Codebook Training for Trellis-Based Hierarchical Grassmannian Classification

Stefan Schwarz, *Senior Member, IEEE*, and Theodoros Tsiftsis, *Senior Member, IEEE*

***Abstract*—We consider classification of points on a complex-valued Grassmann manifold of *m*-dimensional subspaces within the *n*-dimensional complex Euclidean space. We introduce a trellis-based hierarchical classification network, which is based on an orthogonal product decomposition of the orthogonal basis representing the *m*-dimensional subspace. Exploiting the similarity of the proposed trellis classifier with a neural network, we propose stochastic gradient-based training techniques. We apply the proposed methods to two important applications in wireless communication, namely Grassmannian channel state information quantization in multiple-input multiple-output communications and non-coherent Grassmannian multi-resolution transmission.**

***Index Terms*—Grassmannian classification, CSI quantization, non-coherent transmission, trellis network training.**

I. Introduction

COMPLEX-VALUED Grassmann manifolds play a prominent role in wireless communications. Grassmannians have been successfully applied for non-coherent transmissions over block-fading channels [1]–[11], as well as, for channel state information (CSI) quantization in limited feedback based multiple-input multiple-output (MIMO) communications [12]–[16]. In both of these contexts, points on a Grassmann manifold have to be classified at the receivers, either to detect the transmit symbols or to quantize the CSI. Thus, computationally efficient Grassmannian classification is required to support a real-time application of such techniques. In [17], an autoencoder deep neural network has been proposed to tackle this problem.

Real-valued Grassmann manifolds have also found application in image classification problems [18]–[20]. In contrast to wireless communication problems, the dimensions in image classification are commonly much larger, however, computational complexity is not a major concern and therefore very complex classification networks can be implemented.

Contribution: In this letter, we consider a product decomposition of points on a Grassmann manifold as a basis for a hierarchical classifier. We approximate the joint hierarchical classification problem by a trellis-based classifier, generalizing our approach of [10] from one-dimensional to arbitrary *m*-dimensional subspaces. Our initial approach to this problem in [21] was a greedy recursive classifier, which has the advantage that it allows for an analytic performance investigation; however, it entails a significant performance loss compared to a joint classification. In [22], we have therefore generalized the recursive classifier to a tree-based structure, which allows to trade-off performance for complexity by performing a pruned tree search. The trellis classifier proposed in the present paper is basically an approximation of a full tree search, achieved by folding the branches of the tree on top of each other. This approach allows to perform classification on high-dimensional Grassmannians at relatively low complexity. It furthermore supports efficient classifier codebook training employing stochastic gradients and backpropagation, similar to a neural network. We apply the classifier to two important applications in wireless communications, namely Grassmannian CSI quantization and non-coherent Grassmannian transmission.

Notation: The Grassmann manifold of m-dimensional subspaces of the complex-valued n-dimensional Euclidean space is $\mathcal{G}(n,m)$, $n > m$. The conjugate-transpose of matrix $\mathbf{A}$ is $\mathbf{A}^{\mathsf{H}}$, the Frobenius norm is $\|\mathbf{A}\|$ and the d-th diagonal is diag($\mathbf{A}$, d). We use $\mathbf{I}_m$ for an $m \times m$ identity matrix and $\mathbf{1}_m$ for a length m all-ones vector. The subspace spanned by the columns of an orthogonal basis $\mathbf{U} \in \mathbb{C}^{n \times m}$ is $\operatorname{span}(\mathbf{U})$. The operation $a_{\max} = \arg\max_{a \in \mathcal{A}} f(a)$ determines the maximizer $a_{\max}$ of the function $f(a)$ over the set $\mathcal{A}$. The size of a set $\mathcal{A}$ is $|\mathcal{A}|$. The expected value of a random variable r is $\mathbb{E}(r)$.

II. Hierarchical Grassmannian Classification

A. System Model

We consider points on a complex-valued Grassmann manifold $\mathcal{G}(n,m)$, represented by corresponding orthogonal bases $\mathbf{U} \in \mathbb{C}^{n \times m}$, $\mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{I}_m$. These points are hierarchically classified within R Grassmann manifolds $\mathcal{G}(n, d_i)$ of decreasing subspace dimensions d_i, where $i \in \{1,\ldots,R\}$ is denoted as layer index. The subspace dimensions d_i satisfy $d_{i-1} > d_i$, with $d_0 = n$ and $d_R = m$; hence, $R < n - m$. We also define $\Delta_i = d_{i-1} - d_i$ as the subspace step-size of the i-th layer.

