of the critic with a Fourier kernel, which enables stability in the process of neural value approximation. We also examine the use of the destination SNR as a component of the designed MDP state, which constitutes common practice in previous works. We empirically show that, when the channels are spatiotemporally varying, including the SNR in the state representation causes divergence. We provide insight on the aforementioned divergence by demonstrating the effect of the SNR inclusion on the Neural Tangent Kernel of the critic network. Based on our study, we propose a framework for designing actor-critic methods for IRS design and also for more general problems, that is predicated upon sufficient conditions of the critic's Neural Tangent Kernel for convergence under neural value learning.**

***Index Terms*—Intelligent Reflecting Surfaces, deep learning, reinforcement learning, IRS parameter design, Neural Tangent Kernels.**

I. Introduction

THE field of Intelligent Reflecting Surfaces (IRSs) [1], [2], [3] is a recent development at the intersection of devices, signal processing and wireless communications. An IRS is a panel of reflective elements. Each element can be independently controlled to change the phase of the impinging wave. By deploying the IRS in a way to interact with a transmitted communication signal, and by dynamically controlling the IRS elements one can create a smart propagation environment towards the destination, counteracting the effects of attenuation, improving Quality-of-Service (QoS) [4], security [5], efficiency [6], [7] and energy preservation [8], [9], [10].

For the promise of IRSs to be realized, however, a number of research questions need to be answered, mainly related to the complexity of the IRS design problem. Designing the IRS elements to optimize some performance objective typically gives rise to highdimensional nonconvex optimization objectives that pose significant challenges, especially in the case of dynamic environments. While semidefinite relaxations to these problems can achieve good performance [8], [11], [12], they entail significant computational complexity.

Recent developments in deep learning [13] have provided data-driven, reliable and scalable solutions to highly nonconvex problems. Thus, it comes as no surprise that research efforts have been focused on adopting deep learning paradigms for problems that revolve around the designing of the IRS parameters. Training graph neural networks to learn the mappings from channel pilots to IRS elements in a supervised fashion was considered in [14]. In [15], [16], the problem of IRS design is viewed under the prism of supervised learning. A feedforward neural network is employed in [15] to parametrize the mapping from coordinate positions of the mobile receivers to IRS element configurations in order to maximize the received signal strength. In [16], a sparse sensing approach based on deep generative modeling is used to estimate the channels for all IRS elements. The estimated channels are then used to design the IRS parameter values.

Supervised learning methods pose certain distinct drawbacks when it comes to IRS design; most prominently, they require data instances that have been manually labeled or annotated by human experts. These labels serve as references or correct answers. These are typically hard or expensive to obtain. Further, with the exception of supervised learning for time series forecasting [17], deep supervised learning assume that training data (channel measurements) are independent and identically distributed (i.i.d.) samples from a channel distribution. However, this assumption does not hold for

Manuscript received 9 January 2023; revised 17 June 2023 and 13 August 2023; accepted 25 September 2023. Date of publication 13 October 2023; date of current version 7 November 2023. This work was supported in part by the ARO under Grants W911NF2110071 and W911NF2320103, and in part by the NSF under Grants ECCS- 2033433, ECCS-2320568, CNS-2128448, and ECCS-2335876. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ali Tajer. *(Corresponding author: Spilios Evmorfos.)*

Spilios Evmorfos and Athina P. Petropulu are with the Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08854 USA (e-mail: se386@scarletmail.rutgers.edu; athinap@soe.rutgers.edu)

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu)

Digital Object Identifier 10.1109/TSP.2023.3322830

urban channel communication environments, where the shadowing effect gives rise to spatiotemporal correlations [18], [19]. Such urban environments constitute an attractive option for IRS deployment.

The limitations of supervised learning can be overcome via the paradigm of deep Reinforcement Learning (RL) [20]. RL methods do not require ground truth labels. The only stipulation is for a scalar reward signal, the role of which can be played by the QoS metric that is being optimized, for example, the destination SNR in a communication scenario. RL methods do not require i.i.d channel measurements.

Therefore, RL methods are more suitable for controlling the IRS elements over time in settings in which the channels are spatiotemporally correlated. Such a setting is adopted in this study. Deep RL methods for designing IRS-assisted communication systems have been considered in [21], [22], [23]. The work in [24] investigates a typical communication scenario between a Multiple-Input-Multiple-Output (MIMO) source and a mobile destination, assisted by a single IRS (very similar to the one we investigate in the current work). The authors benchmark the performance of a simple deep RL algorithm against multiple naturally applicable methods, such as a Vector Approximate Message Passing (VAMP) [25] and Alternating Direction Method of Multipliers (ADMM) [26]. They demonstrate that, under highly noisy channels, the deep RL method is significantly more robust with respect to the noise level. This constitutes a strong motivation for the current work since it validates the need for improved deep RL algorithms for IRS design. The work of [27] presents a deep RL algorithm intertwined with a convex approximation lower bound formulation, where the IRS parameters and the source precoding weights are selected with the goal of minimizing the source transmit power subject to meeting constraints on the receiver's SNR and the IRS power budget. Deep RL for IRS design in a MISO communication system is considered in [24], [28]. A similar approach is also used in [22] viewing the IRS design as a discrete control problem. However, the discretization induces the curse of dimensionality [29], making the method difficult to scale to scenarios with very large IRSs. The works in [30], [31] extend the use of deep RL for IRS design in scenarios with multiple users. The aforementioned approaches at the intersection of deep RL and IRS phase shift design did not address the case of spatiotemporally correlated channel realizations. The existence of channel correlations requires careful attention when applying deep RL to optimize IRS parameters for wireless communications. The goal of the current work is to address this gap and explore the arising implications of such scenarios.

In this paper, we examine a scenario that is similar to the ones of [8], [11], [22], [24], [27], [28]. In particular, we examine the real-time design of an IRS that assists a MISO communication system with a mobile destination. The key differentiation in comparison to prior works is that we explicitly address the case of spatiotemporally correlated channels. Since the channels are spatiotemporally correlated, the IRS elements will also exhibit correlations in time and space. Therefore, by solely including the IRS parameters of the previous time step in the state representation, as proposed in previous works [8], [11], [21], [22], [24], [27], [28], the state becomes partially observed and the performance plummets. Here, we propose an actor-critic RL algorithm for continuous control to decide upon the IRS parameter values at every time step of system operation. The state of the actor-critic method is constructed to include a history of previous IRS parameter values. We show that if the history's length is at least equal to the temporal correlation of the channels, the algorithm provides about $2dB$ improvement in SNR at the destination as compared to the version of the algorithm where the state includes only the IRS parameter values of the previous time step.

Recent results in deep learning theory indicate that feedforward neural networks with Rectified Linear activations (ReLU MLPs) cannot represent the high frequency components of the target functions in regression tasks, a phenomenon also coined as *spectral bias* [32]. The critic of the proposed deep RL algorithm is an ReLU MLP that is trained to regress, via bootstrapping, the state-action pairs of the induced MDP to the corresponding values. Previous works have demonstrated that value functions that implicitly depend on spatiotemporally correlated communication channels possess high frequency components in the corresponding spectra [33], [34]. In this direction, we propose preprocessing of the state-action vector of the IRS phase shift design actor-critic algorithm with a Fourier kernel to ameliorate the spectral bias. This preprocessing provides an additional $2dB$ improvement in destination SNR.

Finally, it has been common practice in previous works that propose deep RL solutions for IRS optimization problems, to include the QoS metric of interest (in our case the destination SNR), as a component of the state representation. In his article, we illustrate that for the case of spatiotemporally correlated channels, the inclusion of the SNR in the state is a cause of divergence. We provide an explanation for this phenomenon that relates the SNR inclusion to the resulting critic Neural Tangent Kernel (NTK). In particular, we argue that the inclusion of the SNR increases the off-diagonal elements of the critic's NTK and violates sufficient conditions of convergence under neural value approximation. The aforementioned analysis prompts us to suggest a general framework for designing value-based deep RL methods for IRS phase shift design and beyond that predicates upon the structure of the critic's NTK.

The contributions of the current article can be summarized as follows:

- We propose a novel design for an IRS assisting a MISO communication system in the presence of destination mobility and spatiotemporally correlated channels. The design is achieved via a deep RL approach for continuous control, targeting to maximize the expected sum of SNRs over an infinite time horizon at the destination. The proposed RL approach addresses destination mobility by including the position of the receiver as a component of the state, and the spatiotemporally varying nature of the channel by augmenting the state with histories of the IRS parameter values and destination positions. It is illustrated that the augmented state enables almost $2dB$ increase in average destination SNR for a simulated scenario with 20 IRS elements and typical values pertaining to spatial and temporal correlations of urban communication channels.

- We propose preprocessing of the state-action vector with a Fourier kernel before passing it through the critic network in order to better capture high frequency variations of the optimal value function, arising due to the channel variability. This modification enables about $2dB$ additional improvement in average SNR at the receiver for simulated scenarios with 30 IRS elements and about $5dB$ improvement for simulated scenarios with 150 IRS elements as compared to implementations that directly employ ReLU MLPs for the critic. Moreover, the application of the Fourier preprocessing kernel enables stable training of our proposed approach without the need for a target network in critic updates, which constitutes a common heuristic of value-based deep RL methods.
- RL algorithms are designed assuming a state representation that captures the most relevant information regarding the agent's interaction with the environment. Previous works in deep RL for IRS phase shift design have included the metric of interest, such as the receiver's SNR, as part of the RL state vector. In contrast, our proposed work, based on empirical analysis, demonstrates that this design choice leads to instability and divergence specifically in the context of spatiotemporally correlated channels. We also provide an analysis that derives the mechanism that causes the divergence. The analysis is grounded on sufficient conditions on the critic's NTK [35] for convergence of value learning in deep RL. The connection between the NTK and the stability in value learning allows us to propose a principled framework for designing actor-critic methods for IRS phase shift optimization and problems that arise in wireless systems.

Initial results of the current work are presented in [36]. In particular, in the current work we extend the results of [36] by providing additional experimentation and by deriving the mechanism of the instability caused by the inclusion of the SNR as a state component. This analysis connects the design of deep RL algorithms for IRS phase shift design with the properties of the NTK of the critic.

Notation: We denote matrices and vectors by bold uppercase and bold lowercase letters, respectively. The operators $(\cdot)^T$ and $(\cdot)^H$ denote transposition and conjugate transposition respectively. Caligraphic letters will be used to denote sets. The ℓ_p-norm of $\boldsymbol{x} \in \mathbb{R}^n$ is $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x(i)|^p\right)^{1/p}$, for all $\mathbb{N} \ni p \geq 1$. The expectation of a random vector $\mathbf{x}$ is denoted as $\mathbb{E}(\mathbf{x})$.

II. Signal Model

We consider a communication system, like the one depicted in Fig. 1. It is a MISO system in which the source/base station is a Uniform Linear Array (ULA) with N antennas. The IRS is a 2D panel that consists of $M = M_x \times M_y$ passive (reflective) elements, with M_x denoting the per-row number of elements and M_y the per-column number of elements. The effect of the i-th IRS element on the impinging signal is the introduction of a phase shift, represented here by multiplication with $e^{j\theta_i}$, with $\theta_i \in [-\pi, \pi]$. It is assumed that each IRS element can be controlled independently of the other elements.

