Wireless Link Scheduling over Recurrent Riemannian Manifolds

Rashed Shelim, *Graduate Student Member, IEEE*, Ahmed S. Ibrahim, *Member, IEEE*

***Abstract*—Deep learning models for scheduling of potentially-interfering communication pairs, in device-to-device (D2D) settings, require large training samples in the order of hundreds to thousands. Some of the dynamic networks, such as vehicular networks, cannot tolerate the waiting time associated with gathering a large number of training samples. Spatio-temporal correlation among communication pairs in such networks can be utilized to reduce the learning phase. In this paper, we propose a Riemannian-geometric recurrent neural network (R-RNN) method based on statistical recurrent unit (SRU) for wireless link scheduling. First, we represent local graphs around each D2D pair in any finite time frame as a sequence of points on Riemannian manifold thanks to representing its topology as a symmetric positive definite (SPD) matrix. We compute the Riemannian metric, i.e., Stein metric, which are suitable measures of time-dependence among D2D pairs. Then we use the Stein metric in the proposed R-RNN method to forecast the link scheduling decisions for a finite number of successive time slots ahead. Simulation results reveal that the proposed method achieves promising performance against the state-of-the-arts with only 45 training samples.**

***Index Terms*—Dynamic D2D networks, Riemannian geometry, symmetric positive definite matrices, statistical recurrent unit, stein metric, wireless link scheduling.**

I. Introduction

Scheduling of interfering links in device-to-device (D2D) wireless networks is one of the fundamental problems in wireless communications. The D2D link scheduling problem can be extended to dynamic networks, such as vehicular communications [1], [2], that include mobile nodes. Link scheduling requires a judicious activation of a subset of D2D links to mitigate excessive interference. With the goal to maximize the information-theoretic sum rate, the link scheduling in D2D networks can be defined as a non-convex combinatorial optimization problem [3]–[5]. Conventional link scheduling solutions are typically based on sequential link selection algorithm [6], greedy heuristic search algorithm [7], [8], iterative fractional programming algorithm [9], or interference aware methods [10], [11]. Ma *et al.* [12] have proposed a shortest link scheduling problem under signal to interference plus noise ratio (SINR) model. Besides, a link scheduling problem has been addressed under Rayleigh fading model in [13]. On the other hand, machine learning (ML) based link scheduling approaches such as deep reinforcement learning (DRL) in [14], or convolutional neural network (CNN) in [15] have achieved a similar sum rate as optimization-based ones. However, all algorithms mentioned above require channel state information (CSI) estimation which is difficult to obtain in densely deployed D2D networks [5], [16], [17].

This work is supported in part by the National Science Foundation under award number CNS-2144297.

Rashed Shelim and Ahmed S. Ibrahim are with the Department of Electrical and Computer Engineering, Florida International University, Miami, FL-33174, USA

Corresponding author: Rashed Shelim (e-mail: rshel019@fiu.edu).

A. Related Works

Towards the sole utilization of spatial locations of D2D pairs, a spatial deep learning method has been applied in [4] for wireless link scheduling. In [18], a learning based link scheduling model has been proposed to satisfy user's quality of service (QoS) which utilizes the user's location. As hundreds of thousands training samples (i.e., wireless network layouts) are required to train the model of these methods (i.e., $800,000$ in [4] and $100,000$ in [18]), other deep neural networks (DNN) based schemes have been pursued. For instance, in [19], a DNN based link scheduling approach has been proposed, which needs $10,000$ training samples. Further, a DNN method that is based on graph embedding has been presented in [5] which reduces the number of training layouts to 500. However, all of these methods require large training samples in the order of hundreds to thousands to train the model. Dynamic settings, such as vehicular networks, cannot tolerate the waiting time associated with obtaining a large number of training samples. To address this issue, we have proposed two novel geometric machine learning (G-ML) methods in [20] where we have modeled local graph around each D2D pair as a point on the Riemannian manifolds and computed Log-euclidean metric (LEM) as an interference measure among D2D pairs. The proposed supervised G-SVM scheme in [20] requires only 90 training wireless network layouts.

B. Motivation

Movement of mobile users in dynamic D2D networks exhibits spatio-temporal correlation, which can be utilized to further reduce the training samples. Temporal movements of D2D nodes over a series of time slots can be modeled as a time series forecasting problem. To our context, time series forecasting involves taking ML model that learns from historical statistics (i.e., spatio-temporal correlation among D2D pairs at previous time slots) and using them to forecast link scheduling decisions. Statistical recurrent unit (SRU) introduced in [21] can be used in time series forecasting problem over Euclidean space. SRU is able to learn long-term dependencies in time series data by only keeping moving averages of statistics (i.e., *summary statistics*). SRU has been successfully applied in many time series applications, e.g., intelligent systems [22],

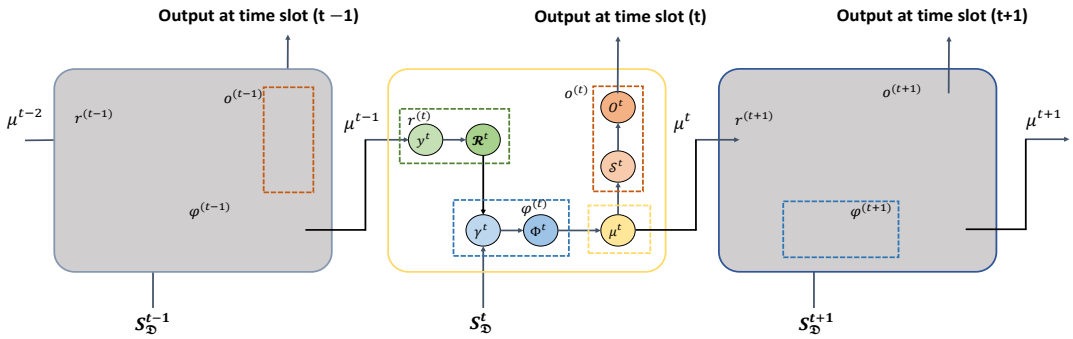

Fig. 1: The temporal behavior of Riemannian recurrent unit over time series. $\mathcal{Y}^t$, γ^t and $\mathcal{S}^t$ are performed using synaptic weights from recurrent unit. $\left(\mu^t\right)^{(\alpha)}$ is the summary statistics at time slot t. O^t is the output generated at time slot t.

pattern recognition [23], and fuel science [24]. Nonetheless, to the best of our knowledge, no prior research work has considered the spatio-temporal correlation in dynamic D2D networks to reduce the number of training samples for link scheduling.

C. Contributions

In this paper, we aim to use a statistical recurrent model over Riemannian manifolds to reduce the number of required training samples further than [20]. Riemannian geometry was recently used to address challenges in communication systems such as the design of beamforming codebooks in [25] or the deployment of relays in [26]. Local graph around each D2D pair for each network layout in the network layout series can be represented as a sequence of points over Riemannian manifolds [20], [27] using a set of regularized Laplacian matrices, which are Symmetric positive definite (SPD) one. SPD matrices can be represented as points over Riemannian manifolds [28], [29]. To this end, we propose a Riemannian-geometric recurrent neural network (R-RNN) method for link scheduling decisions which is based on the statistical recurrent unit on SPDs [30]. The proposed method uses Stein metric [31], [32] as a similarity measure in time series forecasting problem that allows for better characterization of spatio-temporal trends of D2D pairs at a series of time slots, and this is the novel contribution of this paper. We show that our proposed method is competitive with the state-of-the-arts by using only forty five training samples without using any CSI.

The remainder of this paper is organized as follows. We describe the system model in Section II and formulate the sum rate maximization problem in Section III. Section IV proposes a Riemannian-geometric recurrent model for link scheduling. The performance of the proposed method is provided in Section V. We conclude the paper in Section VI.

II. System Model

In this section, we first give a brief overview of Riemannian geometry and Geometric statistical recurrent unit. Next, we present the system model of dynamic networks on the Riemannian manifold.

A. Preliminaries

A differential manifold $\mathcal{M}$ is a topological space [33] that is locally Euclidean around each point. The tangent space $T_p\mathcal{M}$ at any point p on the differential manifold $\mathcal{M}$ is a vector space of all possible tangent vectors passing through the point p. The Riemannian manifold $(\mathcal{M}, \mathcal{L})$ is a real differentiable manifold $\mathcal{M}$ in which each tangent space is equipped with an inner product $\mathcal{L}$, a Riemannian metric such as Stein metric, which varies smoothly from point to point and is studied by Riemannian geometry [33]–[35]. The $n \times n$ SPD matrices Sym_n^{++} lie on the interior of convex cones which are special class of Riemannian manifold [34].

