Generative Diffusion Models for Resource Allocation in Wireless Networks

Yiğit Berkay Uslu, Samar Hadou, Shirin Saeedi Bidokhti, Alejandro Ribeiro
University of Pennsylvania
{ybuslu, selaraby, saeedi, aribeiro}@seas.upenn.edu

***Abstract*—This paper proposes a supervised training algorithm for learning stochastic resource allocation policies with generative diffusion models (GDMs). We formulate the allocation problem as the maximization of an ergodic utility function subject to ergodic Quality of Service (QoS) constraints. Given samples from a stochastic expert policy that yields a near-optimal solution to the constrained optimization problem, we train a GDM policy to imitate the expert and generate new samples from the optimal distribution. We achieve near-optimal performance through the sequential execution of the generated samples. To enable generalization to a family of network configurations, we parameterize the backward diffusion process with a graph neural network (GNN) architecture. We present numerical results in a case study of power control.**

***Index Terms*—wireless resource allocation, generative models, diffusion processes, graph neural networks**

I. Introduction

Most existing formulations and methods for optimal wireless resource allocation, whether classical or learning-based, seek deterministic solutions. In contrast, *optimal solutions* of many non-convex optimization problems (e.g., power control, scheduling) are *inherently probabilistic*, as the optimal solution may lie in the convex hull of multiple deterministic policies. By randomizing between multiple deterministic strategies, stochastic policies can achieve better performance by effectively convexifying the problem [1]. This phenomenon is also fundamental in multi-user information theory, where time sharing plays a critical role in achieving optimal performance across various communication channels [2]–[4]. In this work, we leverage diffusion models to learn generative solutions to stochastic network resource allocation problems.

Generative models (GMs) have shown significant success in generating samples from complex, multi-modal data distributions. Among the wide class of generative models including variational autoencoders (VAEs) and generative adversarial networks (GANs), *generative diffusion models* (GDMs) stand out for their capability of generating high-quality and diverse samples with stable training [5], [6]. GDMs convert target data samples (e.g., images) to samples from an easy-to-sample prior (e.g., isotropic Gaussian noise) by a forward (noising) process, and then learn a backward (denoising) process to transform the prior distribution back to the target data distribution.

A substantial body of the existing literature utilizes GDMs, and GMs in general, for generating domain-specific synthetic data and for data augmentation to enhance the machine-learning models in supervised and reinforcement learning tasks [7], [8]. Yet, research on the use of GMs for wireless network optimization, and GDMs in particular, is scant [9]–[13]. Concurrent works [14]–[19] propose generative model solvers for network optimization as a framework to learn solution distributions that concentrate the probability mass around optimal deterministic solutions. The generative process then converts random noise to high-quality solutions by eliminating the noise introduced in the forward process. However, the problem formulation in the aforementioned studies is deterministic and ignores the probabilistic nature of the optimal solution.

Our work is one of the first to imitate *stochastic* expert policies using GDMs. We emphasize the stochastic nature of certain network optimization problems where random solutions are not only essential for optimality but also are realized by leveraging iterative dual domain algorithms. In our approach, Quality of Service (QoS) near-optimality emerges through the sequential execution of solutions sampled from the optimal generated distribution. Moreover, we use a graph neural network (GNN) architecture as the backbone for the reverse diffusion process to enable learning families of solutions across network topologies. GNNs not only excel in learning policies from graph-structured data [20]–[23] but also exhibit desirable properties such as stability, transferability and permutation-equivariance [24], [25].

This paper tackles imitation learning of stochastic wireless resource allocation policies. A GDM policy is trained to match an optimal solution distribution to a constrained optimization problem from which an expert policy can sample (Section II and Section III). We utilize a GNN-parametrization to condition the generative diffusion process directly on the network graphs (Section IV). We evaluate the proposed GDM policy in a power control setup and demonstrate that the trained GDM policy closely matches the expert policy over a family of wireless networks (Section V).

II. Optimal Wireless Resource Allocation

We represent the channel state of a wireless (network) system with a matrix $\mathbf{H} \in \mathcal{H} \subseteq \mathbb{R}^{N\times N}$ and the allocation of corresponding resources with a vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{N}$. Given $\mathbf{H}$, the choice of resource allocation $\mathbf{x}$ determines several QoS metrics that we represent with an objective utility $f_0 : \mathcal{X} \times \mathcal{H} \mapsto \mathbb{R}$ and a constraint utility $\mathbf{f} : \mathcal{X} \times \mathcal{H} \mapsto \mathbb{R}^c$. We define an optimal resource allocation $\mathbf{x}^*(\mathbf{H})$ as the argument that solves the constrained optimization problem,

$$\tilde{\mathrm{P}}(\mathbf{H}) = f_0\big(\mathbf{x}^*(\mathbf{H}), \mathbf{H}\big) = \underset{\mathbf{x}\in\mathcal{X}}{\text{maximum}} \;\; f_0\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big), \quad \text{subject to} \;\; \mathbf{f}\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big) \geq \mathbf{0}. \tag{1}$$

In (1), we seek a resource allocation $\mathbf{x}^*(\mathbf{H})$ with the largest f_0 utility among those in which the components of the utility $\mathbf{f}$ are nonnegative. This abstract formulation encompasses channel and power allocation [26] in wireless networks (Section V) as well as analogous problems, in, e.g., point-to-point [27], MIMO [28], broadcast [29] and interference channels [30].