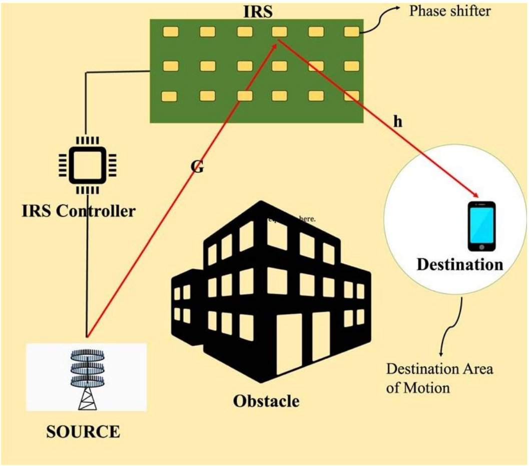


Fig. 1. IRS-aided MISO scenario that involves destination mobility.

The channels from the source to the IRS are denoted by $\mathbf{G} \in \mathbb{C}^{M\times N}$, and from the IRS to the destination by $\mathbf{h} \in \mathbb{C}^{M\times 1}$. All channels are assumed to be flat fading.

We consider a time-slotted scenario. In slot t, the source transmits a unit power symbol $\mu(t) \in \mathbb{C}$ after precoding it by vector $\mathbf{b} \in \mathbb{C}^{N\times 1}$. The transmit power budget is $\|\mathbf{b}\|_2^2 \leq P_{max}$. Let us assume that the direct link from the source to the destination is blocked, and so the source signal arrives at the destination through reflection by the IRS. The signal that is received at the destination can be expressed as:

$$y = \mathbf{h}^H \mathbf{\Phi} \mathbf{G} \mathbf{b} \mu + n \tag{1}$$

where $\mathbf{\Phi}$ is a diagonal matrix with entries $\mathbf{\Phi} = \mathrm{diag}(e^{j\theta_1}, e^{j\theta_2}, \ldots, e^{j\theta_M})$, $n \sim \mathcal{CN}(0, \sigma^2)$ is the reception noise at the destination.

For fixed channels, phase shift matrix and precoding weights, the instantaneous SNR at the destination is

$$SNR(t) = \frac{|\mathbf{h}^H \mathbf{\Phi} \mathbf{G} \mathbf{b}|^2}{\sigma^2} \tag{2}$$

and the optimal precoding vector is [8]

$$\mathbf{b}^* = \sqrt{P_{max}} \frac{(\mathbf{h}^H \mathbf{\Phi} \mathbf{G})^H}{\|\mathbf{h}^H \mathbf{\Phi} \mathbf{G}\|_2^2} \tag{3}$$

In this article we adopt the following assumptions: (i) The source-IRS and IRS-destination channels are random, exhibiting correlations with respect to time. The IRS-destination channel also exhibits correlations with respect to space that depend on the relative position of the destination to the IRS. (ii) The destination can move slowly within a confined small area of the 3D space. (iii) The receiver's position is assumed to be known at the IRS controller by the possible coexistence with a radar perception system [37].

Based on the above assumptions we will denote the channels as $\mathbf{h}(t, \mathbf{x}_t)$ and $\mathbf{G}(t)$, where $\mathbf{x}_t \in \mathbb{R}^3$ is the position of the destination in the 3D space at time step t. Under the assumption that the channels $\mathbf{h}(t, \mathbf{x}_t)$ and $\mathbf{G}(t)$ at time step t are known, the optimal base station precoding vector can be written as

$$\mathbf{b}^*(t) = \sqrt{P_{max}} \frac{(\mathbf{h}(t, \mathbf{x}_t)^H \boldsymbol{\Phi}(t)\mathbf{G}(t))^H}{\|\mathbf{h}(t, \mathbf{x}_t)^H \boldsymbol{\Phi}(t)\mathbf{G}(t)\|_2^2} \quad (4)$$

and the SNR at time step t as

$$SNR(t) = \frac{|\mathbf{h}(t, \mathbf{x}_t)^H \boldsymbol{\Phi}(t)\mathbf{G}(t)\mathbf{b}^*(t)|^2}{\sigma^2} \quad (5)$$

The design of matrix $\boldsymbol{\Phi}(t)$, aiming to further maximize the maximum (with respect to $\mathbf{b}$) receiver SNR can be formulated as

$$\max_{\boldsymbol{\Phi}(t)} \; SNR(t)$$
$$\text{s.t.} \quad |\boldsymbol{\Phi}(t)_{(i,i)}| = 1, \forall i = 1, 2, \ldots, M. \quad (6)$$

Solving problem (6) with Semidefinite Programming [12] requires complexity in the order of $O(M^6)$ [11].

In this work, we take a different approach that overcomes the computational overhead. Instead of solving the problem of (6) at every time step t, we design an actor-critic algorithm with deep value approximation. The actor is a parameterized function which learns the mapping from the state to the phase shift values at every time step. The state is comprised by the position of the destination at time step t, and histories of phase shift values and destination positions of previous time steps. The goal is that the learned actor/policy maximizes the expected sum of receiver SNRs over an infinite time horizon. The policy can be trained offline and subsequently deployed. In that case, computing the optimal phase shift coefficients at each time step involves a forward pass through the actor/policy network. This deployment option requires utilizing data collected from the deployment environment. This approach assumes that the channel statistics remain unchanged during system operation. However, in order to accommodate real-time changes in the channel statistics, updates must be performed during the system operation.

III. Off-Policy Deep Actor-Critic for Continuous Control

RL is concerned with scenarios in which an agent interacts with an environment producing a sequence of states, actions and rewards. This gives rise to a MDP which is defined as a tuple $\langle S, A, R, P, p, \gamma \rangle$; S denotes the state space, A the action space, $R : S \times A \to \mathbb{R}$ is the reward function, $P : S \times A \to S$ is the transition function that implicitly defines the dynamics of the environment, $p(\mathbf{s})$ is the distribution of the initial state, and $\gamma \in (0, 1)$ is the discount factor that quantifies the "interest" of the agent in long-term delayed rewards. The goal of RL is to learn a mapping from states to actions, namely a policy, $\pi(\mathbf{a}|\mathbf{s})$, that maximizes the expected discounted sum of rewards $J = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t)\right]$.

The state-action value function Q^π is defined as the expected discounted sum of rewards starting from a state-action pair and following the policy π thereafter.

$$Q^\pi(\mathbf{s}, \mathbf{a}) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t) | (\mathbf{s}_0, \mathbf{a}_0) = (\mathbf{s}, \mathbf{a})\right] \quad (7)$$

The optimal value function $Q^*(s, a)$ is the fixed point of the Bellman backup operator [38], i.e.,:

$$Q^*(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{(\mathbf{s}' \sim P)}\left[R(\mathbf{s}, \mathbf{a}) + \gamma \max_{\mathbf{a}'} Q^*(\mathbf{s}', \mathbf{a}')\right] \quad (8)$$

In off-policy deep actor-critic methods, we assume the existence of an Experience Replay, denoted as $D = \{\mathbf{s}_i, \mathbf{a}_i, \mathbf{s}'_i, r_i\}_{i=1}^{N_{exp}}$ which comprises N_{exp} transitions resulting from the agent's interaction with the environment. Each transition includes the state $\mathbf{s}_i$, action $\mathbf{a}_i$, subsequent state $\mathbf{s}'_i$, and reward r_i. Experience Replay is a technique utilized in off-policy deep RL, as introduced by [39]. It involves storing past experiences in a buffer and randomly sampling from it during training. This process enables the agent to learn from infrequent or distant events, enhancing sample efficiency, preventing feedback loops. The optimal state-action value function is parameterized as a neural network with parameters $\mathbf{w}$, denoted as $Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})$. The aforementioned network is coined as "critic" or "value network" and we will use both terms interchangeably for the rest of the article. The process of approximating the value function entails the sampling of a batch of transitions from the Experience Replay and the following gradient descent update rule:

$$\mathbf{w} \to \mathbf{w} + \eta \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}',r) \sim D}\left[\left(Q^*_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) - Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})\right)\nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})\right] \quad (9)$$

The scalar parameter η denotes the learning rate. The process of learning the value function is usually unstable due to the fact that the bootstrapping target $Q^*_{\mathbf{w}}(\mathbf{s}, \mathbf{a})$ depends on the estimator $Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})$. One popular heuristic includes the use of a target network with parameters $\mathbf{w}_{target}$ for computing the bootstrapping target. The parameter vector $\mathbf{w}_{target}$ slowly tracks the parameter vector $\mathbf{w}$ with the following update rule [40]:

$$\mathbf{w}_{target} \to \tau\mathbf{w} + (1 - \tau)\mathbf{w}_{target}, \quad (10)$$

where $\tau \ll 1$. The policy is explicitly parametrized by a neural network with parameters $\boldsymbol{\varphi}$, denoted as $\pi_{\boldsymbol{\varphi}}(\mathbf{s})$. The aforementioned neural network is typically called the "actor" or the "policy network". We will use both terms interchangeably for the rest of the article. The actor is a neural network that either maps the state to a distribution over the action space (stochastic policies) or maps the state to the corresponding action (deterministic policies). We will focus on deterministic actors for the problem at hand. The policy is updated by the following update rule that stems from the deterministic policy gradient theorem [41]:

$$\boldsymbol{\varphi} \to \boldsymbol{\varphi} + \eta \mathbb{E}_{\mathbf{s} \sim D}\left[\nabla_{\mathbf{a}} Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi_{\boldsymbol{\varphi}}(\mathbf{s})} \nabla_{\boldsymbol{\varphi}} \pi_{\boldsymbol{\varphi}}(\mathbf{s})\right] \quad (11)$$

IV. Actor-Critic Approach for IRS Phase Shift Optimization

The first step in the process of designing deep actor-critic methods for the IRS phase shift design problem is to explicitly define the elements of the underlying MDP.

State: As stated before, the IRS is aware of the position of the destination during the time step of interest t. Therefore, the first component of the state vector is the current position of the destination in the 3D space, denoted as $\mathbf{x}_t$. Previously proposed methods [21], [24], [28] typically include the phase shift coefficients of the IRS at the previous time step only, i.e., assume the state to be Markovian. However, this assumption ignores significant part of the temporal evolution of the channels. In the case of temporally correlated channels, the MDP becomes Partially Observable (POMDP [42]). To avoid partial observability, we include the phase shift values of the IRS elements for W previous time steps along with the corresponding destination positions. This is similar to the state representation design in the seminal work of [39], where the deep Q learning algorithm is introduced and tested on the environments of the Atari Domain. In [39], the state is constructed by concatenating multiple consecutive frames of the video to avoid partial observability. The presence of temporal correlations in our problem setting poses challenges for adopting even unsupervised approaches like the one proposed in [43]. These approaches typically involve offline data generation and subsequent training of one or an ensemble of neural networks to optimize an unsupervised objective. However, this methodology implicitly assumes that the training data are i.i.d. samples, which is not applicable when dealing with spatiotemporally correlated channels. Therefore the state of the corresponding MDP can be expressed as:

$$\mathbf{s}_t = \mathbf{x}_t, \mathbf{x}_{t-1} \ldots \mathbf{x}_{t-W}, \boldsymbol{\Theta}(t-1) \ldots \boldsymbol{\Theta}(t-W) \tag{12}$$

where $\boldsymbol{\Theta}(j) = [\theta_1^j, \theta_2^j, \ldots, \theta_M^j]$ and θ_i^j denotes the phase shift coefficient of the i-th element of the IRS during the j-th time step of system operation.

Action: The action representation is defined as the component-wise difference between the phase shift coefficients of the IRS during the current step and the previous step.

$$\mathbf{a}_t = \delta\theta_1^t, \delta\theta_2^t, \ldots, \delta\theta_M^t, \tag{13}$$

where $\delta\theta_i^t = \theta_i^t - \theta_i^{t-1}$.