Riemannian statistical recurrent unit employs moving averages of summary statistics to capture sequential information in time series data with long-term dependencies over Riemannian manifolds. Let, $\boldsymbol{S}_{\mathscr{D}}^1$, $\boldsymbol{S}_{\mathscr{D}}^2$,. . .,$\boldsymbol{S}_{\mathscr{D}}^T$ be an input sequence of points on Sym_n^{++}. Then the update rule for the Geometric recurrent unit are as [30]:

$$\mathcal{Y}^t = \mathrm{FM}\Big(\Big\{\left(\mu^{t-1}\right)^{(\alpha)}\Big\}, \Big\{\left(w^t\right)^{(y,\alpha)}\Big\}\Big), \tag{1}$$

$$\mathcal{R}^t = \mathrm{Tr}\Big(\mathcal{Y}^t, \left(g^t\right)^{(r)}\Big), \tag{2}$$

$$\gamma^t = \mathrm{FM}\Big(\Big\{\mathcal{R}^t, \boldsymbol{S}_{\mathscr{D}}^t\Big\}, \left(w^t\right)^{(\gamma)}\Big), \tag{3}$$

$$\Phi^t = \mathrm{Tr}\Big(\gamma^t, \left(g^t\right)^{(\varphi)}\Big), \tag{4}$$

$$\left(\mu^t\right)^{(\alpha)} = \mathrm{FM}\Big(\Big\{\left(\mu^{t-1}\right)^{(\alpha)}, \Phi^t\Big\}, \alpha\Big), \forall \alpha \in J, \tag{5}$$

$$\mathcal{S}^t = \mathrm{FM}\Big(\Big\{\left(\mu^t\right)^{(\alpha)}\Big\}, \Big\{\left(w^t\right)^{(s,\alpha)}\Big\}\Big), \tag{6}$$

$$O^t = \mathrm{Chol}\Big(\mathrm{ReLU}\Big(\mathrm{Chol}\Big(\mathrm{Tr}\Big(\mathcal{S}^t, \left(g^t\right)^{(y)}\Big)\Big)\Big)\Big), \tag{7}$$

where, $\mathrm{FM}(\cdot)$ is the recursive weighted Fréchet mean [30], [36] of $\{\boldsymbol{S}_{\mathscr{D}}^t\}_{t=1}^T$ of wights $\{w^t\}_{t=1}^T$, $\mathrm{Tr}(\cdot)$ is the group operation (i.e, *translation*) [30], $\mathrm{Chol}(\cdot)$ is the Cholesky factorization [30], w^t are the trainable weights matrix. J is the set of different time scales. In the recurrent unit above, the weighted combinations of the summary statistics of previous time (i.e., $\left(\mu^{t-1}\right)^{(\alpha)}$) are computed using different scales $\alpha \in J = \{\alpha_1, \ldots, \alpha_m\}$, where $\alpha_i = [0, 1)$, with a *translation* in (1) and (2). Next, in (3) and (4), the weighted combination of past statistics $\mathcal{R}^t$ and the current point $\boldsymbol{S}_{\mathscr{D}}^t$ are computed with a *translation* to generate present recurrent statistics Φ^t.

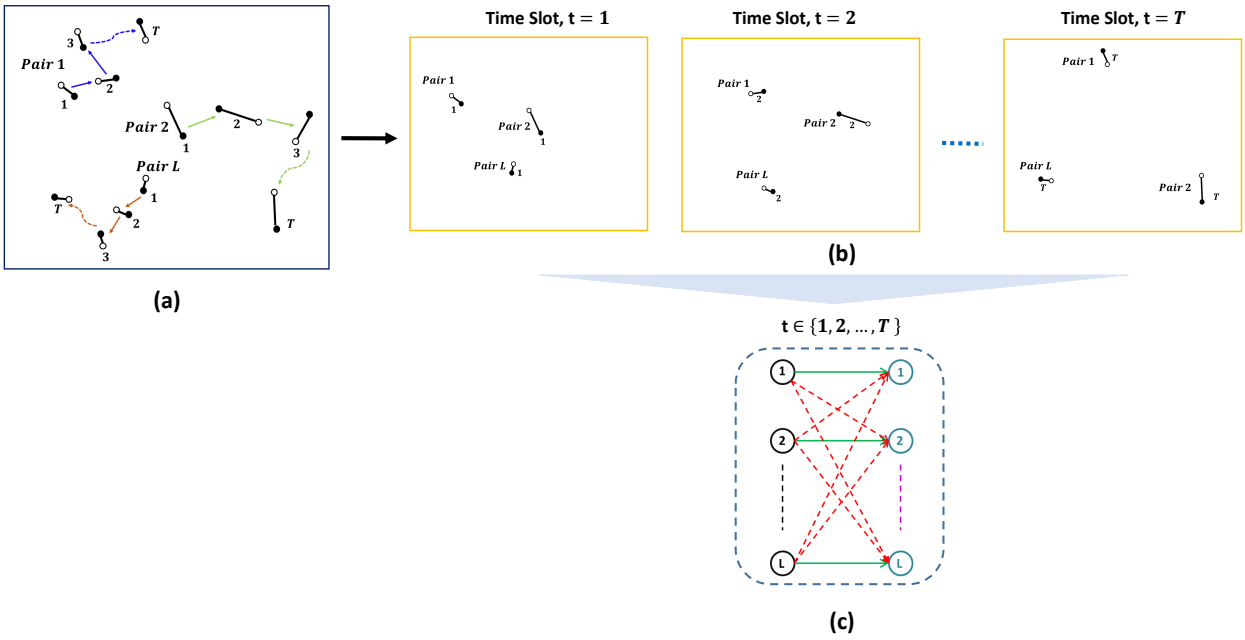

Fig. 2: Sequence of L-user interference channel model at $t = \{1, \ldots, T\}$ time slots.

Then the new summary statistics are updated based on the combined recurrent statistics at the current time (i.e., Φ^t) in (5). Finally, in (6) and (7), weighted combination of the current FMs (average), i.e., $\mathcal{S}^t$ is computed and is used to create an outputs O^t at time t. The temporal behavior of the Riemannian recurrent model is shown in Fig. 1

B. Spatio-Temporal Modeling over Riemannian Manifolds

Consider a dynamic D2D network where the temporal movements of L mobile D2D pairs in a finite time frame $\mathcal{N} \in \mathbb{R}$ is illustrated in Fig. 2a. The time frame is discretized to $T = \frac{N}{z}$ time slots indexed by $t = 1, \ldots, T$, where N is the length of time frame and z is the length of each time slot. Without loss of generality, we assume that z is small enough that the speed of D2D pairs at random direction remains unchanged within any time slot. The temporal movements of L D2D pairs are characterized as a sequence of wireless network layouts which represent snaps of movements in dynamic networks (as shown in Fig. 2b) at each time slot. Fig. 2c illustrates a basic L-user interference channel model at any time slot t. We consider $q \in L$-th D2D pair as a transceiver pair where the transmitter and the receiver is depicted through black and blue circle respectively. We represent dedicated communication links between the transmitter and receiver of any D2D pairs with green solid lines. Whereas, the interference links to and from the neighbor D2D pairs are represented with red dashed lines. With full frequency reuse, the communication between any D2D pair $\mathscr{D}_q^t$ at any time slot t creates interference to the neighbor D2D pairs receivers $\mathscr{D}_i^t$, where $i \neq q$, $i \in L$.