Hierarchical subspace classification is based on an orthogonal product decomposition of the subspace $\operatorname{span}(\mathbf{U})$ as follows

$$\mathbf{U} = \prod_{i=1}^{R} \mathbf{Q}^{(i)}, \ \mathbf{Q}^{(i)} \in \mathbb{C}^{d_{i-1} \times d_i}, \left(\mathbf{Q}^{(i)}\right)^{\mathsf{H}} \mathbf{Q}^{(i)} = \mathbf{I}_{d_i}. \quad (1)$$

Matrix $\mathbf{Q}^{(i)}$ represents a point on $\mathcal{G}(d_{i-1}, d_i)$ and the partial product $\mathbf{U}_r = \prod_{i=1}^{r} \mathbf{Q}^{(i)}$, being an orthogonal basis itself, corresponds to the hierarchical product decomposition of layer r with $\operatorname{span}(\mathbf{U}_r) \in \mathcal{G}(n, d_r)$.

B. Hierarchical Classifier

In hierarchical subspace classification, the exact product decomposition (1) is replaced by the following codebook based

Manuscript received December 6, 2021; accepted December 27, 2021. The associate editor coordinating the review of this article and approving it for publication was C.-K. Wen. *(Corresponding author: Stefan Schwarz.)*

Stefan Schwarz is with the CD-Lab for Dependable Wireless Connectivity, Institute of Telecommunications, Technische Universität Wien, 1040 Vienna, Austria (e-mail: stefan.schwarz@tuwien.ac.at).

Theodoros Tsiftsis is with the Institute of Physical Internet and School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai 519070, China.

Digital Object Identifier 10.1109/LWC.2021.3139166

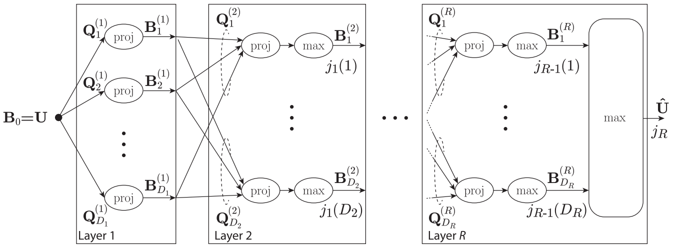

Fig. 1. Structure of a trellis-based Grassmannian hierarchical classification network consisting of R layers and D_i classes in layer i.

classification problem:

$$\hat{\mathbf{U}} = \prod_{i=1}^{R} \mathbf{Q}_{j_i}^{(i)},\ \mathbf{Q}_{j_i}^{(i)} \in \mathcal{Q}_{d_i}^{(d_{i-1})} \subset \mathcal{G}(d_{i-1}, d_i),$$
$$\left\{\mathbf{Q}_{j_1}^{(1)}, \ldots, \mathbf{Q}_{j_R}^{(R)}\right\} = \cdots$$
$$\underset{\mathbf{Q}_{\ell_1}^{(1)} \in \mathcal{Q}_{d_1}^{(d_0)}, \ldots, \mathbf{Q}_{\ell_R}^{(R)} \in \mathcal{Q}_{d_R}^{(d_{R-1})}}{\arg\max} \left\| \mathbf{U}^{\mathsf{H}} \prod_{i=1}^{R} \mathbf{Q}_{\ell_i}^{(i)} \right\|^2. \tag{2}$$

Here, the orthogonal bases $\mathbf{Q}_{j_i}^{(i)}$ of the R layers are taken from sub-codebooks $\mathcal{Q}_{d_i}^{(d_{i-1})} = \{\mathbf{Q}_1^{(i)}, \ldots, \mathbf{Q}_{D_i}^{(i)}\}$ of finite size $D_i = 2^{b_i}$. An optimal solution of this hierarchical classification problem requires a joint search over the product codebook $\mathcal{Q}_{d_R}^{(d_0)} = \mathcal{Q}_{d_1}^{(d_0)} \times \mathcal{Q}_{d_2}^{(d_1)} \times \ldots \times \mathcal{Q}_{d_R}^{(d_{R-1})}$ of size $D = 2^b$ with $b = \sum_{i=1}^{R} b_i$, which is practically not feasible for larger sub-codebook sizes.