Reward: Since we aspire to maximize the expected sum of SNRs at the destination, it is natural to choose the reward at t to be the achieved receiver SNR during the aforementioned time step.

$$r_t = SNR(t) \tag{14}$$

A. Deep Deterministic Policy Gradient

After explicitly defining the MDP, we adapt the algorithmic structure of the Deep Deterministic Policy Gradient (DDPG) algorithm [40] in order to learn the control policy that maps the state of the MDP at time step t to the action that maximizes the sum of SNRs in expectation. The DDPG is an actor-critic RL algorithm that employs deep neural networks for function approximation. The general framework for the DDPG algorithm for IRS phase shift design is depicted in Fig. 3. In terms of the critic, we employ 2 value networks $Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})$ and $Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})$. During every update step, we sample a batch of experiences/transitions from the Experience Replay and we update each critic network independently with the update rule of Eq. (9). The actor is parametrized as a neural network $\pi_{\boldsymbol{\varphi}}(\mathbf{s})$. At every update step, we use the sampled batch of experiences to update the actor with the update rule of Eq. (11).

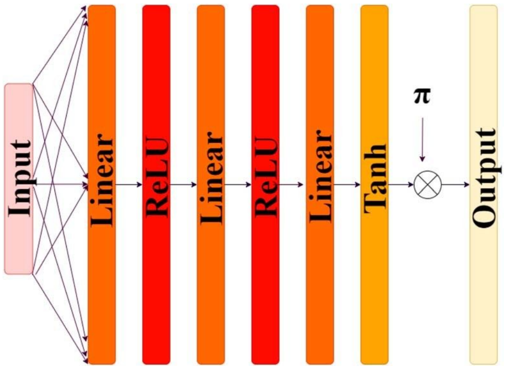


Fig. 2. The architecture of the actor network which ensures, by design, the satisfiability of the unit modulus constraints.

We employ 3 target networks, one for each of the corresponding main ones. Each of the critic target networks is employed to compute the bootstrapping target $Q^*_{\mathbf{w}_i}(\mathbf{s}, \mathbf{a})$ for the corresponding critic update (Eq. (9)). Furthermore, the computing of the bootstrapping target involves the maximization of the critic estimation with respect to the action

$$\mathrm{E}_{\mathbf{s}\sim P} \; R(\mathbf{s}, \mathbf{a}) + \gamma \max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}) \; .$$

Since the actor provides the action that maximizes the value function at each state, we compute the maximum as follows:

$$\max_{\mathbf{a}} Q_{\mathbf{w}_i}(\mathbf{s}, \mathbf{a}) = Q_{\mathbf{w}_i}(\mathbf{s}, \pi_{\boldsymbol{\varphi}}(\mathbf{s})), i = 1, 2 \tag{15}$$

where $\pi_{\boldsymbol{\varphi}}$ is the target network of the actor. At the end of every neural network update step, the target networks' parameters are updated using Eq. (10). The update of the actor involves the differentiation of the critic with respect to the action (Eq. (11)). The critic that is being used for the actor update is the one that corresponds to the minimum value estimation for every state-action pair of the corresponding batch. This is the practice introduced in [44] to mitigate overestimation in the process of neural value approximation.

We employ ReLU MLPs for all the parametrized functions. A critical question in the design of the RL algorithm is how to ensure that the unit modulus constraints of problem (6) are satisfied throughout training. We address the aforementioned issue by properly designing the policy network. The policy estimates the component-wise difference between the phase shift coefficients of the current time step and the phase shift coefficients of the previous time step. Each component of the resulting phase shift vector at every time step should be restricted in the range $[-\pi, \pi]$. Therefore, even though the in-between-layers activations of the actor are ReLUs, we choose the output of the last layer to pass through a hyperbolic tangent (Tanh) activation that squashes each component to the range $[-1, 1]$. Subsequently, each component is multiplied with π. That being the case, each component of the action is ensured to be in the range $[-\pi, \pi]$. Since the action denotes the difference between the phase shift coefficient values for two consecutive time steps, we clip each resulting phase shifter to the range $[-\pi, \pi]$. The

Algorithm 1 RL-IRS-Base

```
Initialize Experience Replay $D$ with experiences from a random policy, the critic networks $Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})$ $Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})$ the actor network $\pi_{\boldsymbol{\varphi}}(\mathbf{s})$, the corresponding target networks $Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})$ $Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})$ $\pi_{\boldsymbol{\varphi}}(\mathbf{s})$, the learning rate $\eta$, the coefficient $\tau$ (Eq. (10) for target updates), the batch size $N_B$, the discount factor $\gamma$.
$\mathbf{w}_1 \to \mathbf{w}_1$
$\mathbf{w}_2 \to \mathbf{w}_2$
$\boldsymbol{\varphi} \to \boldsymbol{\varphi}$
Main Body
for all episodes do
    for all time steps $t$ do
        estimate the position of the destination $\mathbf{x}_t$
        compute $\mathbf{s}_t = [\mathbf{x}_t \ldots \mathbf{x}_{t-W}, \boldsymbol{\Theta}(t-1) \ldots \boldsymbol{\Theta}(t-W)]$
        compute action $\mathbf{a}_t = \pi_{\boldsymbol{\varphi}}(\mathbf{s}_t) = \delta\theta_1^t, \delta\theta_2^t \ldots, \delta\theta_M^t$
        compute the phase shifters $\theta_i^t = \theta_i^{t-1} + \delta\theta_i^t$
        $\theta_i^t = \mathrm{clip}(-\pi, \pi)$
        compute $\boldsymbol{\Phi} = \mathrm{diag}(e^{j\theta_1^t}, \ldots, e^{j\theta_M^t})$
        compute $r_t = SNR(t)$
        compute destination position at next step $\mathbf{x}_{t+1}$
        compute $\mathbf{s}_{t+1}$
        Store $\{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}$ in $D$
        Updates
        Sample batch $\{\mathbf{s}, \mathbf{a}, \mathbf{s}, r\}$ of size $N_B$ from $D$
        Target1 $= r + \gamma Q_{\mathbf{w}_1}(\mathbf{s}, \pi_{\boldsymbol{\varphi}}(\mathbf{s})) - Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})$
        $\mathbf{w}_1 \to \mathbf{w}_1 + \eta$ Target1 $\nabla_{\mathbf{w}_1} Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})$
        Target2 $= r + \gamma Q_{\mathbf{w}_2}(\mathbf{s}, \pi_{\boldsymbol{\varphi}}(\mathbf{s})) - Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})$
        $\mathbf{w}_2 \to \mathbf{w}_2 + \eta$ Target2 $\nabla_{\mathbf{w}_2} Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})$
        $Q_{\mathbf{w}} = \min_{i=1,2} Q_{\mathbf{w}_i}(\mathbf{s}, \mathbf{a})$
        $\boldsymbol{\varphi} \to \boldsymbol{\varphi} + \eta \nabla_{\mathbf{a}} Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi_{\boldsymbol{\varphi}}(\mathbf{s})} \nabla_{\boldsymbol{\varphi}} \pi_{\boldsymbol{\varphi}}(\mathbf{s})$
        $\mathbf{w}_1 \to \tau\mathbf{w}_1 + (1-\tau)\mathbf{w}_1$
        $\mathbf{w}_2 \to \tau\mathbf{w}_2 + (1-\tau)\mathbf{w}_2$
        $\boldsymbol{\varphi} \to \tau\boldsymbol{\varphi} + (1-\tau)\boldsymbol{\varphi}$
    end for
end for
```

architecture of the actor network is illustrated in Fig. 2. We denote this base algorithm as **RL-IRS-Base** and its algorithmic structure is illustrated in Algorithm 1.

B. Fourier Features

As demonstrated in [33], [34], [45], MDP value functions that implicitly depend on spatiotemporally correlated communication channels typically exhibit high local variability that corresponds to high frequency spectrum components. The critics of the proposed **RL-IRS-Base** are ReLU MLPs that are trained to learn the underlying value function of the formulated MDP under a bootstrap regression framework. Due to spectral bias the critics might not be able to accurately capture the high frequency components of the true underlying value function. This can result in erroneous value approximation and in suboptimal learned policies since the actor is updated based on critic estimates.

Inspired by recent results in graphics, scene rendering and low-dimensional regression with neural networks [46], [47], [48], we propose the preprocessing of the state-action vector with a random Fourier feature kernel before passing it through the critic. We denote as $\tilde{\mathbf{s}} \in \mathbb{R}^{(W+1)(M+3)}$ the vector that corresponds to the concatenation of the state vector $\mathbf{s}$ and the action vector $\mathbf{a}$:

$$\tilde{\mathbf{s}} = [\mathbf{s}, \mathbf{a}]^T \in \mathbb{R}^{(W+1)(M+3)} \to \mathbf{v} = [\cos(2\pi\mathbf{B}\tilde{\mathbf{s}}), \sin(2\pi\mathbf{B}\tilde{\mathbf{s}})]^T \quad (16)$$

where the matrix $\mathbf{B} \in \mathbb{R}^{N_f \times (W+1)(M+3)}$ is the transformation of the Fourier kernel. The value N_f corresponds to the number of resulting Fourier features and, in this case, is equal to the number of neurons of the first layer of the critic architecture. Each element of $\mathbf{B}$ is drawn from $N(0, \sigma_{\mathbf{B}}^2)$. The operations $\cos(\cdot)$ and $\sin(\cdot)$ in Eq. (16) are applied element-wise. The variation that employs the Fourier preprocessing on the critic is denoted as **RL-IRS-FF** and the corresponding algorithmic structure is illustrated in Algorithm 2. The key distinctions between **RL-IRS-FF** and **RL-IRS-Base** reside in the computations of $\mathbf{v}$ and $\mathbf{v}$. These vectors correspond to the outputs of the Fourier preprocessing kernel and serve as the input vectors for the critic networks.

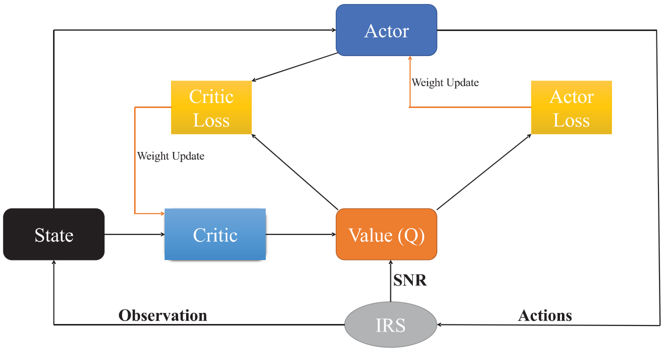


Fig. 3. The general framework of DDPG [40] for IRS phase shift design.