In most cases of interest, the utilities f_0 and $\mathbf{f}$ in (1) are not convex. For this reason, we introduce the convex relaxation in which optimization is over probability distributions of resource allocation variables and QoS is measured in expectation,

$$\mathrm{P}(\mathbf{H}) \;=\; \underset{\mathcal{D}_{\mathbf{x}}}{\text{maximum}} \;\; \mathbb{E}_{\mathcal{D}_{\mathbf{x}}}\Big[\, f_0\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big)\,\Big], \quad \text{subject to} \;\; \mathbb{E}_{\mathcal{D}_{\mathbf{x}}}\Big[\, \mathbf{f}\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big)\,\Big] \geq \mathbf{0}. \tag{2}$$

In (2), we search over stochastic policies $\mathcal{D}_{\mathbf{x}}$ that maximize the *expected* utility $\mathbb{E}_{\mathcal{D}_{\mathbf{x}}}[\, f_0(\mathbf{x}(\mathbf{H}), \mathbf{H})]$ while satisfying the *expected* constraint $\mathbb{E}_{\mathcal{D}_{\mathbf{x}}}[\mathbf{f}\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big)] \geq \mathbf{0}$ when the resource allocation $\mathbf{x}(\mathbf{H})$ is drawn from the distribution $\mathcal{D}_{\mathbf{x}}$. For future reference, we introduce

$\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H})$ to denote a distribution that solves (2). In $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$, the channel state $\mathbf{H}$ is given and allocations $\mathbf{x}$ are sampled.

The important point here is that the performance of stochastic policies is realizable through time sharing if we allocate resources in a faster time scale than QoS perception. Indeed, if we consider independent resource allocation policies $\mathbf{x}_\tau(\mathbf{H}) \sim \mathcal{D}_{\mathbf{x}}$ we have that for sufficiently large T,

$$\frac{1}{T}\sum_{\tau=1}^{T} f_0\big(\mathbf{x}_\tau(\mathbf{H}), \mathbf{H}\big) \approx \mathbb{E}_{\mathcal{D}_{\mathbf{x}}}\Big[\, f_0\big(\mathbf{x}(\mathbf{H}), \mathbf{H}\big)\Big], \tag{3}$$

with an analogous statement holding for the constraint utility $\mathbf{f}$. Since deterministic policies are particular cases of stochastic policies, we know that $\mathrm{P}(\mathbf{H}) \geq \tilde{\mathrm{P}}(\mathbf{H})$. In practice, it is often the case that $\mathrm{P}(\mathbf{H}) \gg \tilde{\mathrm{P}}(\mathbf{H})$ and for this reason, the stochastic formulation in (2) is most often preferred over the deterministic formulation in (1), [1]–[3].

A. Imitation Learning of Stochastic Policies

In this paper, we want to learn to imitate the stochastic policies that solve (2). Consider a distribution $\mathcal{D}_{\mathbf{H}}$ of channel states $\mathbf{H}$. For each realization $\mathbf{H}$, recall that the solution of (2) is the probability distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H})$. Separate from these optimal distributions, we consider a parametric family of conditional distributions $\mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) = \mathcal{D}_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H}; \boldsymbol{\theta})$ in which the channel state $\mathbf{H}$ is given, and resource allocation variables are drawn. Our goal is to find the conditional distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})$ that minimizes the expectation of the conditional KL-divergences $\mathrm{D}_{\mathrm{KL}}\big(\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}) \,\big\|\, \mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})\big)$,

$$\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) = \underset{\mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})}{\operatorname{argmin}} \; \mathbb{E}_{\mathcal{D}_{\mathbf{H}}}\Big[\, \mathrm{D}_{\mathrm{KL}}\big(\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}) \,\big\|\, \mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})\big)\Big], \tag{4}$$

In (4), the distributions $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ are given for all $\mathbf{H}$. The conditional distribution $\mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})$ is our optimization variable, which we compare with $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ through their KL divergence. KL divergences of different channel realizations are averaged over the channel state distribution $\mathcal{D}_{\mathbf{H}}$, which is also given. The optimal distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H}; \boldsymbol{\theta})$ minimizes the expected KL divergence among those representable by the parametric family $\mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})$.

To solve (4), we need access to the expert conditional distributions $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$. This is impossible in general because algorithms that solve (2) do not solve for $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ directly. Rather, algorithms that solve (2) generate samples $\mathbf{x}(\mathbf{H})$ drawn from the optimal distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ [22]. Thus, we recast the goal of this paper as learning to generate samples $\mathbf{x} \,|\, \mathbf{H}$ from the distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta})$ when we are given samples $\mathbf{x}(\mathbf{H})$ of the expert conditional distributions $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ with channel states generated according to $\mathcal{D}_{\mathbf{H}}$:

> **Problem 1:** Given samples $\mathbf{x}(\mathbf{H})$ drawn from the expert distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})\mathcal{D}_{\mathbf{H}} = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H})\mathcal{D}_{\mathbf{H}}$ [cf. (2)], we learn to generate samples $\mathbf{x} \,|\, \mathbf{H}$ drawn from the conditional distributions $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H}; \boldsymbol{\theta})$ [cf. (4)].