C. Trellis-Based Classification

The joint classification (2) can be approximately solved by a trellis structure as illustrated in Fig. 1. In the first layer of this trellis the layer 1 input $\mathbf{B}_0 = \mathbf{U}$ is projected onto the subspaces $\mathbf{Q}_{\ell_1}^{(1)}$ of the first sub-codebook $\mathcal{Q}_{d_1}^{(d_0)}$

$$\mathbf{B}_{\ell_1}^{(1)} = \left(\mathbf{Q}_{\ell_1}^{(1)}\right)^{\mathsf{H}} \mathbf{B}_0,\ \mathbf{B}_{\ell_1}^{(1)} \in \mathbb{C}^{d_1 \times m}, \tag{3}$$
$$\mathbf{Q}_{\ell_1}^{(1)} \in \mathcal{Q}_{d_1}^{(d_0)},\ \ell_1 \ \in \{1, \ldots, D_1\}. \tag{4}$$

Matrices $\mathbf{B}_{\ell_1}^{(1)}$ are then fed into the second layer of the trellis-network, where each is again projected onto the corresponding layer 2 subspaces $\mathbf{Q}_{\ell_2}^{(2)} \in \mathcal{Q}_{d_2}^{(d_1)}$

$$\mathbf{B}_{\ell_2,\ell_1}^{(2)} = \left(\mathbf{Q}_{\ell_2}^{(2)}\right)^{\mathsf{H}} \mathbf{B}_{\ell_1}^{(1)} = \left(\mathbf{Q}_{\ell_1}^{(1)} \mathbf{Q}_{\ell_2}^{(2)}\right)^{\mathsf{H}} \mathbf{B}_0. \tag{5}$$

However, at this point we do not propagate all of these matrices further down the network, but rather decide in each path for one single matrix that provides the largest partial trellis metric

$$\mathbf{B}_{\ell_2}^{(2)} = \mathbf{B}_{\ell_2, j_1(\ell_2)}^{(2)}, \tag{6}$$
$$j_1(\ell_2) = \underset{\ell_1 \in \{1, \ldots, D_1\}}{\arg\max} \left\| \left(\mathbf{Q}_{\ell_2}^{(2)}\right)^{\mathsf{H}} \mathbf{B}_{\ell_1}^{(1)} \right\|^2. \tag{7}$$

In the same way, we proceed through the entire trellis until the last layer R of the network, where we additionally perform a final decision on the active trellis path

$$j_R = \underset{\ell_R \in \{1, \ldots, D_R\}}{\arg\max} \left\| \mathbf{B}_{\ell_R}^{(R)} \right\|^2. \tag{8}$$

The corresponding classified output is $\hat{\mathbf{U}} = \prod_{i=1}^{R} \mathbf{Q}_{j_i^*}^{(i)}$, where we use the short-hand notation j_i^* for the back-propagated indices of the active trellis path, i.e., $j_i^* = j_i(j_{i+1}(\ldots(j_{R-1}(j_R))\ldots))$, applying (7) recursively.

In general, this trellis approach performs slightly worse than a joint optimization; however, our prior studies [10] for $m = 1$ demonstrate that the loss is not significant. In terms of complexity the trellis approach provides a gain, as the total number of codebook searches is reduced from $2^b = 2^{\sum_{i=1}^{R} b_i}$ for the joint optimization to $2^{b_R} + \sum_{i=1}^{R-1} 2^{b_i} 2^{b_{i+1}}$. For example, for $b = 32$ bits equally distributed amongst $R = 8$ layers the number of codebook searches is reduced by a factor of more than 10^7. For a further complexity reduction, it is also possible to prune the trellis in each layer to a few paths with largest norms, without impairing the performance substantially.

D. Application Scenarios

The considered product classification (2) finds application in two scenarios that are relevant for wireless communications: a) Grassmannian CSI quantization in MIMO communications [21], [22]; b) Non-coherent multi-resolution transmission [10], [11]. We briefly outline this connection below.