C. SNR as a Component of the State

A critical part of the process of designing RL algorithms for control problems is how to construct the representation of the state. The incentive behind the state construction is to include as much information that relates to the agent's interaction with the environment as possible. This is especially prevalent in research works that are at the intersection of RL and robotics [49]. Under the aforementioned intuition, previous works that proposed deep RL algorithms for IRS phase shift optimization [24], [27], [28], [30], [31] include the metric of interest (in our case the SNR at the destination) as a component of the state vector. On a first thought this seems like a good design decision since it provides additional information of the agent's behavior regarding the task of interest. On the other hand, deep actor-critic methods employ neural networks for parametrizing the value function, so the state construction should account for the aforementioned element as well. We wish to investigate the effect of the inclusion of the SNR as a state component, both empirically and theoretically, therefore we provide a variation of the **RL-IRS-Base** that includes the receiver's SNR as a

Algorithm 2 RL-IRS-FF

```
Initialize Experience Replay $D$ with experiences from a random policy, the critic networks $Q_{\mathbf{w}_1}(\mathbf{v})$, $Q_{\mathbf{w}_2}(\mathbf{v})$, the actor network $\pi_{\boldsymbol{\varphi}}(\mathbf{s})$, the corresponding target networks $Q_{\mathbf{w}_1}(\mathbf{s}, \mathbf{a})Q_{\mathbf{w}_2}(\mathbf{s}, \mathbf{a})\pi_{\boldsymbol{\varphi}}(\mathbf{s})$, the learning rate $\eta$, the coefficient $\tau$, the batch size $N_B$, the discount factor $\gamma$, the variance $\sigma_{\mathbf{B}}$ of the Normal distribution to sample the elements of the Fourier kernel $\mathbf{B}$ from.
Initialize $\mathbf{B}$ where $\mathbf{B}_{ij} \sim N(0, \sigma^2_{\mathbf{B}})$
$\mathbf{w}_1 \to \mathbf{w}_1$
$\mathbf{w}_2 \to \mathbf{w}_2$
$\boldsymbol{\varphi} \to \boldsymbol{\varphi}$
Main Body
for all episodes do
    for all time steps $t$ do
        estimate the position of the destination $\mathbf{x}_t$
        compute $\mathbf{s}_t = [\mathbf{x}_t, \ldots, \mathbf{x}_{t-W}, \boldsymbol{\Theta}(t-1) \ldots \boldsymbol{\Theta}(t-W)]$
        compute action $\mathbf{a}_t = \pi_{\boldsymbol{\varphi}}(\mathbf{s}_t) = \delta\theta_1^t, \delta\theta_2^t, \ldots, \delta\theta_M^t$
        compute the phase shifters $\theta_i^t = \theta_i^{t-1} + \delta\theta_i^t$
        $\theta_i^t = $ clip$(-\pi, \pi)$
        compute $\Phi = $ diag$(e^{j\theta_1^t}, \ldots, e^{j\theta_M^t})$
        compute $r_t = SNR(t)$
        compute destination position at next step $\mathbf{x}_{t+1}$
        compute next state $\mathbf{s}_{t+1}$
        Store $\{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_t\}$ in $D$
        Updates
        sample batch $\{\mathbf{s}, \mathbf{a}, \mathbf{s}, r\}$ of size $N_B$ from $D$
        compute $\tilde{\mathbf{s}} = [\mathbf{s}, \pi_{\boldsymbol{\varphi}}(\mathbf{s})]^T$
        compute $\tilde{\mathbf{s}} = [\mathbf{s}, \mathbf{a}]^T$
        compute $\mathbf{v} = [cos(2\pi\mathbf{B}\tilde{\mathbf{s}}), sin(2\pi\mathbf{B}\tilde{\mathbf{s}})]$
        compute $\mathbf{v} = [cos(2\pi\mathbf{B}\tilde{\mathbf{s}}), sin(2\pi\mathbf{B}\tilde{\mathbf{s}})]$
        Target1 $= r + \gamma Q_{\mathbf{w}_1}(\mathbf{v}) - Q_{\mathbf{w}_1}(\mathbf{v})$
        $\mathbf{w}_1 \to \mathbf{w}_1 + \eta$Target1$\nabla_{\mathbf{w}_1} Q_{\mathbf{w}_1}(\mathbf{v})$
        Target2 $= r + \gamma Q_{\mathbf{w}_2}(\mathbf{v}) - Q_{w_2}(\mathbf{v})$
        $\mathbf{w}_2 \to \mathbf{w}_2 + \eta$Target2$\nabla_{\mathbf{w}_2} Q_{\mathbf{w}_2}(\mathbf{v})$
        $Q_w = \min_{i=1,2} Q_{w_i}(v)$
        $\boldsymbol{\varphi} = \boldsymbol{\varphi} + \eta \nabla_{\mathbf{a}} Q_{\mathbf{w}}(\mathbf{v}(\mathbf{s}, \mathbf{a}))|_{\mathbf{a}=\pi_{\boldsymbol{\varphi}}(\mathbf{s})} \nabla_{\boldsymbol{\varphi}} \pi_{\boldsymbol{\varphi}}(\mathbf{s})$
        $\mathbf{w}_1 \to \tau\mathbf{w}_1 + (1-\tau)\mathbf{w}_1$
        $\mathbf{w}_2 \to \tau\mathbf{w}_2 + (1-\tau)\mathbf{w}_2$
        $\boldsymbol{\varphi} \to \tau\boldsymbol{\varphi} + (1-\tau)\boldsymbol{\varphi}$
    end for
end for
```

component of the state and we denote this as **RL-IRS-SNR-state**. The structure of **RL-IRS-SNR-state** follows the same framework as **RL-IRS-Base** (Algorithm 1). The distinction lies in the construction of the state representation. In **RL-IRS-Base**, the state comprises the IRS parameter values and the destination locations for a specified number of previous time steps. On the other hand, the **RL-IRS-SNR-state** approach introduces additional components, namely the receiver's SNRs corresponding to all time steps of the window. During system operation, the destination positions, IRS parameter values, and destination SNRs (in the case of **RL-IRS-SNR-state**) for all previous time steps within the current window are stored in memory. The destination position for the current time step is estimated, and all the aforementioned parameters collectively form the current state. This state is then input to the policy network, which produces the corresponding action. Based on the action, the IRS's current parameter values can be computed. Subsequently, the channels are estimated, and the destination SNR for the current step is calculated using Eq. (5).

V. Experiments

Our goal is to simulate a set up as the one depicted in Fig. 1. In addition, we want to integrate in this the notion of channels that exhibit correlations with respect to both time and space.

A. Channel Model

The statistical description of the channel between a source element and position $\mathbf{p} \in \mathbb{R}^3$ during time slot t, can be modeled as a product of four terms [50], i.e.

$$g(\mathbf{p}, t) \quad g^{PL}(\mathbf{p})\, g^{SH}(\mathbf{p}, t)\, g^{MF}(\mathbf{p}, t)\, e^{j2\pi\varphi(t)}, \tag{17}$$

where $g^{PL}(\mathbf{p}) \quad \|\mathbf{p} - \mathbf{p}_S\|_2^{-/2}$ is the path-loss component, with being the path-loss exponent and $\mathbf{p}_S$ the position of the source; $g^{SH}(\mathbf{p}, t)$ the shadow fading component; $g^{MF}(\mathbf{p}, t)$ the multi-path fading component. Finally, $e^{j2\pi\varphi(t)}$ denotes the phase. The parameter φ is uniformly distributed in $[0, 1]$ A similar representation accounts for the channel from the position to the destination.

If we apply the logarithm of the squared channel magnitude of (17), we obtain:

$$\begin{aligned} G(\mathbf{p}, t) \quad & 10\log_{10}(|g(\mathbf{p}, t)|^2) \\ & \alpha^g(\mathbf{p}) + \beta^g(\mathbf{p}, t) + \xi^g(\mathbf{p}, t), \end{aligned} \tag{18}$$

where

$$\alpha^g(\mathbf{p}) \quad -10 \log_{10} \|\mathbf{p} - \mathbf{p}_S\|_2, \tag{19}$$

$$\beta^g(\mathbf{p}, t) \quad 10\log_{10} |g(\mathbf{p}, t)|^2 \sim N(0, \zeta^2), \quad \text{and} \tag{20}$$

$$\xi^g(\mathbf{p}, t) \quad 10\log_{10} |g(\mathbf{p}, t)|^2 \sim N(\rho, \sigma_\xi^2). \tag{21}$$

In the above, ζ^2 is the shadowing power, and ρ, σ_ξ^2 are respectively the mean and variance of the multipath fading component.

Although the multipath fading component, $\xi^g(\mathbf{p}, t)$, is i.i.d. between different positions and times, the shadowing component, $\beta^g(\mathbf{p}, t)$, is correlated. Specifically, the shadowing component between any two positions $\mathbf{p}_i$ and $\mathbf{p}_j$, at two time slots t_a and t_b, exhibits correlations according to [51]

$$\mathrm{E}[\beta^g(\mathbf{p}_i, t_a)\beta^g(\mathbf{p}_j, t_b)] = \tilde{\boldsymbol{\Sigma}}^g(\mathbf{p}_i, \mathbf{p}_j)e^{-\frac{|t_a - t_b|}{c_2}}, \tag{22}$$

where

$$\tilde{\boldsymbol{\Sigma}}^g(\mathbf{p}_i, \mathbf{p}_j) \quad \zeta^2 e^{-\|\mathbf{p}_i - \mathbf{p}_j\|_2 / c_1}. \tag{23}$$

with c_1 denoting the correlation distance, and c_2 the correlation time.

B. Channel Data

We assume that the set up takes place in a 3D cube $(20 \times 20 \times 20)$ The operation space is discretized in cube cells of volume $(1 \times 1 \times 1)$ The 3D positions of the MIMO source and the IRS are fixed beforehand and remain the same throughout every experiment. The destination can freely move in an area comprised by 4 distinct cube cells. All cube cells that constitute the destination area of motion are on the same vertical level (simulating a person walking in a small space and being serviced by the nearby receiver or an autonomous vehicle that navigates itself in a parking lot and is required to exchange information with the nearby base station). The destination occupies one cell per time slot and can also, potentially, move to a neighboring cell at the subsequent slot. We simulate the channel data in the same way that were simulated for the 3D scenario of [34] and so that they have statistics as described in [52]. In particular, the log-magnitude of the channel has 3 additive components, the pathloss with exponent $l = 2.3$, the multipath, which is i.i.d zero-mean Gaussian with variance $\sigma_\xi = 0.6$, and the shadowing which is a zero-mean Gaussian correlated in time and space. The correlation distance is $c_1 = 1.2$, the correlation time $c_2 = 5$ and the shadowing power is $\zeta^2 = 6$. The MIMO source transmission power is $P_{max} = 65dbm$. The variance of the reception noise at the destination is $\sigma^2 = 0.5$. These parameters are consistent with real time measurements [18] for urban channel communication environments.

C. Actor-Critic Specifications

Each neural network that is being employed (2 critics and 1 actor) is feedforward with 3 layers. The activations between layers are ReLU, with the exception of the last layer of the actor where we employ Tanh. Each layer consists of 400 neurons. We employ the Adam optimizer [53] for the updates, with a learning rate of 2e-4 and batch size of 64. The parameter τ for the updates of the target networks is chosen to be 0.005. When it comes to the **RL-IRS-FF**, the Fourier kernel $\mathbf{B}$ is populated with elements drawn from a zero-mean Normal distribution with variance 0.01. The size of the Experience Replay is set to 1e+6 and the discount factor γ is set to 0.99. The MIMO source is comprised by 5 antenna elements.

D. Discussion

The training performances of the 3 proposed deep RL algorithms, namely **RL-IRS-Base**, **RL-IRS-FF** and **RL-IRS-SNR-state** are demonstrated in Figs. 4 and 5. In particular, Fig. 4 corresponds to the training performances of the algorithms for an IRS with 20 phase shifters and Fig. 5 corresponds to the training performances for an IRS with 30 phase shifters. The trajectory of the destination is exactly the same for all experiments to validate the direct comparison. The algorithm that introduces the Fourier feature preprocessing for the critic provides significant improvements in terms of convergence speed, stability and SNR accumulation in comparison to the other 2 approaches. In particular, for an IRS with 20 reflective elements, the **RL-IRS-FF** provides an increase of approximately $2dB$ in terms of the achieved destination SNR. When it comes to the case of an IRS with 30 elements, the SNR increase is almost $3dB$. We attribute this effect on the mitigation of the spectral bias in the process of the neural value approximation. The variation that includes the SNR at the representation of the state (**RL-IRS-SNR-state**) performs poorly and is prone to divergence. We investigate how the inclusion of the SNR as a state component affects function approximation in the process of neural value learning in Section VI.