A solution of Problem 1 is illustrated in Fig. 2. For a given channel state realization $\mathbf{H}$, we show two-dimensional slices of *samples* of an optimal policy (in blue). As indicated by (1), these samples realize optimal QoS metrics for (2) if executed sequentially (Fig. 1). We train a generative diffusion model (Section III) that generates samples (in orange) that are distributed close to samples of an optimal distribution. When executed sequentially, the learned samples realize QoS metrics close to optimal values (Fig. 1). We underscore that neither the optimal distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})\mathcal{D}_{\mathbf{H}} = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H})\mathcal{D}_{\mathbf{H}}$ nor the parametric distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{x} \,|\, \mathbf{H}; \boldsymbol{\theta})$ is computed.

B. Learning in the Dual Domain & Policy Randomization

Most learning approaches to allocating resources in wireless systems contend with the deterministic policy formulation in (1), e.g., [11], [14]–[17], [21], [26], [31]. This is due in part to the use of deterministic learning parameterizations [21], [26], [31] but even recent contributions that propose diffusion models do so for deterministic policies [11], [14]–[18]. This is a well-known limitation that has motivated, e.g., state-augmented algorithms that leverage dual gradient descent dynamics to randomize policy samples [22], [23]. These algorithms generate trajectories of primal and dual iterates by operating on a convex hull relaxation of the Lagrangian for the original problem and iteratively solving a sequence of Lagrangian maximization subproblems. Each subproblem is an unconstrained, deterministic problem to which regular learning methods apply, and near-optimality and feasibility guarantees are established neither for individual primal iterates nor their averages, but only for the sequential execution of the generated policy iterates.

A shortcoming of state-augmented algorithms—and iterative dual domain algorithms in general—is that they incur a transient period where suboptimal policies are executed. Reducing the length of this transient period typically requires larger step sizes, which in turn introduces a trade-off with respect to solution optimality. Learning a generative model to sample from the stationary (optimal) policy distribution emerges as a promising approach for overcoming this trade-off. To the best of our knowledge, our paper is the first to develop and demonstrate imitation of stochastic policies that solve a constrained optimization problem with generative diffusion models.

III. Policy Generative Models

GDMs involve a forward and a backward diffusion process. The *forward* process defines a Markov chain of diffusion steps to progressively add random noise to data. For a given $\mathbf{H}$ and a data sample $\mathbf{x}_0 = \mathbf{x}(\mathbf{H})$ drawn from the expert distribution $\mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$, the forward chain follows

$$q(\mathbf{x}_k \,|\, \mathbf{x}_0; \mathbf{H}) = \mathcal{N}(\mathbf{x}_k; \sqrt{\bar{\alpha}_k}\mathbf{x}_0, (1 - \bar{\alpha}_k)\mathbf{I}), \tag{5}$$

where $\bar{\alpha}_k := \prod_{i=1}^{k} \alpha_i$, $\alpha_k := 1 - \beta_k$, and β_k is a monotonically increasing noise schedule, e.g., linear. For a sufficiently large K, (5) converts the data sample $\mathbf{x}_0$ into a sample that is approximately isotropic Gaussian distributed, i.e., $\mathbf{x}_K \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

The reverse process of (5) is approximated by a chain of Gaussian transitions with a parametrized mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and fixed variance $\sigma_k^2\mathbf{I}$,

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{k-1} \,|\, \mathbf{x}_k; \mathbf{H}) = \mathcal{N}\big(\mathbf{x}_{k-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_k, k; \mathbf{H}), \sigma_k^2\mathbf{I}\big). \tag{6}$$

A *backward* diffusion process samples $\mathbf{x}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively runs the backward chain in (6) for $k = K, \ldots, 1$. With reparametrization of (5) as $\mathbf{x}_k(\mathbf{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_k}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_k}\boldsymbol{\epsilon}$ [5], sampling $\mathbf{x}_{k-1} \sim p_{\boldsymbol{\theta}}(. \,|\, \mathbf{x}_k; \mathbf{H})$ amounts to updating

$$\mathbf{x}_{k-1} = \frac{1}{\sqrt{\alpha_k}}\left(\mathbf{x}_k - \frac{\beta_k}{\sqrt{1 - \bar{\alpha}_k}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_k, k; \mathbf{H})\right) + \sigma_k\mathbf{w}, \tag{7}$$

where $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_k, k; \mathbf{H})$ predicts the noise $\boldsymbol{\epsilon}$ added to $\mathbf{x}_0 \sim \mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$ from noisy sample $\mathbf{x}_k$ at timestep k.