1) Grassmannian CSI Quantization: In MIMO wireless communications, CSI at the transmitter (CSIT) about the wireless channel matrix $\mathbf{H} \in \mathbb{C}^{N_t \times N_r}$, where $N_t = n$ and $N_r = m < n$ denote the numbers of transmit and receive antennas, is frequently obtained by limited feedback from the receiver. For many transmit strategies, the relevant CSIT is the subspace $\mathrm{span}(\mathbf{H}) \in \mathcal{G}(n, m)$. Representing $\mathrm{span}(\mathbf{H})$ by an orthogonal basis $\mathbf{U}$, the CSI feedback is obtained from the following quantization problem

$$\hat{\mathbf{U}} = \underset{\mathbf{Q}_\ell \in \mathcal{Q}_m^{(n)}}{\arg\max} \left\| \mathbf{U}^{\mathsf{H}} \mathbf{Q}_\ell \right\|^2, \tag{9}$$

where $\mathcal{Q}_m^{(n)} \subset \mathcal{G}(n, m)$ denotes the finite size quantization codebook. For large-scale MIMO systems, the required codebook size $D = |\mathcal{Q}_m^{(n)}| = 2^b$ to achieve a sufficiently small quantization error can become prohibitively large [23]. A hierarchical product codebook together with the proposed trellis classifier can alleviate this problem.

2) Non-Coherent Multi-Resolution Transmission: In non-coherent Grassmannian transmission, the information is encoded in matrix $\mathbf{U} \in \mathbb{C}^{n \times m}$ taken from a Grassmannian symbol constellation/codebook $\mathcal{Q}_m^{(n)} \subset \mathcal{G}(n, m)$. $\mathbf{U}$ is transmitted during n time instances over the block-fading MIMO channel $\mathbf{H} \in \mathbb{C}^{N_t \times N_r}$, where $N_t = m$ and $N_r \geq m$. The corresponding input-output relationship is

$$\mathbf{Y} = \sqrt{\rho n}\, \mathbf{U}\mathbf{H} + \mathbf{Z}, \tag{10}$$

where $\mathbf{Z} \in \mathbb{C}^{n \times N_r}$ denotes the unit-variance channel noise and ρ is the signal-to-noise-ratio (SNR).

Non-coherent detection of $\mathbf{U}$ can be achieved by

$$\hat{\mathbf{U}} = \underset{\mathbf{Q}_\ell \in \mathcal{Q}_m^{(n)}}{\arg\max} \left\| \mathbf{U}_y^{\mathsf{H}} \mathbf{Q}_\ell \right\|^2, \tag{11}$$

where $\mathbf{U}_y \in \mathbb{C}^{n\times m}$ is an orthogonal basis for $\mathbf{Y}$'s left singular vectors corresponding to the m largest singular values.

In the non-coherent multi-resolution transmission scheme of [10], $\mathbf{U}$ is constructed from R streams utilizing (1)

$$\mathbf{U} = \prod_{i=1}^{R} \mathbf{Q}^{(i)}_{k_i},\ \mathbf{Q}^{(i)}_{k_i} \in \mathcal{Q}^{(d_{i-1})}_{d_i} \subset \mathcal{G}(d_{i-1}, d_i). \qquad (12)$$

Joint detection of these R streams leads to the same classification problem as in (2) and can therefore be efficiently approximated by the proposed trellis classifier.

E. Classifier Training

The subspaces $\mathbf{Q}^{(i)}_{\ell_i}$ of the individual layers of the trellis can be trained for different tasks, similar to the weights of a neural network. In this section, we develop unsupervised learning algorithms based on stochastic gradients and back-propagation, for our two specific application scenarios of Grassmannian CSI quantization and non-coherent multi-resolution transmission. In both cases, the training will in general only provide a local optimum; thus, multiple training runs with different initial random states of the subspaces $\mathbf{Q}^{(i)}_{\ell_i}$ can be advantageous.

1) Grassmannian CSI Quantization: In this application, the goal commonly is to find a quantization codebook that minimizes the average normalized chordal distance distortion

$$\bar{\mathrm{d}}^2_{\mathrm{c}}\left(\mathbf{U}, \hat{\mathbf{U}}\right) = 1 - \frac{1}{m}\mathbb{E}\left(\left\|\mathbf{U}^{\mathsf{H}}\hat{\mathbf{U}}\right\|^2\right). \qquad (13)$$

For isotropically distributed subspaces $\mathrm{span}(\mathbf{U})$ of the source samples, it is well known that this is achieved by maximally spaced subspace packings [24]. Yet, in general, the source samples may not be isotropically distributed, implying that the quantization sub-codebooks $\mathcal{Q}^{(d_{i-1})}_{d_i}$ of the trellis should then be adapted to the actual distribution of the source. This can be achieved by training the classification network with similar approaches as in deep neural network training, namely stochastic gradients and back-propagation.