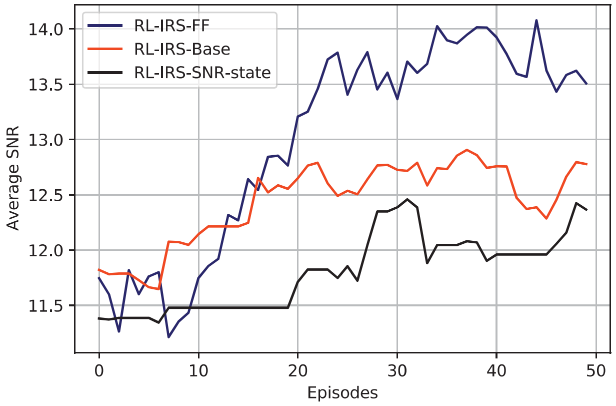


Fig. 4. Curves for the training performances of the 3 discussed algorithms, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state** for IRS with $\mathbf{20}$ phase shift elements. Each plot corresponds to the training for 50 episodes and each episode is comprised by 300 steps. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

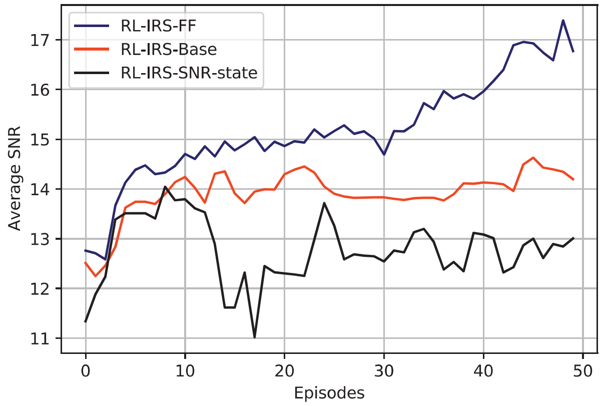


Fig. 5. Curves for the training performance of the 3 discussed algorithms, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state** for IRS with $\mathbf{30}$ phase shift elements. Each plot corresponds to the training for 50 episodes and each episode is comprised by 300 steps. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

Fig. 6 illustrates the average SNR, per time step, achieved by **RL-IRS-FF** and **RL-IRS-Base**, after convergence, for different numbers of IRS phase shift elements. The performance of **RL-IRS-SNR-state** is omitted because the approach frequently diverges. As can be extrapolated, the performance of **RL-IRS-FF** is consistently better than the performance of **RL-IRS-Base** for all sizes of the IRS. Both methods seem to plateau at about 25 elements. This is an indication that we have reached the representational capacity of the critic.

In order to achieve consistent increase in the destination SNR by applying the 2 algorithms to scenarios with IRSs with larger numbers of reflective elements than 30, we would need to increase the number of parameters for the actor and the critic architectures.

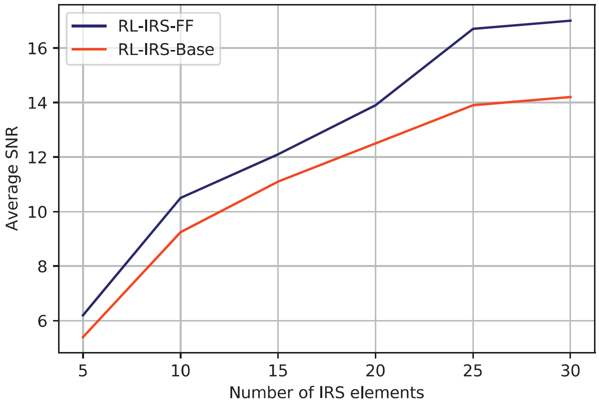


Fig. 6. The average SNR at the destination achieved by **RL-IRS-FF** and **RL-IRS-Base**, after convergence, with respect to the number of IRS elements.

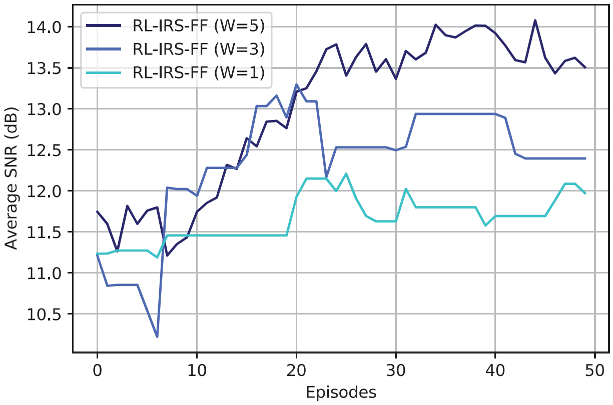


Fig. 7. The training performance of **RL-IRS-FF** for 3 different values of the window size W. Each episode is comprised by 300 time steps and each curve is an average over 10 seeds.

E. How to Choose the Size of the Window W

A critical hyperparameter, when it comes to the performance of all proposed approaches, is the length of the window of previous time steps that pertain to the state construction. It is prevalent that we would like to make the window length arbitrarily large to ensure that the state is fully observable. An increase in the window size requires an increase in the memory requirements for training since the proposed approaches are off-policy RL methods and they involve storing transitions in an Experience Replay Memory. Fig. 7 demonstrates the performance of **RL-IRS-FF** for 3 different values of W, namely 1, 3 and 5. The performance improves with an increase in window size. The length of the window needs to be at least equal to the temporal coefficient of the shadowing correlation of the channels (in our case 5). Any increase in the window length beyond the value of the temporal correlation coefficient does not seem to correspond to significant improvement in performance.

F. Remarks on Stability

Performance stability in actor-critic algorithms can be assessed from two perspectives: reward stability and value learning stability. Reward stability is determined by the variance of accumulated (or average) per-episode rewards. On the other hand, value learning stability focuses on how well the algorithm approximates the optimal value of the MDP. In highly stochastic environments, the optimal reward can vary across episodes. Therefore, a policy that corresponds to the optimal or near-optimal value function may exhibit higher reward variability compared to a policy that corresponds to a suboptimal value function estimation.

To illustrate this concept, consider a hypothetical experiment with a 3-state MDP and three discrete actions. The agent always starts at state s_0 and has the option to stay at s_0 (reward of 1), move left to s_1 (reward of -1000), or move right, accumulating a stochastic reward that is consistently higher than 1 but changes every episode. Choosing to move left is always unfavorable as it results in a reward of -1000 in every episode. Now, let's examine the remaining two policies. The policy of staying at the same state will yield a stable constant reward of 1 in every episode, resulting in a reward performance variability of 0. On the other hand, the optimal policy of always moving right will exhibit higher variability in reward performance due to the stochastic nature of the associated reward, yet the estimation of the optimal value function corresponding to this policy is accurate. This is because the agent has learned the optimal value function, which indicates that the action with the highest value is to move right.

The difference between these policies in the hypothetical environment (staying in the same position vs. moving right) mirrors the distinction between **RL-IRS-Base** and **RL-IRS-FF**. While **RL-IRS-FF** displays a more noisy reward performance compared to **RL-IRS-Base** (Figs. 4 and 5), its overall performance is significantly better. This is because **RL-IRS-FF** is capable of estimating the optimal value function more accurately, allowing it to learn a policy that is closer to optimal. Consequently, although the stochastic dynamics introduce variability in the optimal reward per episode, the superior policy achieved by **RL-IRS-FF** can approach higher rewards in practice.

G. Why Not Use Sinusoidal Representation Networks for the Critic Parametrization?

The Fourier features preprocessing is not the only way to mitigate the spectral bias. The authors in [48] propose a novel neural network architecture to overcome the spectral bias in low-dimensional regression tasks. They name this new architectural scheme as Sinusoidal Representation Networks (SIRENs). The architecture is feedforward with sinusoids as activation functions between layers.

In particular, assuming an intermediate layer of the neural network with input $\mathbf{x} \in \mathbb{R}^n$, then the output is an affine transformation $sin(\mathbf{w}^{\mathsf{T}}\mathbf{x} + b)$. Since the layer is not the network's first, the input $\mathbf{x}$ is arcsine distributed. Under these assumptions, it was shown in [48] that, if the elements of $\mathbf{w}$, namely w_i, are initialized from a uniform distribution $w_i \sim U(-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{n}})$, then $\mathbf{w}^{\mathsf{T}}\mathbf{x} \sim N(0, 1)$ as n grows. Therefore one should initialize the weights of all intermediate layers with $w_i \sim U(-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{n}})$. The neurons of the first layer are initialized with the use of a scalar hyperparameter ω_0, so that the output of the first layer, $sin(\omega_0 \boldsymbol{W}\boldsymbol{x} + b)$ spans multiple periods over $[-1, 1]$. $\boldsymbol{W}$ is a matrix whose elements correspond to the weights of the first layer.

It would, at first, seem reasonable to parametrize the critics of our proposed approaches as SIRENs. The SIREN architecture was specifically designed to process input vectors whose components take values roughly in the same range. In the case of our proposed deep RL approaches for IRS phase shift design, the state-action vector is comprised by components that take values in very different ranges (the destination position is a coordinate vector in $\mathbb{R}^3$ that takes values in the range $[0, 20]^3$ and the each phase shifter takes values in the range $[-\pi, \pi]$). Therefore it is impractical to tune the SIREN for the task at hand, especially the parameter w_0.

H. Robustness With Respect to Radar Noise

The position of the destination at time step t is estimated with the use of a coexisting radar perception system [37]. The receiver position is subject to the radar's range and angle resolution, therefore cannot be assumed to be precisely known.

In order to test the robustness of the proposed methods with respect to the noise induced by the finite range and angle resolution of the radar perception system, we conduct another set of experiments depicted in Fig. 8. The typical range resolution for contemporary mmWave radar systems for automotive and urban applications is about $0.2m$[54]. We simulate the noisy radar estimates by conducting the same set of experiments as in Fig. 5, but we add a vector that we sample from a uniform distribution $U(\mathbf{0}, [0.2, 0.2, 0.2]^T)$ to the destination position at every time step ($\mathbf{x}_t$).

As can be extracted by Fig. 8, both methods **RL-IRS-FF** and **Rl-IRS-Base** are generally robust with respect to the noise induced by the radar resolution. Still, the superior performance of the **RL-IRS-FF** remains in this setting.

I. Simulations With Large IRS

Subsequently we present a larger example, where the number of IRS elements is 150 and the destination motion is confined in a larger area of space (25 grid cells). For this particular example we make use of a larger critic and a larger policy network. In particular, the number of neurons at each layer is 2048 and the number of layers remains the same as for the previous experiments. The variance of the Fourier kernel is chosen to be 0.001The corresponding SNR results throughout training for all examined methods are illustrated in Fig. 9. The **RL-IRS-FF** algorithm demonstrates a significant improvement in SNR compared to **RL-IRS-Base**, achieving an approximate SNR gain of $5dm$ by the end of training. However, in this particular setting, the divergence of **RL-IRS-SNR-state** is less pronounced. Despite this, **RL-IRS-SNR-state** still exhibits inferior performance compared to **RL-IRS-Base**.

It is important to consider the practical deployment of IRS systems. In practice, IRS elements can often be grouped together and controlled by the same circuit, resulting in the same phase shift parameter value being applied to the entire group at each time step. Under this realistic assumption, the proposed approach (**RL-IRS-FF**) can be adapted to estimate the shared phase shift value for each group of phase shifters controlled by the same underlying circuit.