In this work, we incorporate M-nearest neighbor based finite directed graph model in which interfering links from the M nearest nodes are considered, where $M < L$. The M-nearest neighbor-based local graph modeling is reasonable since the interference caused by a transmitter on the considered D2D pair is negligible if the transmitter is too far from the targeted D2D pair [5], [20]. Dynamic scene can be modeled as a sequence of finite weighted and directed graphs $G^t(V, E)$, $t = 1, \ldots, T$. Each of these graphs is composed of a set of $n = 2L$ nodes, V, and a set of m edges E. The edges comprise the communication links between the D2D pairs and the interfering links to all M-nearest neighbors at any time slot t. The incidence matrix $\boldsymbol{A}^t \in \mathbb{R}^{n \times m}$ of graph G^t at time slot t is the matrix with k-th column given by edge vector $\boldsymbol{a}_k^t$. The edge vector $\boldsymbol{a}_k^t \in \mathbb{R}^n$ is defined as $a_{k_i}^t = 1$, $a_{k_j}^t = -1$ and rest of the entries are zero for the edges which connect nodes i to j, where $i, j \in M$ and $i \neq j$. The weight matrix $\boldsymbol{W}^t \in \mathbb{R}^{m \times m}$ is defined as a diagonal matrix where each diagonal entry represents the weight of k-th edge. The weight of each edge is defined by the Euclidean distance between its two nodes and quantified finite number of r bits through uniform quantization [37] to shrink the dimension from infinite to 2^r.

Finally, the Laplacian matrix $\boldsymbol{L}^t \in \mathbb{R}^{n \times n}$ at time slot t can be computed as $\boldsymbol{L}^t = \boldsymbol{A}^t \boldsymbol{W}^t (\boldsymbol{A}^t)^{\boldsymbol{T}}$, where $\boldsymbol{T}$ denotes the matrix transposition. The Laplacian matrices are positive semi-definite. Accordingly, a simple regularization step by adding a scaled identity matrix produces a regularized SPD Laplacian matrix [38] at time slot t as

$$\boldsymbol{S}_{\mathscr{D}}^{\boldsymbol{t}} = \boldsymbol{A}^t \boldsymbol{W}^t (\boldsymbol{A}^t)^{\boldsymbol{T}} + \gamma \boldsymbol{I}, \tag{8}$$

where, $\gamma > 0$ is a regularization parameter and $\boldsymbol{I}$ is the $n \times n$ identity matrix. The Riemannian metrics such as Stein metric [31], [32] can be used to measure geodesic distance between SPD matrices that correspond to the temporal-dependency among D2D pairs at successive time slots. The distance between $\boldsymbol{S}_{\mathscr{D}}^{\boldsymbol{t}}$ and $\boldsymbol{S}_{\mathscr{D}}^{\boldsymbol{t'}}$ through Stein metric can be computed

as

$$\mathcal{L}(\boldsymbol{S}^t_{\mathscr{D}},\boldsymbol{S}^{t'}_{\mathscr{D}})=\sqrt{\log\det\left(\frac{\boldsymbol{S}^t_{\mathscr{D}}+\boldsymbol{S}^{t'}_{\mathscr{D}}}{2}\right)-\log\det\left(\boldsymbol{S}^t_{\mathscr{D}}\boldsymbol{S}^{t'}_{\mathscr{D}}\right)}, \quad (9)$$

where, $t, t' \in T$, and $t \neq t'$.

We provide spatio-temporal correlation information at time slot t for a given sequence of SPD points $\boldsymbol{S}^1_{\mathscr{D}}, \ldots, \boldsymbol{S}^T_{\mathscr{D}} \in \text{Sym}^{++}_n$ through recurrent statistics $\phi^t_{\mathscr{D}}$ that depend on the values of SPD points at previous time slots. That is, we compute the recurrent statistics at t-th time slot not only as a function of $\boldsymbol{S}^t_{\mathscr{D}}$, but also as a function of the previous statistics $\boldsymbol{S}^{t-1}_{\mathscr{D}}$, $\phi^{t-1}_{\mathscr{D}}$ (which itself depends on $\phi^{t-2}_{\mathscr{D}}$ and so on). Then the long-term dependencies in the input sequence of SPD points are captured through the exponential moving average of recurrent statistics $\phi^t_{\mathscr{D}}$ at different time scales (i.e., summary statistics). The summary statistics at time slot t over Riemannian manifold can be computed by weighted Fréchet mean (wFM) [30], [36] as follows:

$$\left(\mu^t_{\mathscr{D}}\right)^{(\alpha)} = \underset{\mu}{\text{argmin}} \sum_{t=1}^{T} \alpha \mathcal{L}^2\left(\left(\mu^{t-1}_{\mathscr{D}}\right)^{(\alpha)}, \phi^t_{\mathscr{D}}\right), \quad (10)$$

where, $\alpha \in J = \{\alpha_1, \ldots, \alpha_m\}$, and $\alpha_i \in [0, 1)$. The link scheduling decisions $\boldsymbol{d^t}$ of D2D pairs at time slot t are forecasted using the summary statistics $\left(\mu^t_{\mathscr{D}}\right)^{(\alpha)}$.

III. Problem Formulation

In this section, we formulate the sum rate maximization problem in L-user interference channel, as depicted in Fig. 2c. We introduce $\boldsymbol{d}^t = [d^t_1, \ldots, d^t_L]^{\boldsymbol{T}}$ as the indicator vector of D2D pairs states, with $d^t_q = 1$ if $\mathscr{D}^t_q$ is activated, and $d^t_q = 0$ if not at any time slot $t = 1, \ldots, T$. With full frequency reuse over bandwidth B, our objective is to find the optimal combinations of the indicator vector that maximize the summation of instantaneous information-theoretic rates over T time slots as given by

$$\max_{\boldsymbol{d}^t} \frac{1}{T} \sum_{t=1}^{T} \sum_{q=1}^{L} B \log_2\left(1 + \frac{p d^t_q |h^t_{qq}|^2 (\rho^t_{qq})^{-\alpha}}{\sum_{i \neq q} p d^t_i |h^t_{iq}|^2 (\rho^t_{iq})^{-\alpha} + \sigma^2}\right), \quad (11)$$

$$\text{s.t. } d^t_q \in \{0, 1\}, \forall q = 1, 2, \ldots, L, t = 1, 2, \ldots, T,$$

where p is the transmission power which is same for all links, h^t_{iq} is the fast-fading channel gain which is modeled as a circularly-symmetric complex-Gaussian random variable, and ρ^t_{iq} is Euclidean distance between the i-th transmitter and q-th receiver respectively at time slot t. Moreover, α is the path loss exponent and σ^2 is the noise variance.

IV. Geometric Recurrent Model for Link Scheduling

In this section, we describe the proposed link scheduling scheme in dynamic networks. We start with modeling the temporal movement of each D2D pair on the manifold in any finite time frame of T time slots. Then we explain how we use the local graph modeling in R-RNN.

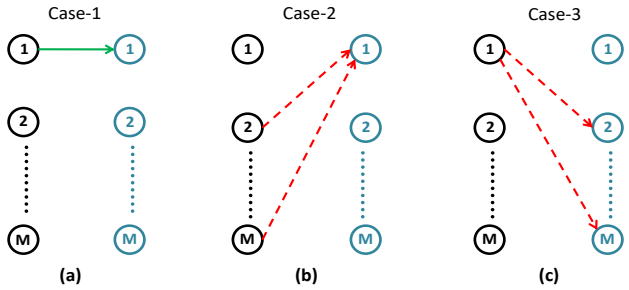

Fig. 3: Example of local graph modeling of $\mathscr{D}^t_{q=1}$ at time slot t: a) modeling communication link, b) modeling interference links from its neighbor D2D pairs, and c) modeling interference links to its neighbor D2D pairs.