An *optimal GDM-policy parametrization* $\boldsymbol{\theta}^*$ minimizes the $\mathbf{H}$-expectation of the DDPM loss function [5] given by

$$\mathcal{L}_{\mathrm{GDM}}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_0, k, \mathbf{H}, \boldsymbol{\epsilon}}\; \omega(k)\big\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\mathbf{x}_k(\mathbf{x}_0, \boldsymbol{\epsilon}), k; \mathbf{H}\big) - \boldsymbol{\epsilon}\big\|^2. \tag{8}$$

In (8), $\omega(k)$ is a time-dependent weighting function and the expectation is over random timesteps $k \sim \mathrm{Uniform}([1, K])$, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, expert (data) samples $\mathbf{x}_0 \sim \mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$, and conditioning networks $\mathbf{H} \sim \mathcal{D}_{\mathbf{H}}$.

We note that the DDPM loss in (8) is a variational upper bound on the expected KL divergence loss in (4), which becomes tight when $\boldsymbol{\theta} = \boldsymbol{\theta}^*$. Thus, running (7) with optimal parametrization $\boldsymbol{\epsilon}_{\boldsymbol{\theta}^*}$ for a given $\mathbf{H}$ generates samples from the expert conditional distribution, i.e., $\mathbf{x}_0 \sim \mathcal{D}_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}^*) = \mathcal{D}^*_{\mathbf{x}}(\mathbf{H}; \boldsymbol{\theta}) \approx \mathcal{D}^*_{\mathbf{x}}(\mathbf{H})$.

IV. GNN-Parametrizations for GDM Policies

We employ GNNs for GDM parameterization, as they are well-suited for processing network data, such as resource allocations. Moreover, GNNs inherently take graphs as input, making them a natural fit for GDMs conditioned on $\mathbf{H}$.

GNNs process graph data through a cascade of L graph convolutional network (GCN) layers [32]. Inputs are node signals (features) and graph shift operators (GSO) while outputs are node embeddings. Each GCN layer $\mathbf{\Psi}^{(\ell)}$ is a nonlinear aggregation function obtained by the composition of a graph convolutional filter and a pointwise nonlinearity φ (e.g., relu),

$$\mathbf{Z}^{(\ell)} = \mathbf{\Psi}^{(\ell)}\left(\mathbf{Z}^{(\ell-1)}; \mathbf{H}, \Theta^{(\ell)}\right) = \varphi\left[\sum_{m=0}^{M_\ell} \mathbf{H}^m \mathbf{Z}^{(\ell-1)} \mathbf{\Theta}_m^{(\ell)}\right]. \quad (9)$$

In (9), $\Theta^{(\ell)} = \{\mathbf{\Theta}_k^{(\ell)} \in \mathbb{R}^{F_{\ell-1}\times F_\ell}\}_{k=0}^K$ is a set of learnable weights, M_ℓ denotes the number of hops, and $\mathbf{Z}^{(\ell-1)} \in \mathbb{R}^{N\times F_{\ell-1}}$ is the input node signal to layer ℓ. The GSO, $\mathbf{H}$, encodes the underlying connectivity of the network, which is the network state in our case.

For improved and more stable training, we take advantage of normalization layers and residual connections. To this end, we redefine φ in (9) as the composition of a normalization layer followed by a pointwise nonlinearity, while the first term in the sum, $\mathbf{Z}\mathbf{\Theta}_0$, inherently represents a learnable residual connection.

We view $\mathbf{x}_k$ and $\mathbf{k} = k\mathbf{1}_N$ as node signals and introduce a read-in layer $\mathbf{\Phi}^{(0)} = (\mathbf{\Phi}_\mathbf{x}, \mathbf{\Phi}_\mathbf{t})$ that adds sinusoidal-time embeddings to the input node features. That is, we have

$$\mathbf{Z}^{(0)} = \mathbf{\Phi}^{(0)}(\mathbf{x}_k, \mathbf{k}) = \mathbf{\Phi}_\mathbf{x}(\mathbf{x}_k) + \mathbf{\Phi}_\mathbf{k}(\mathbf{k}), \quad (10)$$

where $\mathbf{\Phi}_\mathbf{x} : \mathbb{R}^N \mapsto \mathbb{R}^{N\times F_0}$ is a multilayer perceptron (MLP) layer, and $\mathbf{\Phi}_\mathbf{t} : \mathbb{R}^N \mapsto \mathbb{R}^{N\times F_0}$ is a cascade of a sinusoidal time embedding and MLP layers. Finally, we add a readout MLP layer $\mathbf{\Phi}^{(L)} : \mathbb{R}^{N\times F_L} \mapsto \mathbb{R}^N$ that learns to predict the noise $\boldsymbol{\epsilon}$ from the output node embeddings.

V. Case Study: Power Control in Multi-User Interference Networks

We consider the problem of power control in N-user interference channels. Similar setups have been investigated in [22], [23] and should be referred to for more details.

A. Wireless Network & Power Control Setup

To summarize the setup briefly, all network realizations are sampled from a family of network configurations with $N = 100$ transmitters-receiver (tx-rx) pairs (also nodes in our graphs) and an average density of 12 tx-rx pairs/km^2. For each network, we first drop the transmitters randomly in a square grid world, and each transmitter is paired with a neighboring receiver. Signals coming from all but their respective transmitters are treated as interference by the receivers.