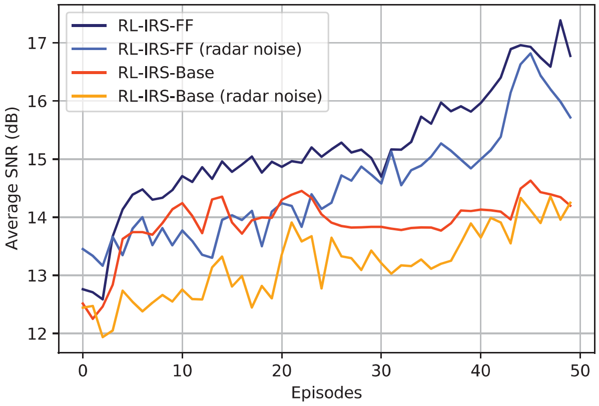


Fig. 8. The blue line and the orange line correspond to the performances of **RL-IRS-FF** and **RL-IRS-Base**, respectively, under perfect knowledge of the destination position. The light blue and light orange lines correspond to the performances of **RL-IRS-FF** and **RL-IRS-Base**, respectively, under imprecise knowledge of the destination position (induced by the finite range and angle resolution of the radar perception system). Each curve corresponds to the training for 50 episodes and each episode is comprised by 300steps. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

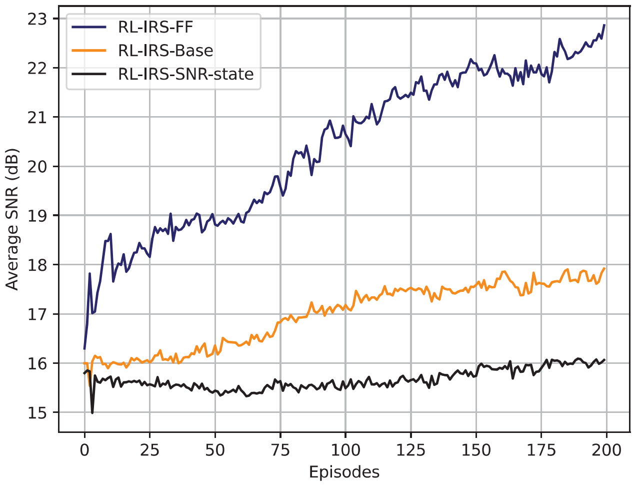


Fig. 9. Curves for the training performance of the 3 discussed algorithms, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state** for IRS with $\mathbf{150}$ phase shift elements. Each plot corresponds to the training for 200 episodes and each episode is comprised by 1000steps. The range of the destination motion is 25grid cells. Each curve is the average over 10different seeds and we omit the variance to avoid clutter.

J. Training Without Target Networks

The target network, as introduced by [39], has been widely recognized as a crucial component for the success of deep Q learning. However, it requires careful tuning of the hyperparameter τ to ensure effective target network updates. To shed light on the impact of the target network, we present Fig. 10, which showcases the performance of the discussed approaches in the same scenario as Fig. 9, but without utilizing the target network to estimate the targets for the critic updates.

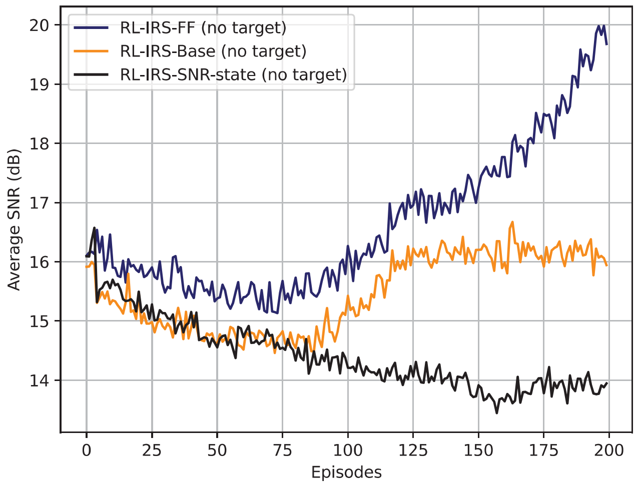


Fig. 10. Curves for the training performance of the 3 discussed algorithms, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state** for IRS with **150** phase shift elements, but without the utilization of the target network for the critic updates. Each plot corresponds to the training for 200 episodes and each episode is comprised by 1000 steps. The range of the destination motion is 25 grid cells. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

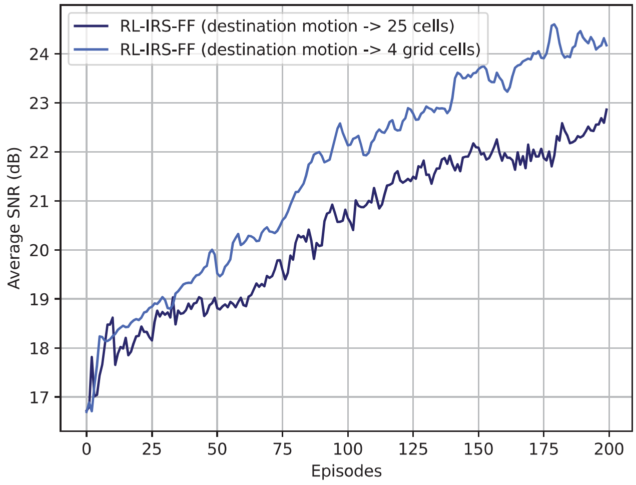


Fig. 11. Curves for the training performance of **RL-IRS-FF** for 2 different ranges of destination motion (namely 25 grid cells and 4 grid cells). The IRS is comprised by **150** phase shift elements. Each plot corresponds to the training for 200 episodes and each episode is comprised by 1000 steps. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

From the results, it is evident that both **RL-IRS-Base** and **RL-IRS-SNR-state** exhibit significant divergence in the absence of the target network. In contrast, the **RL-IRS-FF** approach manages to converge to a satisfactory performance, albeit at a slower rate compared to the experiment involving the target network (Fig. 9).

K. Comparing for Different Ranges of Destination Motion

A crucial aspect to consider is how the **RL-IRS-FF** approach performs under different ranges of destination motion. In Fig. 11, we present the performance of **RL-IRS-FF** in the same experimental setup as in Fig. 9, but for two distinct ranges of destination motion: 25 grid cells and 4 grid cells.

There are a couple of noteworthy observations. First, the plot line corresponding to 4 grid cells appears smoother. This is expected because, under similar channel conditions, less mobility of the destination leads to reduced variability in the optimum SNR per time step. Overall, the performance in the case of 4 grid cells is superior to that of 25 grid cells, and the convergence speed is also better for the former. This can be attributed to the fact that less mobility of the destination reduces the need for exploration by the RL algorithm, and it is well-known that exploration plays a significant role in determining performance [55].

VI. Why Does the Inclusion of the SNR in the State Causes Divergence?

In order to study the divergence caused by the inclusion of the SNR in the state representation we first have to discuss about divergence in deep value learning.

A. Divergence in Neural Value Learning

We retreat to the general case of off-policy deep Q learning. The critic is parametrized by parameter vector $\mathbf{w}$ and is denoted as $Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})$. Let us assume that we sample a transition that involves the state-action pair $(\mathbf{s}, \mathbf{a})$ from the Experience Replay and update $\mathbf{w}$:

$$\mathbf{w} = \mathbf{w} + \eta\, Q^*_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) - Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) \quad (24)$$

We examine how the aforementioned update affects the value estimate of a different state-action pair $(\bar{\mathbf{s}}, \bar{\mathbf{a}}) \neq (\mathbf{s}, \mathbf{a})$ by applying the Taylor expansion of Q around $\mathbf{w}$ and keeping only the first order term.

$$Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}) \approx Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}) + \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}})^T (\mathbf{w} - \mathbf{w}) \quad (25)$$

We plug Eqs. (24) into (25) and we obtain:

$$Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}) \approx Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}) + \eta K_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}; \mathbf{s}, \mathbf{a}) \times Q^*_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) - Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a}), \quad (26)$$

where $K_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}; \mathbf{s}, \mathbf{a}) = \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}})^T \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}, \mathbf{a})$ is the element of the Neural Tangent Kernel (NTK) of the critic network [56].

Let us also assume an MDP that corresponds to a discrete state-action space with N_{MDP} state-action pairs. All N_{MDP} state-action pairs are included in the contents of the Experience Replay. If we update $\mathbf{w}$ by applying the update rule defined in Eq. (11) with all N_{MDP} state-action pairs, then the updated vector of state-action values, $\mathbf{Q}_{\mathbf{w}}$ can be expressed as:

$$\mathbf{Q}_{\mathbf{w}} \approx \mathbf{Q}_{\mathbf{w}} + 2\eta \mathbf{K}_{\mathbf{w}} \mathbf{D}_{\rho} (\mathbf{Q}^*_{\mathbf{w}} - \mathbf{Q}_{\mathbf{w}}), \quad (27)$$

where $\mathbf{D}_{\rho}$ is a diagonal matrix with entries given by $\rho(s, a)$, the distribution induced by the content of the Experience Replay. The $\mathbf{K}_{\mathbf{w}}$ is the NTK of the critic and corresponds to a symmetric matrix where:

- $\mathbf{K}_{\mathbf{w}}(i, j) = K_{ij} = \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_i, \mathbf{a}_i)^T \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_j, \mathbf{a}_j)\ i = j$
- $\mathbf{K}_{\mathbf{w}}(i, i) = K_{ii} = \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_i, \mathbf{a}_i)\,{}^2_2$

At this point, we refer to the work by [35]. The authors make the following assumption. The first order approximation of the

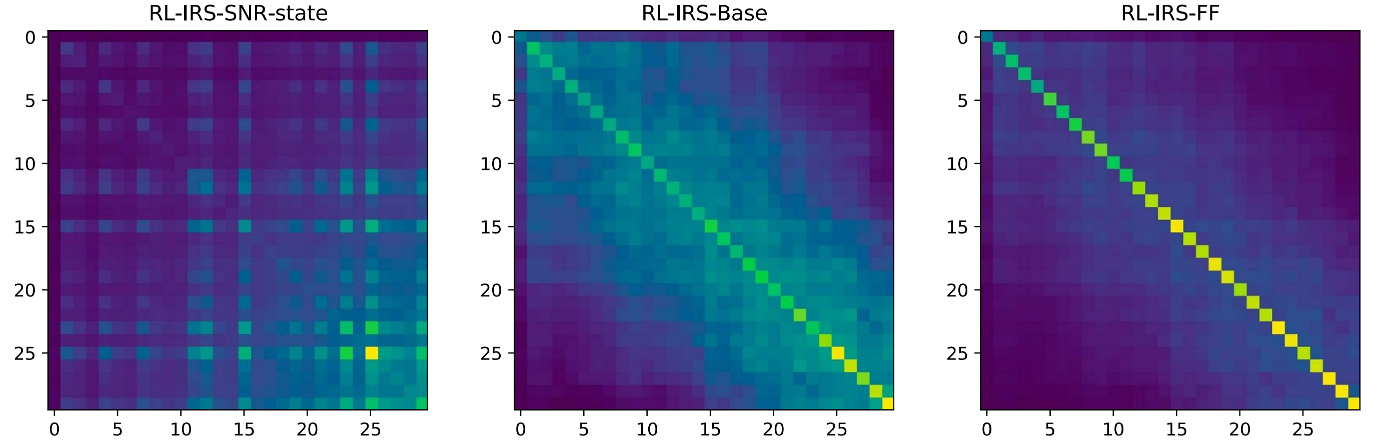


Fig. 12. The visualization of the NTK, for the same batch of experiences for the 3 proposed deep RL algorithms, namely **RL-IRS-FF**, **RL-IRS-Base**, **RL-IRS-SNR-state**. The states of the batch that are used for the NTK calculation of **RL-IRS-SNR-state** are augmented with the SNR at the destination.

critic update being a contraction in the sup norm is sufficient condition for convergence in deep value learning. Based on the said presupposition, they prove the following theorem:

Theorem 1: [35] Let indices i, j refer to state-action pairs. Suppose that $K_{\mathbf{w}}, \rho, \gamma < 1, \eta$ satisfy the following conditions:

$$\forall i,\ \ 2\eta K_{ii}\rho_i \leq 1, \tag{28}$$

$$\forall i,\ \ (1+\gamma) \underset{j=i}{\quad} K_{ij}\rho_j \leq (1-\gamma)K_{ii}\rho_i. \tag{29}$$

Then the critic update is a contraction in the sup norm and the fixed point of the update operator is the optimal value function of the MDP.