A. Problem Modeling on Riemannian manifold

For link scheduling, we need to know the impact of the interference on the wireless networks if any D2D pair is scheduled at any time slot t. So, we consider three special cases of graph G^t_q at time slot $t \in T$ as shown in Fig. 3. The case-1 of graph G^t_q is the one that models the communication link between the targeted D2D pair $\mathcal{D}^t_q$ at time slot t, denoted as $G^t_{\mathscr{C}_q}(V, E_{\mathscr{C}_q})$ (Fig. 3a). Here, we only consider the communication link of the targeted D2D pair which provides indication on the signal-to-noise level for a given D2D pair. The interference, on the other hand, is considered in the other two cases as follows. The case-2 of graph G^t_q is the one that models the impact of the interference on the targeted D2D pair if the M-nearest neighbor of the targeted D2D pair is scheduled at time slot t, denoted as $G^t_{\mathscr{I}_q}(V, E_{\mathscr{I}_q})$ (Fig. 3b). We consider the interference links from the M-nearest transmitters to the receiver of the targeted D2D pair in case-2. Finally, the case-3 of graph G^t_q is the one that models the impacts of the interference on the receivers of M-nearest neighbors if the communication link of $\mathscr{D}^t_q$ is scheduled at time slot t, denoted as $G^t_{\mathscr{N}_q}(V, E_{\mathscr{N}_q})$ (Fig. 3c). In case-3, we consider the interference links from the transmitter of the targeted D2D pair to the receivers of its M-nearest neighbors. Then we formulate the sequence of SPD matrices as follows:

We use the graph models corresponding to three special cases to compute three Laplacian matrices $\boldsymbol{L}^t_{\mathscr{C}_q}$, $\boldsymbol{L}^t_{\mathscr{I}_q}$ and $\boldsymbol{L}^t_{\mathscr{N}_q}$ at each time slot t which are positive semi-definite. Then We formulate three SPD matrices $\boldsymbol{S}^t_{\mathscr{C}_q}$, $\boldsymbol{S}^t_{\mathscr{I}_q}$ and $\boldsymbol{S}^t_{\mathscr{N}_q}$ with the simple regularization step as in (8). Addressing that the sum of SPD matrices is also a positive definite matrix [39], we add these three SPD matrices and form a single SPD matrix to get complete interference information for the targeted D2D pair at any time slot t. We use Algorithm 1 to represent the D2D pairs local graphs as sequence of points $\{\boldsymbol{S}^t_{\mathscr{D}_q}\}^T_{t=1}, \forall q \in L$ on manifold for a series of time slots $t = 1, \ldots, T$.

B. Riemannian-Geometric RNN

Now, for a given sequence of local graph modeling, our objective is to use the summary statistics in R-RNN to classify the D2D pair's state (i.e., active or inactive) at N successive time slots ahead. The proposed link scheduling scheme is

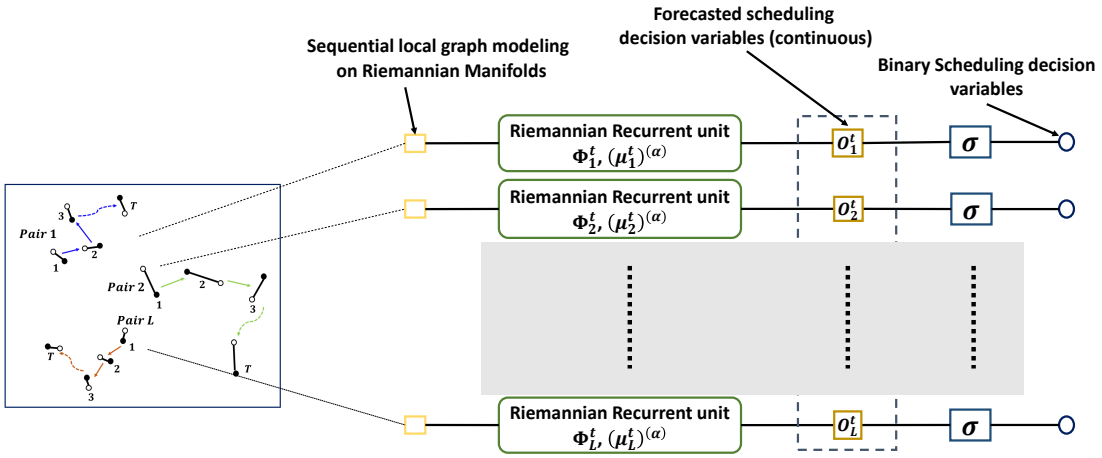

Fig. 4: Overall proposed R-RNN model as applied to link scheduling problem with a single Riemannian recurrent unit per D2D pair.

Algorithm 1 Sequential local graph modeling on manifold

1: **Input:** $\boldsymbol{L}^t_{\mathscr{C}_q}$, $\boldsymbol{L}^t_{\mathscr{I}_q}$, $\boldsymbol{L}^t_{\mathscr{N}_q}$, $\forall q \in L$, $t \in T$
2: **Initialization:** $\boldsymbol{S}^t_{\mathscr{D}_q} = 0$, $\forall q \in L$, $t \in T$
3: **for** $t = 1, \ldots, T$ **do**
4: **for** $q = 1, \ldots, L$ **do**
5: · Compute $\boldsymbol{S}^t_{\mathscr{C}_q}$, $\boldsymbol{S}^t_{\mathscr{I}_q}$, $\boldsymbol{S}^t_{\mathscr{N}_q}$ using (8);
6: · $\boldsymbol{S}^t_{\mathscr{D}_q} = \boldsymbol{S}^t_{\mathscr{C}_q} + \boldsymbol{S}^t_{\mathscr{I}_q} + \boldsymbol{S}^t_{\mathscr{N}_q}$;
7: **end for**
8: **end for**
9: **return** $\{\boldsymbol{S}^t_{\mathscr{D}_q}\}_{q=1}^{L}, \forall t = \{1, \ldots, T\}$

summarized in Algorithm 2 and illustrated in Fig. 4 and Fig. 5.

The *inputs* to the model are the training data set $\boldsymbol{D}_{t_r} = \{(\boldsymbol{S}^{t_r}_{\mathscr{D}_q}, \hat{d}^{t_r}_q)\}_{t_r=1}^{\tau+N-1}$, test data points $\boldsymbol{D}_{t_s} = \{\boldsymbol{S}^{t_s}_{\mathscr{D}_q}\}_{t_s=N+1}^{T=\tau+2N-1}$, and the variable *phase*, where $\hat{d}^{t_r}_q$ are the true targets or label to train the model (lines 1-4). Both data sets are segmented into N Batches with size τ, i.e., windows $\{\boldsymbol{D}_1, \ldots, \boldsymbol{D}_N\}$, as shown in Fig. 5. The *outputs* are the forecasted link scheduling decisions $\{d^{t_s}_q\}_{t_s=N+1}^{T=\tau+2N-1} \in \{0, 1\}$ (lines 5-6).

At the *initial* phase, the finite time frame of T time slots is divided into two consecutive phases (Fig. 5a), such as *training* phase $\{1, \ldots, \tau+N-1\}$, and *test* phase $\{\tau+N, \ldots, \tau+2N-1\}$ (line 7). The summary statistics $\left(\mu^0_q\right)^{(\alpha)}$ is initialized to be a $n \times n$ matrix with small values.

At the *training* phase, the training data set is fit to the model (lines 11-12) as shown in Fig. 5b. The algorithm feeds the samples to the model from each window to update the Riemannian recurrent unit (as described in Section II-A) by computing the summary statistics and forecasts a continuous link scheduling decision variable of q-th D2D pair at time slot t_r (lines 16-24). For instance, the first forecast would be made at time slot $t_r = \tau$ using summary statistics $\left(\mu^\tau_q\right)^{(\alpha)}$ (5), kept at various scales $\alpha \in J = \{\alpha_1, \ldots, \alpha_m\}$, where $\alpha_i = [0, 1)$ which is computed from the input points $\{\boldsymbol{S}^1_{\mathscr{D}_q}, \ldots, \boldsymbol{S}^\tau_{\mathscr{D}_q}\}$ in window $\boldsymbol{D}_1$. The summary statistics $\left(\mu^\tau_q\right)^{(\alpha)}$, are of recurrent statistics Φ^τ_q (3) and (4) that is dependent not only on the current input point $\boldsymbol{S}^\tau_{\mathscr{D}_q}$ but also on the features of averages

Algorithm 2 Proposed R-RNN scheme

1: **Inputs:** Training data set $\boldsymbol{D}_{t_r} = \{(\boldsymbol{S}^{t_r}_{\mathscr{D}_q}, \hat{d}^{t_r}_q)\}_{t_r=1}^{\tau+N-1}$, $q \in L$,
2: Test data points $\boldsymbol{D}_{t_s} = \{\boldsymbol{S}^{t_s}_{\mathscr{D}_q}\}_{t_s=N+1}^{T=\tau+2N-1}$, $q \in L$, and
3: *phase*; where each data set is divided into N batches
4: $\{\boldsymbol{D}_1, \ldots, \boldsymbol{D}_N\}$
5: **Outputs:** Forecasted scheduling of N successive time slots for
6: q-th D2D pair;
7: **Initialization:** Divide the time frame into *train* and *test* phases;
8: $\left(\mu^0_q\right)^{(\alpha)}$ is initialized to be a $n \times n$ matrix with
9: small values;
10: **Process:**
11: **if** *phase*="train" **then**
12: Fit training data set $\boldsymbol{D}_{t_r}$;
13: **else**
14: Select test data set $\boldsymbol{D}_{t_s}$;
15: **end if**
16: Set $b \leftarrow 1$; ▷ selecting first input window
17: **while** $b \leq N$ **do** ▷ N window slides
18: $\boldsymbol{D}_b = \{(\boldsymbol{S}^l_{\mathscr{D}_q}, y^l_q)\}_{l=b}^{\tau+b-1}$;
19: Set $l \leftarrow 1$;
20: **while** $l \leq \tau$ **do**
21: · Slice the window at time slot l: $\boldsymbol{S}^l_{\mathscr{D}_q} \leftarrow \boldsymbol{D}_{b,l}$;
22: · Update the Riemannian recurrent unit;
23: · $l \leftarrow l + 1$;
24: **end while**
25: Binarize forecasted scheduling decision variable using (12);
26: **if** *phase*="train" **then**
27: Calculate the error and update the weights;
28: **end if**
29: $b \leftarrow b + 1$;
30: **end while**