We optimize the transmit power levels $\mathbf{x} \in [0, P_{\max}]^N$ where $P_{\max} = 10$ mW is the maximum transmit power budget. The channel bandwidth and noise power spectral density (PSD) are set to $W = 20$ MHz and $N_0 = -174$ dBm/Hz, respectively. The network state $\mathbf{H}$ is the matrix of long-term channel gains which follow a log-normal shadowing with a standard deviation of 7 plus the standard dual-slope path-loss model. For a given $\mathbf{H}$, the short-term (instantaneous) channel gains $\widetilde{\mathbf{H}}$ vary following Rayleigh fading. To evaluate the performance of a policy $\mathbf{x}$, we define the instantaneous rate of receiver i as

$$\widetilde{r}_i(\mathbf{x}, \widetilde{\mathbf{H}}) = \log_2\left(1 + \frac{x_i \cdot |\widetilde{h}_{ii}|^2}{WN_0 + \sum_{j\neq i} x_j \cdot |\widetilde{h}_{ji}|^2}\right), \quad (11)$$

where x_i is the ith component of $\mathbf{x}$ and $\widetilde{h}_{ji}$ is the (j,i)th element in matrix $\widetilde{\mathbf{H}}$. Observe that a policy $\mathcal{D}_\mathbf{x}(\mathbf{H})$ is determined only by the long-term gains, whereas the instantaneous rate depends on the short-term channel gains. Ergodic rates are defined as

$$\mathbf{r}(\mathcal{D}_\mathbf{x}(\mathbf{H}), \mathbf{H}) := \mathbb{E}_{\mathcal{D}_\mathbf{x}(\mathbf{H}),\, \widetilde{\mathbf{H}}|\mathbf{H}}\left[\widetilde{\mathbf{r}}\big(\mathbf{x}, \widetilde{\mathbf{H}}\big)\right]. \quad (12)$$

In (12), we take an expectation over the policy and the fading jointly. In our experiments, we draw 200 samples from the trained GDM policy for each network and evaluate the joint expectation over 200 time steps, with each time step spanning 10 ms.

A minimum ergodic rate requirement of $f_{\min} = 0.6$ bps/Hz is imposed for all receivers by setting the utility constraints $\mathbf{f}(\mathcal{D}_\mathbf{x}(\mathbf{H}), \mathbf{H}) := \mathbf{r}(\mathcal{D}_\mathbf{x}(\mathbf{H}), \mathbf{H}) - \mathbf{1}_N f_{\min}$ whereas the utility objective is the network-wide average of the ergodic rates given by $f_0(\mathcal{D}_\mathbf{x}(\mathbf{H}), \mathbf{H}) := \mathbf{1}_N^\top \mathbf{r}(\mathcal{D}_\mathbf{x}(\mathbf{H}), \mathbf{H})/N$.

B. State-Augmented Primal-Dual Expert Policy & Baselines

To generate samples from an optimal solution distribution, we first train a GNN-parametrized model via a state-augmented primal-dual (SA) learning algorithm as in [23]. The trained model is executed online for each given network $\mathbf{H}$ over a sufficiently long time window to generate trajectories of primal and dual iterates with near-optimality and feasibility guarantees. We collect the resulting primal iterates $\{\mathbf{x}_b^\dagger(\mathbf{H})\}_{b=0}^{B-1}$, i.e., resource allocation vectors, in a buffer with capacity $B = 500$. During GDM policy training, expert policy samples are uniformly drawn from the buffer, i.e., we have $\mathcal{D}_\mathbf{x}^*(\mathbf{H}) = \text{Uniform}\big[\{\mathbf{x}_0^\dagger(\mathbf{H}), \ldots, \mathbf{x}_{B-1}^\dagger(\mathbf{H})\}\big]$ for all $\mathbf{H} \sim \mathcal{D}_\mathbf{H}$. Note that while we parametrize the expert policy and run a state-augmented training algorithm over a training dataset of networks $\mathcal{D}_\mathbf{H}$, one can alternatively run the dual descent without the parametrization and store the resource allocation iterates for each instance $\mathbf{H} \sim \mathcal{D}_\mathbf{H}$.

We compare the GDM and expert policies with two baseline deterministic policies:

1) *Average-power transmission policy*: For each given $\mathbf{H}$, we compute the time-average of the expert policy samples–similar to primal averaging in the optimization literature–and fix it, i.e., we set $\mathbf{x}(\mathbf{H}) \approx \mathbb{E}_{\mathcal{D}_\mathbf{x}^*(\mathbf{H})}[\mathbf{x}^\dagger(\mathbf{H})]$ at all time steps.
2) *Full-power transmission policy (FP)*: All transmitters use all the transmission power available at all time steps, i.e., $\mathbf{x}(\mathbf{H}) = P_{\max}\mathbf{1}_N$.