B. How the SNR at the State Affects the Critic NTK

Let us examine Theorem 1 in the context of designing the IRS phase shifters for the scenario described in Section II. The reward signal is not sparse, therefore, without the need for extensive exploration, we can safely assume that $\rho_i > 0$ everywhere. Moreover, the discount factor γ is chosen to be 0.99(very close to 1). Consequently, the aforementioned theorem implies that, in order to achieve convergence in the process of learning the value function of the IRS phase shift optimization MDP, we require that the resulting critic NTK should have a strong diagonal and small non-diagonal elements:

$$\forall j = i, \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_i, \mathbf{a}_i)^{\mathsf{T}} \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_j, \mathbf{a}_j) \quad \nabla_{\mathbf{w}} Q_{\mathbf{w}}(\mathbf{s}_i, \mathbf{a}_i)\ {}_2^2 \tag{30}$$

The non-diagonal elements of the critic NTK constitute a measure of the generalization of the critic. As it is prevalent by Eq. (26), the larger the absolute value of $K_{\mathbf{w}}(\bar{\mathbf{s}}, \bar{\mathbf{a}}; \mathbf{s}, \mathbf{a})$, the more the update using $(\mathbf{s}, \mathbf{a})$ affects the estimation of the value function for the $(\bar{\mathbf{s}}, \bar{\mathbf{a}})$ pair.

The NTK depends on both the architecture of the critic and the choice of the state and action vector representations. Therefore, the association between the divergent behavior of deep value learning and the NTK of the critic, outlined by expression (30), provides us with a tool to examine the effects of including the SNR as a component of the state.

Fig. 12 provides the visualization of the empirical NTK for the same batch of state-action pairs, at the beginning of training, for the three deep RL schemes that have been proposed in the previous subsection, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state**. What needs to be stressed regarding the aforementioned Figure is that the batch is exactly the same for the NTKs that correspond to **RL-IRS-FF** and **RL-IRS-Base**. The states of the batch for the calculation of the NTK that corresponds to **RL-IRS-SNR-state** are augmented with the SNRs at the destination. The exact same critic neural network was used at initialization with the exception that the critic of the **RL-IRS-FF** included the Fourier features preprocessing. All empirical NTKs were computed using tools from the PyTorch [57] repository introduced in [58].

As can be inferred by Fig. 12, the condition underlined in Eq. (30) is best satisfied by the NTK that corresponds to the **RL-IRS-FF**. It constitutes a stationary kernel with a very strong diagonal and relatively small non-diagonal elements. In that case, the critic gradient vectors for different state-action pairs are almost orthogonal to each other. This explains, to a large degree, the superior performance of **RL-IRS-FF** on the tested scenarios. Furthermore, the noteworthy ability of **RL-IRS-FF** to achieve satisfactory performance without relying on the conventional practice of using a frozen target network (as observed in Fig. 10) can be attributed to the inherent properties of its corresponding NTK. In particular, the NTK associated with **RL-IRS-FF** demonstrates minimal generalization by having significantly smaller off-diagonal elements compared to the elements on the main diagonal. This constitutes an indication that the success of the target network, as proposed in [39] is predicated on its contribution in mitigating aggressive generalization during Q updates. The performance discrepancy between **RL-IRS-FF** and **RL-IRS-Base** can also be explained by the NTK visualization. The NTK of **RL-IRS-Base** exhibits relatively larger non-diagonal elements in comparison to the NTK of **RL-IRS-FF** and, therefore, cannot estimate the value function as accurately. The NTK that corresponds to **RL-IRS-SNR-state** has elements of the main diagonal that are of small magnitude relative to the non-diagonal elements.

The sufficient condition for convergence underlined in Eq. (30) is significantly violated. This explains the divergent behavior of **RL-IRS-SNR-state**.

The inclusion of the SNR in the state representation induces a "state-aliasing" effect when combined with function approximation. Since the channels exhibit spatiotemporal correlations, two different states that correspond to different previous sequences of IRS phase shift values and different positions of the receiver might result in similar (or even the same) SNR at the destination. Therefore, the inclusion of the SNR as a state component makes the critic gradient vectors for different state-action pairs relatively coherent to each other. In such cases, the critic generalizes relatively aggressively and the value learning process is prone to diverge. Besides since the critic is trained to estimate the optimal value function, assuming good convergence of the value approximation, the effects of the channels on control performance can be implicitly inferred.

It is important to note the absence of channels (base station-IRS and IRS-destination) in the state representation, despite their assumed knowledge during training for reward/SNR computation. Similar to the SNR, the channels exhibit temporal and spatial correlation. Consequently, different time steps and destination positions can correspond to the same channel realizations, leading to gradient aliasing on the critic and violating the sufficient conditions for convergence.

C. Fixed Destination Position

The analysis above raises the question of whether fixing the position of the destination throughout training leads to divergence. Fig. 13 depicts the performance of the discussed algorithms in a setup identical to Fig. 9, except that the position of the destination remains fixed for all steps.

As expected, the variability of the reward between episodes is smaller compared to the case where the destination is moving (Fig. 9). Notably, the **RL-IRS-FF** method outperforms its performance when the destination is moving. Regarding the **RL-IRS-Base** and **RL-IRS-SNR-state** methods, both exhibit slight improvements compared to their performance with a mobile destination. However, the difference between the two methods is minimal, which can be attributed to the fixed destination causing an increase in gradient interference for both approaches. Although the performance of both methods is relatively poor compared to **RL-IRS-FF**, there is no significant increase in divergence.

The rationale behind this lies in the following explanation. Increasing gradient interference affects the performance of deep Q learning, but in the case of a fixed destination, this increase in gradient alignment (and thus generalization) is evenly distributed among all state-action pairs. On the other hand, when the SNR is included, aliasing occurs in a more stochastic manner. Consequently, a single update step may have a disproportionate effect on certain state-action pairs, leading to overestimation/underestimation in the value learning process, which is further amplified by bootstrapping. Not all increases in generalization have an equal negative impact. The inclusion of the SNR causes different state-action pairs to appear more similar than they actually are, resulting in overestimations and underestimations during training. Conversely, a fixed destination position increases generalization uniformly, reducing the likelihood of overestimations and underestimations. Alternatively, one can interpret this as follows. If we consider two different time steps that correspond to the same destination position, these two states contain similar information about the reward. As a result, updating the value estimation with one of them is expected to impact the estimation of the other. Conversely, two different time steps that correspond to the same SNR may not correspond to "similar" states due to the dynamic nature of the channels. Consequently, updating the value estimation with one of these states should not significantly influence the value estimation of the other.

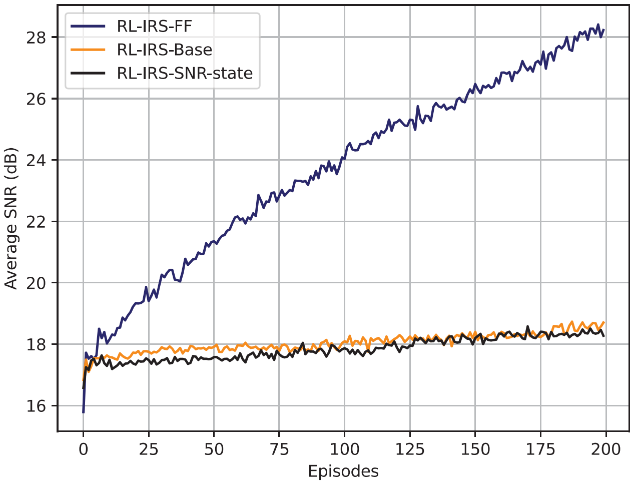


Fig. 13. Curves for the training performance of the 3 discussed algorithms, namely **RL-IRS-FF**, **RL-IRS-Base** and **RL-IRS-SNR-state** for IRS with 150 phase shift elements. Each plot corresponds to the training for 200 episodes and each episode is comprised by 1000steps. The position of the destination is fixed throughout training. Each curve is the average over 10 different seeds and we omit the variance to avoid clutter.

VII. The Critic NTK as a Tool for the Design of Deep RL Approaches

The availability of annotated data for wireless communications is very limited [59]. Consequently, deep RL algorithms have become very attractive for innovative solutions in the area of wireless systems [60], [61], [62], [63] since they are generally deprived of the need for ground truth labels. Besides the fact that the primary objective of the current article is to study the problem of IRS phase shift design in spatiotemporally correlated channel environments, there is also the motivation to incentivize the research community to utilize the properties of the NTK of the value network as a guide in the process of designing value-based deep RL algorithms for wireless communications. We consider that the connection between the generalization properties of the value network (that can be quantified via the value network's NTK) and the stability of neural value approximation is powerful and should be added in the quiver of paradigms for the development of deep RL algorithms in pursuit of wireless autonomy [64]. The design of deep RL methods for wireless communications (and for other domains) revolves around the choice for different state-action representations and different exotic neural network architectures a lot. This process can, sometimes, become chaotic and the best combination can be elusive. Eq. (30) along with the NTK visualization can become a single and reliable point of reference in order to both construct deep RL approaches and reason about their behavior in practice.

VIII. Conclusion

This article has considered the problem of dynamically controlling the values of IRS phase shifters over time in spatiotemporally correlated channel environments. In particular, the examined scenario involves a multi-antenna source and a single-antenna receiver that wish to communicate. The line-of-sight communication is blocked, and therefore an IRS is employed to reflect the signal of interest from the source to the receiver which can move within a confined area. The goal is to determine the phase shift values of the IRS at every time slot in order to maximize the sum of SNRs at the destination over an infinite time horizon. We have proposed a deep actor-critic algorithm that takes into account both the destination motion and the spatiotemporal evolution of the channels. The high variability of the channels with respect to both time and space induce high frequency components in the spectrum of the underlying value function of the defined MDP. Recent results in deep learning regression have demonstrated an impotence of feedforward neural networks in capturing high frequencies of the target function. We have thus proposed preprocessing the input of the critic with a Fourier features kernel to assist in the process of accurately estimating the value function. Our proposed approach has been seen to provide significant improvements in stability and reward accumulation. Finally, most previous works that have proposed deep RL for IRS phase shift optimization in similar settings have included the optimization metric (in our case the destination SNR) as a component of the MDP state. We have provided an analysis that hints at the fact that, for spatiotemporally varying channels, inclusion of SNR in the state representation can cause instability in the process of deep value learning. Our analysis has been predicated upon the coupling between convergence of deep Q learning and the properties of the NTK of the critic network. More concretely, including the SNR in the state increases the destructive generalization of the critic and, therefore, can cause instability and divergence.

References

[1] 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.