$\mathcal{R}^\tau_q$ (1) and (2). Then the weighted combination of the current summary statistics, i.e., $\{\left(\mu^\tau_q\right)^{(\alpha_1)}, \ldots, \left(\mu^\tau_q\right)^{(\alpha_m)}\}$ is computed as in (6) and is used to create the output O^τ_q (7) at time slot $t_r = \tau$. Likewise, by shifting the window by 1 time slot, the R-RNN forecasts the scheduling decision at time slot $t_r = \tau+1$. The forecast is made based on summary statistics $\left(\mu^{\tau+1}_q\right)^{(\alpha)}$ using the points in window D_2 and so on. The algorithm discretizes the outputs (line 25) to obtain binary scheduling decision variable $d^{t_r}_q \in \{0, 1\}$ by

$$d^{t_r}_q = f_{\text{round}}\left(\sigma(O^{t_r}_q)\right), \tag{12}$$

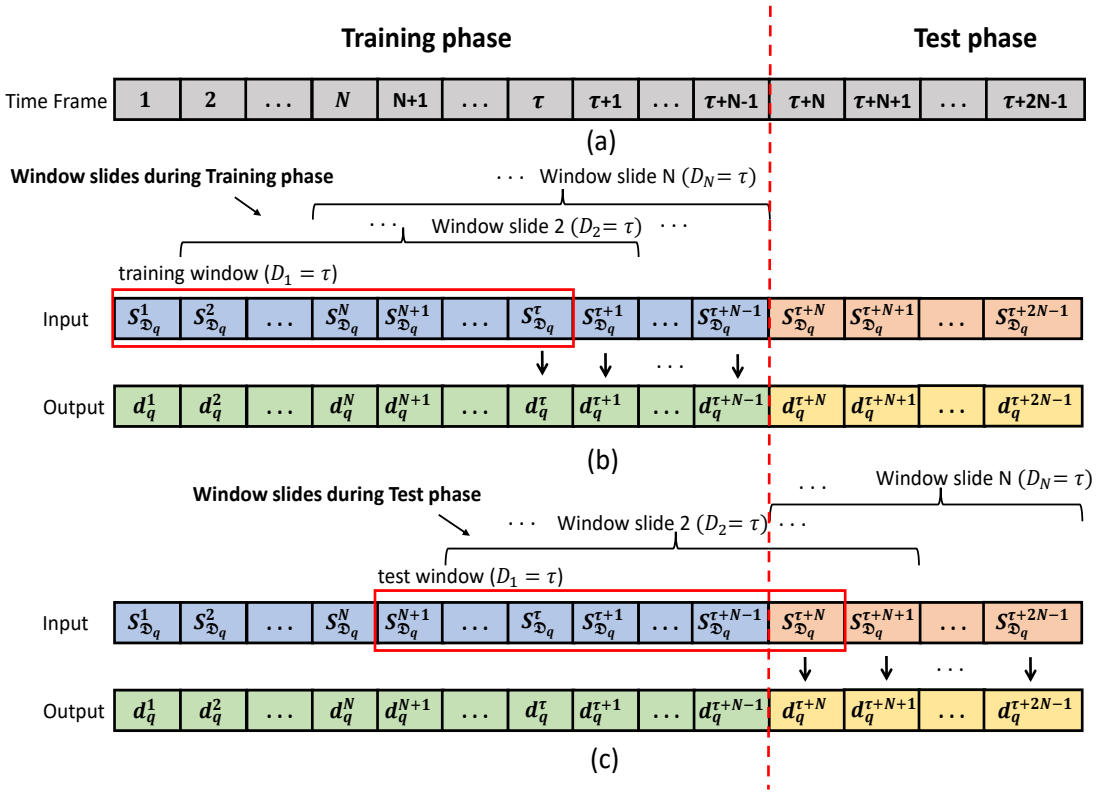

Fig. 5: Detailed illustration of train and test data processing: a) a finite time frame of $T = \tau + 2N - 1$ time slots divided into *train* and *test* phases, b) the N input windows of size τ during *training* phase consist with the sequence of points $\boldsymbol{S}^1_{\mathscr{D}_q}, \ldots, \boldsymbol{S}^{\tau+N-1}_{\mathscr{D}_q}$, and c) the N input windows of size = τ during *test* consist with the sequence of points $\boldsymbol{S}^{N+1}_{\mathscr{D}_q}, \ldots, \boldsymbol{S}^{\tau+2N-1}_{\mathscr{D}_q}$ and the forecasted link scheduling decisions $\{d_q^{t_s}\}_{t_s=\tau+N}^{\tau+2N-1} \in \{0,1\}$ of N successive time slots ahead.

where σ is a sigmoid activation function and $f_{\text{round}}()$ coverts the floating point number to decimal values, i.e., 0 or 1. Then it compares the forecasted binary link scheduling decision $d_q^{t_r}$ with the target $\hat{d}_q^{t_r}$, calculates the error and updates the trainable parameters (i.e., $\left(w^t\right)^{(y,\alpha)}$, $\left(w^t\right)^{(\gamma)}$, and $\left(w^t\right)^{(s,\alpha)}$ in (1-7)) of the Riemannian recurrent unit (lines 26-28). We use *softmax_cross_entropy_with logits* loss function [40] and ADADELTA optimizer [41] where the initial learning rate is set to 0.9, decay rate is set to 0.99 and decay step is set to 1000. The algorithm repeats the above steps until forecasting N successive decisions for N *training* batches.

At the *test* phase, the test data points $\boldsymbol{D}_{t_s} = \{\boldsymbol{S}^{t_s}_{\mathscr{D}_q}\}_{t_s=N+1}^{T=\tau+2N-1}$ are used (lines 13-14) to forecast the scheduling decisions of N successive *test* time slots by only using the summary statistics of N *test* batches (Fig. 5c). At this phase, no weights and biases of the Riemannian recurrent unit are updated. The rest of the steps are exactly the same as those done in the training phase (lines 16-30).

We emphasize that the geometric recurrent model is designed on a per link basis where a single geometric recurrent unit is used for each D2D pair. Hence, the overall model is scalable with respect to the number of D2D pairs in the dynamic networks.

V. PERFORMANCE EVALUATION

This section presents the simulation results of our proposed R-RNN method and compares it against other benchmark solutions in terms of sum rate performance.

A. Simulation Setup

In this experiment, we set up the moving pattern of D2D pairs in a finite time frame of T time slots as follows: First, we consider a $500m \times 500m$ two-dimensional square are with L D2D pairs. The transmitters of L D2D pair are deployed by uniform distribution in the area and the receivers are distributed in a disk centered by its corresponding transmitter with uniform pairwise distance between $2m$ to $65m$. This wireless network layout represents the dynamic scene at the very first time slot $t = 1$. Then, for each succeeding time slot, the transmitter of each D2D pair is first randomly displaced by a distance between $2m$ to $60m$ in any random direction from its previous position, while the receivers are deployed within the disk (with a uniformly distributed radius between $2m$ to $65m$) around its corresponding transmitter pair similarly as before.