C. Implementation & Training Details for GDM Policy

For the GNN-parametrizations, the GSO representation of the network state $\mathbf{H}$ is a fully connected graph where nodes correspond to transmitter-receiver pairs, and the edge weights between nodes i and j, e_{ij}, are set to log-normalized long-term channel gains given by $e_{ij} \propto \log_2\left(1 + \frac{P_{\max}|h_{ij}|^2}{WN_0}\right)$. The GNN has $L = 6$ layers, each with $F_\ell = 128$ hidden features and $M_\ell = 2$ hops (filter taps). We set the number of diffusion time steps to $K = 500$, use a cosine noise schedule β_k [33] and train the GDM policy to minimize the DDPM objective in (8) with a log-SNR weighting function $\omega(k) = \log\big(\text{SNR}(k)\big)$, where $\text{SNR}(k) := \alpha_k^2/\sigma_k^2$.

To evaluate our method, we draw a total of 128 network realizations. Following a $5:1:2$ split, we obtain training, validation, and test datasets of size $|\mathcal{D}_\mathbf{H}| = 80$, $|\mathcal{V}_\mathbf{H}| = 16$, and $|\mathcal{T}_\mathbf{H}| = 32$ networks, respectively. The expert policy solutions are obtained across all three datasets for evaluation and benchmarking purposes. We train the GDM policy over the training dataset $\mathcal{D}_\mathbf{H}$ for a maximum of 10^4 epochs with an ADAM optimizer [34], an initial learning rate of 10^{-2}, and a learning rate schedule that follows a cosine decay with warm restarts. In each epoch, we iterate over the whole dataset with mini-batches of 16 graphs and sample 250 graph signals, i.e., expert policy data $\mathbf{x}(\mathbf{H}) \sim \mathcal{D}_\mathbf{x}^*(\mathbf{H})$, for each graph in the batch. Every 200 epochs, we evaluate the checkpointed GDM model on the validation dataset $\mathcal{V}_\mathbf{H}$ and save the best model in terms of 5th percentile ergodic rates. We test the saved model on $\mathcal{T}_\mathbf{H}$.

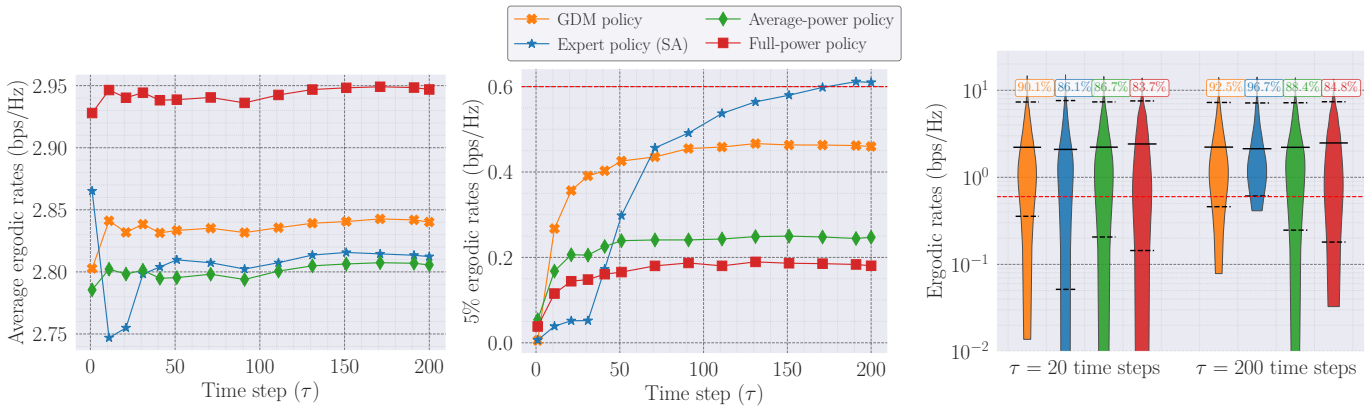

Fig. 1: Comparison of the test performance of GDM policy with the expert policy (SA) and other baselines. Leftmost and middle plots show the time evolution of the average and 5th percentile of ergodic rates, respectively. The minimum rate requirement $f_{\min}$ is shown with a dashed, red line. Rightmost plot shows the distribution of ergodic rates and constraint satisfaction percentages, evaluated up to $\tau = 20$ and $\tau = 200$ time steps. Solid black lines are drawn at the median values while the dashed black lines are for the 5th and 95th percentile values.

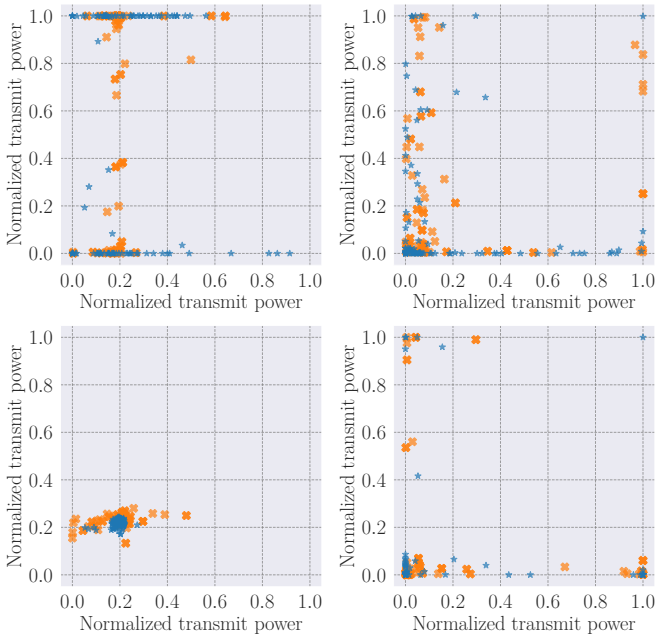

Fig. 2: Example two dimensional slices of expert policy (blue) and GDM policy (orange) samples are shown for two pairs of neighboring nodes from the test dataset. Transmit powers are normalized by $P_{\max}$.