Specifically, given a training sample $\mathbf{U}$, we first propagate $\mathbf{U}$ through the trellis and determine the corresponding active indices $\{j_1^*, \ldots, j_R^*\}$ and classified output $\hat{\mathbf{U}}$ according to (7) and (8). To train layer r, we update the corresponding codebook entry $\mathbf{Q}^{(r)}_{j_r^*}$ such as to increase the quantization metric $\|\mathbf{U}^{\mathsf{H}}\hat{\mathbf{U}}\|^2$ via a stochastic gradient step. We therefore calculate the gradient of $\|\mathbf{U}^{\mathsf{H}}\hat{\mathbf{U}}\|^2$ w.r.t. $\mathbf{Q}^{(r)}_{j_r^*}$

$$\nabla_r = \mathbf{A}_r^{\mathsf{H}}\mathbf{A}_r\mathbf{Q}^{(r)}_{j_r^*}\mathbf{B}_r\mathbf{B}_r^{\mathsf{H}},$$
$$\mathbf{A}_r = \prod_{i=1}^{r-1}\mathbf{Q}^{(i)}_{j_i^*}, \quad \mathbf{B}_r = \prod_{i=r+1}^{R}\mathbf{Q}^{(i)}_{j_i^*}. \qquad (14)$$

Next, we project this gradient onto the tangent-space of $\mathbf{Q}^{(r)}_{j_r^*}$ to find an ascent direction on the manifold [25]

$$\nabla_r^{(t)} = \left(\mathbf{I}_{d_r} - \mathbf{Q}^{(r)}_{j_r^*}\left(\mathbf{Q}^{(r)}_{j_r^*}\right)^{\mathsf{H}}\right)\nabla_r. \qquad (15)$$

Finally, we move along the geodesic defined by the projected gradient to update the codebook entry [26]

$$\mathbf{Q}^{(r)}_{j_r^*} \leftarrow \mathbf{Q}^{(r)}_{j_r^*}\mathbf{V}_r\cos(\boldsymbol{\Sigma}_r t)\mathbf{V}_r^{\mathsf{H}} + \mathbf{U}_r\sin(\boldsymbol{\Sigma}_r t)\mathbf{V}_r^{\mathsf{H}},$$
$$\nabla_r^{(t)} = \mathbf{U}_r\tilde{\boldsymbol{\Sigma}}_r\mathbf{V}_r^{\mathsf{H}},\ \boldsymbol{\Sigma}_r = \arctan\left(\tilde{\boldsymbol{\Sigma}}_r\right). \qquad (16)$$

Here, matrices $\mathbf{U}_r$, $\tilde{\boldsymbol{\Sigma}}_r$ and $\mathbf{V}_r$ are obtained from a compact singular value decomposition (SVD) of the projected gradient. The step-size/learning-rate t can be tuned for convergence speed and accuracy [27].

These stochastic gradients steps are performed recursively for each training sample $\mathbf{U}$, starting at the last layer R and back-propagating to the first layer. Thereby many calculations, especially for matrices $\mathbf{A}_r$ and $\mathbf{B}_r$, can be reused when moving from one layer to the next to reduce the training complexity.

Finally, these learning steps are repeated over a large training set of independent samples $\mathbf{U}$. To achieve convergence, the step-size t is reduced throughout the learning process.

2) Non-Coherent Multi-Resolution Transmission: For this application, it is known that the maximum likelihood detection performance is optimized if the minimum distance of the effective product symbol constellation $\mathcal{Q}^{(d_0)}_{d_R} = \mathcal{Q}^{(d_0)}_{d_1} \times \mathcal{Q}^{(d_1)}_{d_2} \times \cdots \times \mathcal{Q}^{(d_{R-1})}_{d_R}$ is maximized [1], [11]. We therefore propose to train the classifier to improve the pairwise minimum distance of constellation points as described below.

First, in each training iteration, we randomly sample a constellation point $\mathbf{U}$ from the current product symbol constellation $\mathcal{Q}^{(d_0)}_{d_R}$. We then propagate $\mathbf{U}$ through the trellis and determine the indices not only of the best trellis path corresponding to $\hat{\mathbf{U}} = \mathbf{U}$ according to (7), (8), but also of the second-best path:

$$j_R^{\mathrm{2nd}} = \underset{\ell_R \in \{1,\ldots,D_R\}\setminus j_R}{\arg\max} \left\|\mathbf{B}^{(R)}_{\ell_R}\right\|^2, \qquad (17)$$
$$j_i^{\mathrm{2nd}} = j_i(j_{i+1}(\ldots(j_{R-1}(j_R^{\mathrm{2nd}}))\ldots)), \qquad (18)$$

corresponding to $\hat{\mathbf{U}}^{\mathrm{2nd}}$.