In our simulation, we use a *single* Riemannian recurrent unit per D2D pair for link scheduling forecasting in proposed R-RNN method. Besides, we consider a distance-based path-loss according to the outdoor model ITU-1411. We set the number of quantization bits, $q = 3$ to quantify the distances

TABLE I: NETWORK SIMULATION PARAMETERS

Parameter	Value
D2D distance $(d_{\min}, d_{\max})$	2m to 65m
Carrier Frequency	2.4 GHz
Bandwidth, B	5MHz
Transmit power p of the activated link	40 dBm
Antenna height (both Transmitter and Receiver)	1.5m
Antenna gain (both Transmitter and Receiver)	2.5 dB
Noise spectral density	-169 dBm/Hz

between the nodes in the Euclidean space. Moreover, the link scheduling decisions of FPLinQ algorithm [9] are used as targets to train the proposed model. Rest of simulation parameters are kept same as in [9], [4], [5] and [27] and included in Table I to have a fair comparison. The design paramenters of Riemannian recurrent neural networks are summarized in Table II.

TABLE II: RIEMANNIAN RECURRENT NEURAL NETWORK PARAMETERS

Parameter	Value
Number of Riemannian Recurrent Unit	1 unit per D2D pair
Input Matrix size	Defined by the dimension of input SPDs
Batch Size	38
Number of Batches	8
Set of scales used for Computing Summary Statistics	$J = \{0.01, 0.25, 0.5, 0.9, 0.99\}$

B. Simulation Results

Since the proposed R-RNN method forecasts the scheduling decisions by only using the summary statistics, errors that occurred in previous time slots will be propagated to the prediction of spatio-temporal correlations among D2D pairs at the succeeding time slots ahead. Hence, it is difficult to accurately forecast the scheduling decisions for an unlimited number of time slots ahead. So, we first test the sum rate performance of proposed R-RNN method for a different number of training samples and its impact on the number of forecasted link scheduling decisions at successive time slots ahead for 50 D2D pairs. In Table III, we compare the achievable sum rate performance and the number of forecasted scheduling time slots under different number of training samples. As shown in the table, 45 training samples are enough to train the proposed model for forecasting scheduling up to $N = 8$ successive time slots ahead, due to the fact that there is no notable performance enhancement with the increase in the number of training samples and our goal is to reduce the requirement of the number of training samples.

TABLE III: IMPACT OF THE NUMBER OF TRAINING SAMPLES ON THE NUMBER OF FORECASTED LINK SCHEDULING AT SUCCESSIVE TIME SLOTS AHEAD

Number of Training Network Layouts	30	45	60	90	200	500
Scheduling for Number of time slots ahead	7	8	8	10	11	12
Average Sum Rate (%)	89.14%	95.79%	95.46%	95.12%	94.87%	95.23%

Next, we experiment with the M-nearest neighbor D2D pairs for local graph modeling to test its impact on the sum rate performance. We still consider 50 pairs in a $500m \times 500m$ region for each scenario and the results are summarized in Table IV. Here, the *upper bound* on the sum rate performance is given by all D2D neighbor pairs local graph modeling. Table IV shows that the sum rate performance varies between 95% to 97% of FPLinQ algorithm [9] for various numbers of nearest neighbors. This implies that 10-nearest neighbor pairs are enough for local graph modeling and there is no performance loss for not using a higher number of neighbors.

TABLE IV: AVERAGE SUM RATE FOR M-NEAREST NEIGHBOR D2D PAIRS. THE RESULTS ARE NORMALIZED BY THE SUM RATE OF FPLinQ [9]

M-nearest D2D pairs	10	20	36	40	49 (Full connectivity)
Average Sum Rate (%)	95.79%	96.02%	96.18%	96.37%	97.06%

The average sum rate performances of various wireless link scheduling methods for 50 D2D pairs in a $500m \times 500m$ region is summarized in Table V. From the table, the spatial learning [4] and graph embedding [5] need 800,000 and 500 *training wireless network layouts*, respectively, to approach the sum rate performance as FPLinQ [9] while requiring no CSI. In contrast, the G-SVM method in [20] reduces the number of training wireless network layouts to 90 to achieve a comparable performance as [4] and [5] and without using any CSI. However, our proposed R-RNN method only needs 45 training wireless network layouts to achieve a good sum rate without using CSI. Hence, the requirement of the number of training samples is reduced by 91% and 50% compared to the graph embedding [5] and G-SVM [20] method respectively.

TABLE V: LINK SCHEDULING METHODS WITH AVERAGE SUM RATES FOR 50 D2D PAIRS. THE RESULTS ARE NORMALIZED BY THE SUM RATE OF CSI- BASED FPLinQ [9]

Scheduling Methods	CSI	Achievable Sum rate (on Avg.)	Required Training Samples	Scheme
FPLinQ [9]	Yes	100%	/	Fractional programming optimization
Proposed R-RNN	No	95.79%	45	Graph Modeling & RNN
Fully Connected R-RNN (upper bound)	No	97.06%	45	Graph Modeling & RNN
G-SVM [20]	No	94.84%	90	Graph Modeling, Kernel & SVM
Spatial learning [4]	No	96.20%	80,000	Kernel method & DNN
Graph embedding [5]	No	95.29%	500	Graph embedding & DNN
LEM-based sequential [27]	No	86.15%	/	Sequential link selection

1) Scalability to Different Pairwise Distances and Link Densities: We first test the performance of the proposed method on scenarios with different pairwise distances for $L = 50$ D2D pairs in the network area of $500m \times 500m$ and the results are summarized in Table VI. From the Table, the performance of the proposed method deteriorates with the decrease of the D2D pairwise distribution interval. However, the proposed method still achieves promising sum rate performance against other benchmark algorithms. For example,

the proposed method achieves comparable average sum rates as G-SVM [20], graph embedding [5] and supervised spatial learning [4] for scenarios with pairwise distance $(2m \sim 65m)$ and $(10m \sim 50m)$, and outperforms the G-SVM [20] and spatial learning [4] by 3.74% and 9.32% respectively for the pairwise distance $(30m \sim 70m)$. The proposed method also outperforms the G-SVM and the spatial learning method for the scenario of pairwise distance (*all* $30m$).

On the other hand, the proposed R-RNN method approaches the Greedy heuristic which requires CSI. Moreover, the other simple heuristic methods (i.e., strongest link, random, and all active) have poor performance with different pairwise distances, compared to the proposed approach. This emphasizes that the machine learning based method is necessary for link scheduling.

TABLE VI: AVERAGE SUM RATE WITH DIFFERENT PAIRWISE DISTANCES. THE RESULTS ARE NORMALIZED BY THE SUM RATE OF FPLinQ [9]

Pairwise Distance $d_{\min} \sim d_{\max}(m)$	**CSI**	$2 \sim 65$	$10 \sim 50$	$30 \sim 70$	all 30
Proposed R-RNN	No	95.79%	93.55%	92.52%	89.34%
G-SVM [20]	No	94.84%	90.13%	88.78%	81.20%
Spatial learning [4]	No	96.20%	90.30%	83.20%	82.00%
Graph embedding [5]	No	95.29%	93.08%	91.76%	87.18%
Greedy	Yes	97.08%	94.00%	84.76%	84.56%
Strongest Link	No	82.03%	75.41%	59.66%	N/A
Random	No	47.47%	49.63%	35.30%	50.63%
All Active	No	54.18%	48.22%	26.74%	43.40%
Fully Connected R-RNN (upper bound)	No	97.06%	95.21%	93.76%	90.14%

To understand the behavior of the link scheduling schemes, we compare their link activation (i.e., scheduling ratios) for different link densities (i.e., $Area/L$) at 5-th sample time slot in Fig. 6. As shown in the figure, our proposed method closely follows the link activation pattern of CSI based FPLinQ [9], whereas the other schemes fail to follow the state-of-the-art behavior. This illustrates that the proposed R-RNN method is capable of learning the state-of-the-art optimization strategy.

Next, we test the scalability of our proposed method to topologies with different link densities, i.e., the ratio between the layout area and the number of D2D pairs inside that area, with respect to $500m \times 500m$ region. As shown in Table VII, the proposed R-RNN method achieves similar sum rate performance as [5] for all scenarios except 500 D2D pairs and achieves slightly higher performance on average than G-SVM [20], while having a few fold reduction in training samples. However, the performance decreases by around 7% with the scenario of 500 D2D pairs compared to the 50 D2D pairs scenario.