[2] L. Subrt and P. Pechac, “Intelligent walls as autonomous parts of smart indoor environments,” *IET Commun.*, vol. 6, no. 8, pp. 1004–1010, 2012.

[3] J. Zhao, “A survey of intelligent reflecting surfaces (IRSs): Towards 6G wireless communication networks,” 2019, *arXiv:1907.04789*.

[4] T. Jiang and Y. Shi, “Over-the-air computation via intelligent reflecting surfaces,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[5] X. Yu, D. Xu, Y. Sun, D. W. K. Ng, and R. Schober, “Robust and secure wireless communications via intelligent reflecting surfaces,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2637–2652, Nov. 2020.

[6] 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.

[7] X. Yang, C.-K. Wen, and S. Jin, “MIMO detection for reconfigurable intelligent surface-assisted millimeter wave systems,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1777–1792, Aug. 2020.

[8] Q. Wu and R. Zhang, “Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.

[9] 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.

[10] J. Zhu, Y. Huang, J. Wang, K. Navaie, and Z. Ding, “Power efficient IRS-assisted NOMA,” *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 900–913, Feb. 2021.

[11] X. Yu, D. Xu, and R. Schober, “MISO wireless communication systems via intelligent reflecting surfaces: (Invited Paper),” in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, 2019, pp. 735–740.

[12] Q. Wu and R. Zhang, “Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5394–5409, Nov. 2019.

[13] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] T. Jiang, H. V. Cheng, and W. Yu, “Learning to reflect and to beamform for intelligent reflecting surface with implicit channel estimation,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1931–1945, Jul. 2021.

[15] C. Huang, G. C. Alexandropoulos, C. Yuen, and M. Debbah, “Indoor signal focusing with deep learning designed reconfigurable intelligent surfaces,” in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*. Piscataway, NJ, USA: IEEE, 2019, pp. 1–5.

[16] A. Taha, M. Alrabeiah, and A. Alkhateeb, “Enabling large intelligent surfaces with compressive sensing and deep learning,” *IEEE Access*, vol. 9, pp. 44304–44321, 2021.

[17] B. Lim and S. Zohren, “Time-series forecasting with deep learning: A survey,” *Philos. Trans. Roy. Soc. A*, vol. 379, no. 2194, 2021, Art. no. 20200209.

[18] R. Wang et al., “Stationarity region of Mm-Wave channel based on outdoor microcellular measurements at 28 GHz,” in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, 2017, pp. 782–787.

[19] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[20] D. Bertsekas, *Reinforcement Learning and Optimal Control*. Belmont, MA, USA: Athena Scientific, 2019.

[21] C. Zhong et al., “Deep reinforcement learning-based optimization for IRS-assisted cognitive radio systems,” *IEEE Trans. Commun.*, vol. 70, no. 6, pp. 3849–3864, Jun. 2022.

[22] A. Taha, Y. Zhang, F. B. Mismar, and A. Alkhateeb, “Deep reinforcement learning for intelligent reflecting surfaces: Towards standalone operation,” in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2020, pp. 1–5.

[23] L. Wang, K. Wang, C. Pan, and N. Aslam, “Joint trajectory and passive beamforming design for intelligent reflecting surface-aided UAV communications: A deep reinforcement learning approach,” *IEEE Trans. Mobile Comput.*, early access, 2022.

[24] A. Feriani, A. Mezghani, and E. Hossain, “On the robustness of deep reinforcement learning in IRS-aided wireless communications systems,” 2021, *arXiv:2107.08293*.

[25] S. Rangan, P. Schniter, and A. K. Fletcher, “Vector approximate message passing,” *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6664–6684, Oct. 2019.

[26] S. Boyd et al., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends® Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[27] J. Lin, Y. Zout, X. Dong, S. Gong, D. T. Hoang, and D. Niyato, "Deep reinforcement learning for robust beamforming in IRS-assisted wireless communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. Piscataway, NJ, USA: IEEE, 2020, pp. 1–6.

[28] K. Feng, Q. Wang, X. Li, and C.-K. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 745–749, May 2020.

[29] M. Köppen, "The curse of dimensionality," in *Proc. 5th Online World Conf. Soft Comput. Ind. Appl.* (WSC5), 2000, vol. 1, pp. 4–8.

[30] 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.

[31] M. Shehab, B. S. Ciftler, T. Khattab, M. M. Abdallah, and D. Trinchero, "Deep reinforcement learning powered IRS-assisted downlink NOMA," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 729–739, 2022.

[32] N. Rahaman et al., "On the spectral bias of neural networks," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2019, pp. 5301–5310.

[33] S. Evmorfos, K. I. Diamantaras, and A. P. Petropulu, "Reinforcement learning for motion policies in mobile relaying networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 850–861, 2022.

[34] S. Evmorfos and A. P. Petropulu, "Deep actor-critic for continuous 3D motion control in mobile relay beamforming networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 5353–5357.

[35] J. Achiam, E. Knight, and P. Abbeel, "Towards characterizing divergence in deep Q-learning," 2019, *arXiv:1903.08894*.

[36] S. Evmorfos, A. P. Petropulu, and H. V. Poor, "Deep reinforcement learning for IRS phase shift design in spatiotemporally correlated environments," 2022, *arXiv:2211.09726*.

[37] Y. Li and A. Petropulu, "Dual-function radar-communication system aided by intelligent reflecting surfaces," 2022, *arXiv:2204.04721*.

[38] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[39] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[40] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[41] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2014, pp. 387–395.

[42] M. T. Spaan, "Partially observable Markov decision processes," in *Reinforcement Learning*. Berlin, Germany: Springer, 2012, pp. 387–414.

[43] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760–1776, Mar. 2020.

[44] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn*. PMLR, 2018, pp. 1587–1596.

[45] S. Evmorfos, K. Diamantaras, and A. Petropulu, "Deep Q learning with Fourier feature mapping for mobile relay beamforming networks," in *Proc. IEEE 22nd Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2021, pp. 126–130.

[46] M. Tancik et al., "Fourier features let networks learn high frequency functions in low dimensional domains," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 7537–7547, 2020.

[47] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[48] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 7462–7473.

[49] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[50] R. W. Heath, *Introduction to Wireless Digital Communication: A Signal Processing Perspective*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2017.

[51] D. S. Kalogerias and A. P. Petropulu, "Spatially controlled relay beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 24, pp. 6418–6433, Dec. 2018.

[52] D. S. Kalogerias and A. P. Petropulu, "Spatially controlled relay beamforming: 2-stage optimal policies," 2017, *arXiv:1705.07463*.

[53] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[54] X. Gao, G. Xing, S. Roy, and H. Liu, "Experiments with mmWave automotive radar test-bed," in *Proc. 53rd Asilomar Conf. Signals, Syst., Comput.*, 2019, pp. 1–6.

[55] R. McFarlane, "A survey of exploration strategies in reinforcement learning," McGill University, Montreal, Canada, 2018. [Online]. Available: https://www.cs.mcgill.ca/~cs526/roger.pdf

[56] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 8580–8589.

[57] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32, pp. 8026–8037.

[58] A. Engel, Z. Wang, A. D. Sarwate, S. Choudhury, and T. Chiang, "TorchNTK: A library for calculation of neural tangent kernels of PyTorch models," 2022, *arXiv:2205.12372*.

[59] I. Khan et al., "A data annotation architecture for semantic applications in virtualized wireless sensor networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*. Piscataway, NJ, USA: IEEE, 2015, pp. 27–35.

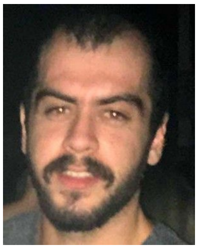
[60] K. K. Nguyen, A. Masaracchia, V. Sharma, H. V. Poor, and T. Q. Duong, "RIS-assisted UAV communications for IoT with wireless power transfer using deep reinforcement learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 5, pp. 1086–1096, Aug. 2022.

[61] R. Zhong, Y. Liu, X. Mu, Y. Chen, and L. Song, "AI empowered RIS-assisted NOMA networks: Deep learning or reinforcement learning?" *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 182–196, Jan. 2022.

[62] J. Hu, H. Zhang, L. Song, Z. Han, and H. V. Poor, "Reinforcement learning for a cellular Internet of UAVs: Protocol design, trajectory control, and resource management," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 116–123, Feb. 2020.

[63] X. Jia and X. Zhou, "IRS-assisted ambient backscatter communications utilizing deep reinforcement learning," *IEEE Wireless Commun. Lett.*, vol. 10, no. 11, pp. 2374–2378, Nov. 2021.

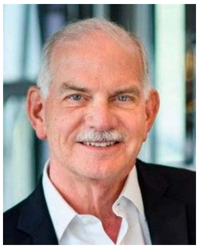
[64] Z. Zhang et al., "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, Sep. 2019.



Spilios Evmorfos received the M.Eng. degree in electrical and computer engineering (ECE) from the National Technical University of Athens (NTUA), in 2018. From 2018 to 2020, he worked as a Junior Researcher with the Institute of Communication and Computer Systems (ICCS) with specialization in machine learning. Since 2020, he has been working toward the Ph.D. degree in electrical and computer engineering with Rutgers, The State University of New Jersey, under the supervision of Prof. Athina Petropulu. During the summers of 2022 and 2023, he worked as a Research Scientist Intern at Siemens Technology's Autonomous Systems and Control Group, Princeton, NJ, USA. He was the recipient of the 2023 IEEE International Workshop on Machine Learning for Signal Processing (MLSP) Best Student Paper Award.

Athina P. Petropulu (Fellow, IEEE) is Distinguished Professor with the Electrical and Computer Engineering (ECE) Department at Rutgers, having served as the Chair of the Department during 2010–2016. Prior to joining Rutgers, she was a Professor in ECE with Drexel University, in 1992–2010. She held Visiting Scholar appointments at SUPELEC, Université Paris Sud, Princeton University, and the University of Southern California. Her research interests span the areas of statistical signal processing, wireless communications, signal processing in networking, physical layer security, and radar signal processing. Her research has been funded by various government industry sponsors, including the National Science Foundation (NSF), the Office of Naval research, the U.S. Army, the National Institute of Health, the Whitaker Foundation, Lockheed Martin, and

Raytheon. She is a fellow of the American Association for the Advancement of Science (AAAS), and the 2022–2023 President of the IEEE Signal Processing Society. She was a recipient of the 1995 Presidential Faculty Fellow Award given by NSF and the White House. She has served as the Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2009–2011) and IEEE Signal Processing Society Vice President-Conferences (2006–2008). She was a Technical Program Co-Chair of the 2023 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), a General Co-Chair of the 2018 IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), and a General Chair of the 2005 ICASSP. She has been a Distinguished Lecturer for the Signal Processing Society and the IEEE Aerospace and Electronics Systems Society. She was a recipient of the 2012 IEEE Signal Processing Society Meritorious Service Award, and a co-recipient of the 2005 IEEE Signal Processing Magazine Best Paper Award, the 2020 IEEE Signal Processing Society Young Author Best Paper Award (B. Li), the 2021 IEEE Signal Processing Society Young Author Best Paper Award (F. Liu), the 2021 Barry Carlton Best Paper Award by IEEE Aerospace and Electronic Systems Society, the 2022 IEEE Sensor Array and Multichannel Signal Processing Workshop Best Student paper Award (Y. Li), the 2023 IEEE Machine Learning for Signal Processing Workshop Best Student paper Award (S. Evmorfos), and the 2023 Stephen O. Rice Prize Best Paper Award by the IEEE Communications Society.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University, in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning, and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the recent book *Machine Learning and Wireless Communications* (Cambridge University Press, 2022). Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.