We apply an affine transform $[0, P_{\max}] \mapsto [-1/2, 1/2]$ to map the policy space to a centered diffusion space. To sample from the diffusion model, we run the DDPM sampling equation (7) with standardized variables, invert the affine transform, and project the generated policy samples to the support $[0, P_{\max}]^N$.[1]

D. Performance of GNN-Parametrized GDM Policy

Fig. 1 showcases the test performance of the GDM policy, expert policy, and the baselines over a time horizon of 200 time steps. We estimate the ergodic rate vector for a given network $\mathbf{H}$ and time step τ as $(1/\tau)\sum_{s=0}^{\tau-1} \widetilde{\mathbf{r}}\big(\mathbf{x}_s, \widetilde{\mathbf{H}}\big)$ where $\mathbf{x}_s$ denotes the power allocation decision at an earlier time step $s < \tau$ [c.f. (12)]. In the first two plots, we report the time evolution of the average ergodic rate, i.e., the mean-rate objective utility, and the 5th-percentile of ergodic rates, both of which are computed across all $|\mathcal{T}_h| \times N = 32 \times 100 = 3200$ tx-rx pairs in the test dataset. The rightmost violin plot shows the histogram of ergodic rates of all receivers evaluated at two different time steps. We verify that the expert policy eventually satisfies almost all the constraints unlike the baselines. Although the GDM policy does not exhibit strict feasibility, it converges to a near-feasible and near-optimal policy in very few time steps compared to the expert policy and does not incur a long transient period. Moreover, both the full-power and average-power baselines are outperformed by the GDM policy in terms of percentile rates and constraint violations.

In Fig. 2, example two-dimensional (2D) slices of 100-dimensional learned GDM policies are overlaid with expert policy samples. The GDM policy generalizes to unseen test networks drawn from $\mathcal{T}_{\mathbf{H}}$, and the conditional distribution of GDM policy samples significantly resembles that of the expert policy. A peculiarity of optimal power control policies is that they tend to be probabilistic and involve multiple transmission modes. That is especially true for tx-rx pairs with less favorable channel conditions for which the policy randomization becomes more nuanced. In such cases, similar to time-sharing strategies, pairs that would otherwise generate considerable mutual interference adopt a policy-switching mechanism where they take turns to transmit at high power during periods of minimal interference (e.g., the top-left and bottom-right corners in the rightmost plot of Fig. 2). By alternating their transmissions, all tx-rx pairs satisfy their minimum ergodic rate requirements.

Strong test generalization evidenced in both figures notwithstanding, the GDM policy exhibits a small feasibility gap compared to the expert policy in Fig. 1. We attribute this gap primarily to the supervised training algorithm not accounting directly for the sensitivity of the constraints and the aforementioned policy-switching phenomenon. The feasibility gap and overall performance of the GDM policies can be further improved by incorporating the QoS requirements and additional variance constraints directly into the training loss and/or generative process.

VI. Conclusion

This work demonstrated that generative diffusion processes can imitate expert policies that sample from optimal solution distributions of stochastic network optimization problems. We employed a GNN to condition the generative process on a family of wireless network graphs. More broadly, we anticipate that our attempt at generative diffusion-based sampling of random graph signals will be of interest beyond resource optimization in wireless networks. We leave constraint-aware, unsupervised training of GDM policies across a wider range of network topologies as future research directions.

[1]During inference, we observed negligible difference in the quality of generations when we swapped the DDPM sampler with other samplers, e.g., a DDIM sampler or its accelerated counterpart with fewer denoising time steps.

REFERENCES

[1] Michael Neely, *Stochastic network optimization with application to communication and queueing systems*, Morgan & Claypool, 2010.

[2] R. Gallager, "A perspective on multiaccess channels," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 124–142, 1985.

[3] I. Sason, "On achievable rate regions for the Gaussian interference channel," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1345–1356, 2004.

[4] Qing He, Di Yuan, and Anthony Ephremides, "Optimal scheduling for emptying a wireless network: Solution characterization, applications, including deadline constraints," *IEEE Transactions on Information Theory*, vol. 66, no. 3, pp. 1882–1892, 2020.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of CVPR*, June 2022, pp. 10684–10695.

[7] Ali Kasgari, Walid Saad, Mohammad Mozaffari, and H. Vincent Poor, "Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 884–899, 2020.