2) Generalizability to Different Fading Scenes: We test the generalizability of the proposed method for various fast fading scenes. Generalizability testing is performed in a different way

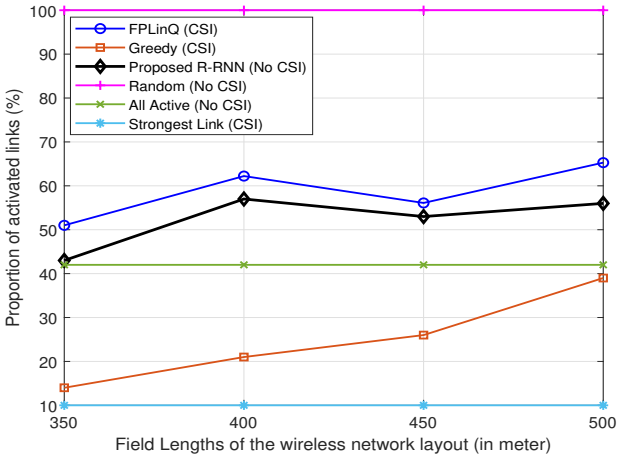

Fig. 6: Proportion of activated links for various link scheduling scheme

TABLE VII: SCALABILITY TO DIFFERENT LINK DENSITIES. THE RESULTS ARE NORMALIZED BY THE SUM RATE OF FPLinQ [9]

D2D Pairs	10	30	50	100	500
Proposed R-RNN	/	96.35%	95.79%	95.16%	88.74%
G-SVM [20]	/	94.39%	94.84%	94.52%	86.38%
Graph embedding [5]	97.51%	96.14%	95.29%	93.53%	91.13%
Fully Connected R-RNN (upper bound)	97.32%	97.58%	97.06%	96.56%	89.52%

than scalability testing. Unlike the scalability testing where the training is performed every time for different scenarios, we train our proposed model only once for any particular scenario and we use the already trained model for different topological scenarios for generalizability testing.

TABLE VIII: AVERAGE SUM RATE WITH DIFFERENT FAST FADING SCENARIOS. THE RESULTS ARE NORMALIZED BY THE SUM RATE OF FPLinQ [9]

Various Fast Fading Realization	I	II	III	IV	V	VI	Avg.
Proposed R-RNN	87.74%	86.13%	84.77%	90.22%	87.34%	86.52%	87.12%
Fully Connected R-RNN (upper bound)	88.45%	89.53%	86.23%	91.54%	88.76%	87.65%	88.69%
G-SVM [20]	83.62%	92.30%	84.34%	82.24%	83.72%	84.05%	85.05%
FPLinQ [9] without knowing fading	89.76%	88.45%	88.76%	84.39%	89.51%	87.78%	88.11%

We introduce Rayleigh fast fading conditions into the ITU-1411 model and show six realizations in Table VIII. We notice some performance deterioration while testing the proposed method with different fast fading scenes. However, the proposed R-RNN method achieves similar performance as FPlinQ [9] without knowing fading. Moreover, the sum rate performance of the proposed method and the fully connected R-RNN (i.e., upper bound) are similar.

C. Computational Complexity

In this subsection, we inspect the computational complexity for the recurrent neural network based method and make a

TABLE IX: Computational Complexity Analysis

Methods	FPLinQ [9]	Spatial Learning [4]	Graph Embedding [5] (M-nearest neighbor)	G-SVM [20] (M-nearest neighbor)	Proposed R-RNN (M-nearest neighbor)
D2D Network Representation Complexity	/	Defined by Convolutional Filters	$\mathcal{O}(L)$	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2)$
Machine Learning Complexity (scale as L)	/	$\mathcal{O}(L)$	$\mathcal{O}(L)$	$\mathcal{O}(L)$	$\mathcal{O}(L)$
Total Computational Complexity	= $\mathcal{O}(L^2)$	= $\mathcal{O}(L)$	= $\mathcal{O}(L)$	= $\mathcal{O}(L^2)$	= $\mathcal{O}(L^2)$

comparison with other state-of-the-arts in Table IX.

The proposed method is formulated in two steps: 1) sequential local graph modeling on Riemannian manifold, and 2) forecasting link scheduling for N consecutive time slots. Considering the formation of a series of SPD matrices for a fixed number of time slots T, the complexity for the manifold representation is $\mathcal{O}(L^2)$. On the other hand, with a fixed number of iteration, the computational complexity for R-RNN forecasting is:

$$\mathcal{O}(LW\tau N) \approx \mathcal{O}(L), \tag{13}$$

where W is the total number of trainable parameters in the Riemannian recurrent unit, τ is the size of the input window, and N is the number of batches (i.e., windows) that are fixed in our context. So, the total computational complexity for our proposed method is $\mathcal{O}(L^2)$. Thus, our proposed method has similar complexity with FPLinQ method [9] but does not need any CSI that is hard to obtain in practice. The complexity is also the same as G-SVM [20] method, but it requires 50% less training wireless network layouts than G-SVM.

From the analysis above, the sequential local graph modeling of D2D pairs on Riemannian manifold brings up additional computational complexity in comparison to the graph embedding method in [5] and spatial learning [4]. However, the proposed R-RNN method requires less training samples than those benchmarks to achieve comparable performance.

VI. Conclusion

In this paper, we have introduced a Riemannian-geometric recurrent model for wireless link scheduling problems in dynamic D2D networks. Our aim is to reduce the requirement of the number of training samples further than state-of-the-arts by capturing the spatio-temporal correlations in dynamic networks such as vehicular communications. To this aim, we have first modeled the temporal movements of D2D nodes in any finite time interval as a time series forecasting problem. Then we have proposed the Riemannian-geometric recurrent model based on the statistical recurrent unit on SPDs that learns the spatio-temporal correlation among D2D pairs over a series of time slots through the summary statistics. The proposed R-RNN method is able to forecast link scheduling decisions for a finite number of successive time slots ahead. We have shown that our proposed method achieves promising performances for sum rate maximization with only 45 training wireless network layouts compared to the requirement of hundreds to thousands of training samples by the state-of-the-arts.

References

[1] Z. Liu, X. Han, Y. Liu and Y. Wang, "D2D-Based Vehicular Communication With Delayed CSI Feedback," in IEEE Access, vol. 6, pp. 52857-52866, 2018, doi: 10.1109/ACCESS.2018.2870166.

[2] Z. Liu, Y. Xie, K. Y. Chan, K. Ma and X. Guan, "Chance-Constrained Optimization in D2D-Based Vehicular Communication Network," in IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 5045-5058, May 2019, doi: 10.1109/TVT.2019.2904291.

[3] O. Goussevskaia, Y.A. Oswald, and R. Wattenhofer, Complexity in geometric SINR, Proc. of the 8th ACM MOBIHOC, pp. 100–109, September 2007.

[4] W. Cui, K. Shen, and W. Yu, "Spatial Deep Learning for Wireless Scheduling," IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1248–1261, 2019.

[5] M. Lee, G. Yu and G. Y. Li, "Graph Embedding-Based Wireless Link Scheduling With Few Training Samples," in IEEE Transactions on Wireless Communications, vol. 20, no. 4, pp. 2282-2294, April 2021, doi: 10.1109/TWC.2020.3040983.

[6] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," IEEE J. Sel. Areas Commun., vol. 32, no. 6, pp. 1139–1151, Jun. 2014.

[7] X. Wu et al., "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," IEEE/ACM Trans. Netw., vol. 21, no. 4, pp. 1215–1228, Aug. 2013.

[8] T. Tony, S. Soh, M. Lazarescu and K. -W. Chin, "Link Scheduling in Rechargeable Wireless Sensor Networks with a Dual-Battery System," ICC 2021 - IEEE International Conference on Communications, 2021, pp. 1-7, doi: 10.1109/ICC42927.2021.9500710.

[9] K. Shen and W. Yu, "FPLinQ: A Cooperative Spectrum Sharing Strategy for Device-to-Device Communications," in 2017 IEEE International Symposium on Information Theory (ISIT), 2017, pp. 2323–2327.

[10] M. Klügel and W. Kellerer, "Semi-Decentralized Interference Aware Scheduling in D2D-Enabled Cellular Networks," in IEEE Access, vol. 8, pp. 132285-132301, 2020, doi: 10.1109/ACCESS.2020.3009778.

[11] J. Yu et al., "Efficient Link Scheduling in Wireless Networks Under Rayleigh-Fading and Multiuser Interference," in IEEE Transactions on Wireless Communications, vol. 19, no. 8, pp. 5621-5634, Aug. 2020, doi: 10.1109/TWC.2020.2994998.

[12] C. Ma, J. Yu, B. Huang and Y. Meng, "Shortest link scheduling in wireless networks with oblivious power control," in China Communications, vol. 18, no. 4, pp. 137-152, April 2021, doi: 10.23919/JCC.2021.04.011.

[13] K. Yu, J. Yu, X. Cheng, D. Yu and A. Dong, "Efficient Link Scheduling Solutions for the Internet of Things Under Rayleigh Fading," in IEEE/ACM Transactions on Networking, vol. 29, no. 6, pp. 2508-2521, Dec. 2021, doi: 10.1109/TNET.2021.3093306.

[14] I. Budhiraja, N. Kumar and S. Tyagi, "Deep-Reinforcement-Learning-Based Proportional Fair Scheduling Control Scheme for Underlay D2D Communication," in IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3143-3156, 1 March1, 2021, doi: 10.1109/JIOT.2020.3014926.

[15] Y. Shen, Y. Shi, J. Zhang and K. B. Letaief, "A Graph Neural Network Approach for Scalable Wireless Power Control," 2019 IEEE Globecom Workshops (GC Wkshps), 2019, pp. 1-6, doi: 10.1109/GCWkshps45667.2019.9024538.

[16] J. Joo, M. C. Park, D. S. Han and V. Pejovic, "Deep Learning-Based Channel Prediction in Realistic Vehicular Communications," in IEEE Access, vol. 7, pp. 27846-27858, 2019, doi: 10.1109/ACCESS.2019.2901710.

[17] Y. Ren, F. Liu, Z. Liu, C. Wang and Y. Ji, "Power Control in D2D-Based Vehicular Communication Networks," in IEEE Transactions on Vehicular Technology, vol. 64, no. 12, pp. 5547-5562, Dec. 2015, doi: 10.1109/TVT.2015.2487365.

[18] C. Tatino, N. Pappas, I. Malanchini, L. Ewe and D. Yuan, "Learning-Based Link Scheduling in Millimeter-wave Multi-connectivity Scenarios," ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9148883.
[19] Z. Liu, Z. Chen, L. Luo, M. Hua, W. Li and B. Xia, "Age of Information-based Scheduling for Wireless Device-to-Device Communications using Deep Learning," 2021 IEEE Wireless Communications and Networking Conference (WCNC), 2021, pp. 1-6, doi: 10.1109/WCNC49053.2021.9417493.
[20] R. Shelim and A. S. Ibrahim, "Geometric Machine Learning Over Riemannian Manifolds for Wireless Link Scheduling," in IEEE Access, vol. 10, pp. 22854-22864, 2022, doi: 10.1109/ACCESS.2022.3153324.
[21] J. B. Oliva, B. P´oczos, and J. Schneider,"The Statistical Recurrent Unit", in ICML, 2017, pp. 2671-2680.
[22] A. Ribeiro de Miranda, T. M. G. de Andrade Barbosa, A. G. Scolari Conceição and S. G. Soares Alcalá, "Recurrent Neural Network Based on Statistical Recurrent Unit for Remaining Useful Life Estimation," 2019 8th Brazilian Conference on Intelligent Systems (BRACIS), 2019, pp. 425-430, doi: 10.1109/BRACIS.2019.00081.
[23] X. S. Nguyen, L. Brun, O. Lézoray and S. Bougleux, "Learning Recurrent High-order Statistics for Skeleton-based Hand Gesture Recognition," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 975-982, doi: 10.1109/ICPR48806.2021.9412036.
[24] Deepthi Sen, Hongquan Chen, Akhil Datta-Gupta, "Inter-well connectivity detection in CO2 WAG projects using statistical recurrent unit models, Fuel", Volume 311, 2022, 122600, ISSN 0016-2361, https://doi.org/10.1016/j.fuel.2021.122600.
[25] Nasim and A. S. Ibrahim, "Millimeter Wave Beamforming Codebook Design via Learning Channel Covariance Matrices Over Riemannian Manifolds," in IEEE Access, vol. 10, pp. 119617-119629, 2022, doi: 10.1109/ACCESS.2022.3222032.
[26] A. S. Ibrahim, "Rethinking Maximum Flow Problem and Beamforming Design through Brain-inspired Geometric Lens" Proc. the IEEE Global Communications Conference (Globecom'20), Dec. 2020, doi: 10.1109/GLOBECOM42002.2020.9321983.
[27] A. S. Ibrahim, "Wireless Link Scheduling via Interference-aware Symmetric Positive Definite Connectivity Manifolds," 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021, pp. 1-5, doi: 10.1109/ICCWorkshops50388.2021.9473707.
[28] Suvrit Sra and Reshad Hosseini, "Conic Geometric Optimization on the Manifold of Positive Definite Matrices," SIAM J. Optimization (SIOPT), vol. 25, no. 1, pp. 713–739, 2015.
[29] Suvrit Sra, "A New Metric on the Manifold of Kernel Matrices with Application to Matrix Geometric Means," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 144–152. Curran Associates, Inc., 2012.
[30] R. Chakraborty, C. Yang, X. Zhen, M. Banerjee, D. Archer, D. Vaillancourt, V. Singh, and B. C. Vemuri, "A statistical recurrent model on the manifold of symmetric positive definite matrices", in Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18).2018. Curran Associates Inc., Red Hook, NY, USA, 8897–8908.
[31] Sra, S., "Positive definite matrices and the S-divergence", Proceedings of the American Mathematical Society, 2016, 144(7), 2787–2797.
[32] H. Salehian, G. Cheng, B. C. Vemuri and J. Ho, "Recursive Estimation of the Stein Center of SPD Matrices and Its Applications," 2013 IEEE International Conference on Computer Vision, 2013, pp. 1793-1800, doi: 10.1109/ICCV.2013.225.
[33] M.P. do Carmo, Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition, Dover Books on Mathematics. Dover Publications, 2016.
[34] J. M. Lee, Introduction to Riemannian Manifolds, Springer, 2018, doi: 10.1007/978-3-319-91755-9.
[35] S. Gudmundsson, An Introduction to Riemannian Geometry, Lund University, 2019.
[36] Maurice Fréchet, "Les éléments aléatoires de nature quelconque dans un espace distancié." Ann. Inst. H. Poincaré, 10(3):215–310, 1948.
[37] A. Grami, Introduction to Digital Communications, 1st ed. New York, NY, USA: Academic, 2015.
[38] L. Dodero, H. Q. Minh, M. S. Biagio, V. Murino, and D. Sona, "Kernel based Classification for Brain Connectivity Graphs on the Riemannian Manifold of Positive Definite Matrices," in 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), April 2015, pp. 42–45.
[39] Roger A. Horn and Charles R. Johnson, "Matrix Analysis", Cambridge University Press, 1996, p.398
[40] Ian Goodfellow and Yoshua Bengio and Aaron Courville, "Deep Learning", MIT Press, 2016.
[41] Matthew D Zeiler, "Adadelta: An adaptive learning rate method", arXiv preprint arXiv:1212.5701, 2012.

Rashed Shelim is currently pursuing his Ph.D. degree at the Electrical and Computer Engineering Department at Florida International University (FIU), Miami, FL, USA. He received his BSc degree from North South University, Bangladesh (with summa cum laude) and MSc Degree from RWTH Aachen University, Germany. He is a senior lecturer at the Department of Electrical and Computer Engineering, North South University, Bangladesh (on study leave). Previously he worked as a Core Network Engineer (Packet Switching) at Huawei Technologies Bangladesh LTD. He also worked as a Research Assistant at P3 Communication GmbH, Germany. His principal research interests lie in the field of vehicular communications and rethinking wireless resource allocation by Geometric Machine Learning through the lens of Riemannian geometry.

AHMED S. IBRAHIM is currently an associate professor at the Electrical and Computer Engineering Department at Florida International University (FIU), Miami, FL, USA. He received the B.S. and M.S. degrees in electronics and electrical communications engineering from Cairo University, Cairo, Egypt, in 2002 and 2004, respectively. He received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2009. He is a recipient of the NSF CAREER Award in 2022. Prior to joining FIU, Dr. Ibrahim was an assistant professor at Cairo University, wireless research scientist at Intel Corporation, and senior engineer at Interdigital Communications Inc. Dr. Ibrahim's research interests span various topics of wireless systems including drone-assisted millimeter wave communications, vehicular communications, and rethinking wireless networks through the lens of Riemannian geometry.