[8] Wafa Njima, Ahmad Bazzi, and Marwa Chafii, "DNN-based indoor localization under limited dataset using GANs and semi-supervised learning," *IEEE Access*, vol. 10, pp. 69896–69909, 2022.

[9] Elhadj Moustapha Diallo, "Generative model for joint resource management in multi-cell multi-carrier NOMA networks," 5 2024.

[10] Yuxiu Hua, Rongpeng Li, Zhifeng Zhao, Xianfu Chen, and Honggang Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 334–349, 2019.

[11] Hongyang Du, Ruichen Zhang, Yinqiu Liu, Jiacheng Wang, Yijing Lin, Zonghang Li, Dusit Niyato, Jiawen Kang, Zehui Xiong, Shuguang Cui, Bo Ai, Haibo Zhou, and Dong In Kim, "Enhancing Deep Reinforcement Learning: A Tutorial on Generative Diffusion Models in Network Optimization," *IEEE Communications Surveys and Tutorials*, 2024.

[12] Salar Nouri, Mojdeh Karbalaee Motalleb, and Vahid Shah-Mansouri, "Diffusion-RL for scalable resource allocation for 6G networks," 2025.

[13] Feiran You, Hongyang Du, Xiangwang Hou, Yong Ren, and Kaibin Huang, "DRESS: Diffusion reasoning-based reward shaping scheme for intelligent networks," 2025.

[14] Ruihuai Liang, Bo Yang, Zhiwen Yu, Bin Guo, Xuelin Cao, Mérouane Debbah, H. Vincent Poor, and Chau Yuen, "DiffSG: A Generative Solver for Network Optimization with Diffusion Model," 8 2024.

[15] Ruihuai Liang, Bo Yang, Pengyu Chen, Xianjin Li, Yifan Xue, Zhiwen Yu, Xuelin Cao, Yan Zhang, Mérouane Debbah, H. Vincent Poor, and Chau Yuen, "Diffusion models as network optimizers: Explorations and analysis," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[16] Amirhassan Babazadeh Darabi and Sinem Coleri, "Diffusion model based resource allocation strategy in ultra-reliable wireless networked control systems," *IEEE Communications Letters*, 2024.

[17] Xinyu Lu, Zhanbo Feng, Jiawei Sun, Jiong Lou, Chentao Wu, Wugedele Bao, and Jie Li, "Generative diffusion model-based energy management in networked energy systems," in *ICASSP*, 2025, pp. 1–5.

[18] Yifan Xue, Ruihuai Liang, Bo Yang, Xuelin Cao, Zhiwen Yu, Mérouane Debbah, and Chau Yuen, "Joint task offloading and resource allocation in low-altitude MEC via graph attention diffusion," 2025.

[19] Xudong Wang, Lei Feng, Jiacheng Wang, Hongyang Du, Changyuan Zhao, Wenjing Li, Zehui Xiong, Dusit Niyato, and Ping Zhang, "Graph diffusion-based AeBS deployment and resource allocation for RSMA-enabled URLLC low-altitude economy networks," 2025.

[20] Yifei Shen, Yuanming Shi, Jun Zhang, and Khaled B Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 101–115, 2020.

[21] Zhiyang Wang, Mark Eisen, and Alejandro Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.

[22] Navid NaderiAlizadeh, Mark Eisen, and Alejandro Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Transactions on Signal Processing*, 2022.

[23] Yiğit Berkay Uslu, Navid NaderiAlizadeh, Mark Eisen, and Alejandro Ribeiro, "Fast state-augmented learning for wireless resource allocation with dual variable regression," 2025.

[24] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro, "Graph neural networks: Architectures, stability, and transferability," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021.

[25] Lucia Testa, Claudio Battiloro, Stefania Sardellitti, and Sergio Barbarossa, "Stability of graph convolutional neural networks through the lens of small perturbation analysis," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 6865–6869.

[26] Fei Liang, Cong Shen, Wei Yu, and Feng Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1760–1776, 2019.

[27] Wei Yu, "Sum-capacity computation for the Gaussian vector broadcast channel via dual decomposition," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 754–759, 2006.

[28] Xingqin Lin, Robert W. Heath, and Jeffrey G. Andrews, "The interplay between massive MIMO and underlaid D2D networking," *IEEE Transactions on Wireless Communications*, vol. 14, no. 6, pp. 3337–3351, 2015.

[29] DN Tse, "Optimal power allocation over parallel Gaussian broadcast channels," in *Proc. of IEEE International Symposium on Information Theory*, 1997, p. 27.

[30] Krishna Chaitanya A, Utpal Mukherji, and Vinod Sharma, "Power allocation for interference channels," in *2013 National Conference on Communications (NCC)*, 2013, pp. 1–5.

[31] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nikos D Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *IEEE 18th SPAWC*, 2017, pp. 1–6.

[32] Fernando Gama, Antonio G Marques, Geert Leus, and Alejandro Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2018.

[33] Alexander Quinn Nichol and Prafulla Dhariwal, "Improved denoising diffusion probabilistic models," in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, PMLR.

[34] Diederik P Kingma and Jimmy Ba, "ADAM: A method for stochastic optimization," 2014.