To train layer r, we increase the distance between $\hat{\mathbf{U}}$ and its nearest trellis neighbor $\hat{\mathbf{U}}^{\mathrm{2nd}}$ via a stochastic gradient step. We therefore calculate the gradient ∇_r^{2nd} of $\|\mathbf{U}^{\mathsf{H}}\hat{\mathbf{U}}^{\mathrm{2nd}}\|^2$ w.r.t. $\mathbf{Q}^{(r)}_{j_r^{\mathrm{2nd}}} \in \mathcal{Q}^{(d_{r-1})}_{d_r}$, similar to (14), and project it onto the tangent space of $\mathbf{Q}^{(r)}_{j_r^{\mathrm{2nd}}}$, similar to (15).

As we intend to increase the pairwise distance, we now have to move along the geodesic defined by the negative projected gradient $-\nabla_r^{\mathrm{2nd},(t)}$. At the same time, however, we have to be careful not to move too close to other constellation points and thereby reduce the distance to them too much. This optimization therefore requires a smart selection of the step-size t.

Specifically, we perform a line-search over t, where we increase t as long as the minimum distance w.r.t. neighboring constellation points of $\hat{\mathbf{U}}^{\mathrm{2nd}}$ is improved. Neighboring constellation points in layer r are defined as those that differ from $\hat{\mathbf{U}}^{\mathrm{2nd}}$ only in the corresponding layer r codebook entry $\mathbf{Q}^{(r)}_{j_r^{\mathrm{2nd}}} \in \mathcal{Q}^{(d_{r-1})}_{d_r}$. Details are described in Algorithm 1.

III. Simulations

A. Grassmannian CSI Quantization

We consider quantization of orthogonal bases $\mathbf{U} \in \mathbb{C}^{64\times 4}$ corresponding to points on $\mathcal{G}(64, 4)$. These bases are generated from spatially correlated Rayleigh fading MIMO channels $\mathbf{H} \in$

Algorithm 1 Step-Size Line-Search

```
Initialize pairwise distance of active trellis pair
    $$\bar{\mathrm{d}}_{\mathrm{c}}^2\left(\hat{\mathbf{U}}, \hat{\mathbf{U}}^{2\mathrm{nd}}\right) = 1 - \frac{1}{m}\left\|\hat{\mathbf{U}}^{\mathsf{H}}\hat{\mathbf{U}}^{2\mathrm{nd}}\right\|^2$$
Initialize sub-constellation distances of layer $r$
    $$\bar{\mathrm{d}}_{\mathrm{c}}^2\left(\mathbf{Q}_{\ell_r}^{(r)}, \mathbf{Q}_{j_r^{2\mathrm{nd}}}^{(r)}\right) = 1 - \frac{1}{d_r}\left\|\left(\mathbf{Q}_{\ell_r}^{(r)}\right)^{\mathsf{H}}\mathbf{Q}_{j_r^{2\mathrm{nd}}}^{(r)}\right\|^2, \forall \ell_r \neq j_r^{2\mathrm{nd}}$$
Initialize current minimum distance
    $$\bar{\mathrm{d}}_{\min}^2 = \min\left\{\bar{\mathrm{d}}_{\mathrm{c}}^2\left(\hat{\mathbf{U}}, \hat{\mathbf{U}}^{2\mathrm{nd}}\right), \bar{\mathrm{d}}_{\mathrm{c}}^2\left(\mathbf{Q}_{\ell_r}^{(r)}, \mathbf{Q}_{j_r^{2\mathrm{nd}}}^{(r)}\right)\right\}$$
Initialize step-size $t$ and growth rate $g$ (e.g., $t = 0.01$, $g = 1.5$)
while Minimum distance $\bar{\mathrm{d}}_{\min}^2$ improves do
    Update constellation point $\mathbf{Q}_{j_r^{2\mathrm{nd}}}^{(r)}$ by moving along the geodesic defined by $-\nabla_r^{2\mathrm{nd},(t)}$ utilizing (16)
    Update minimum distance $\bar{\mathrm{d}}_{\min}^2$
    Update step-size $t \leftarrow t \cdot g$
end while
```

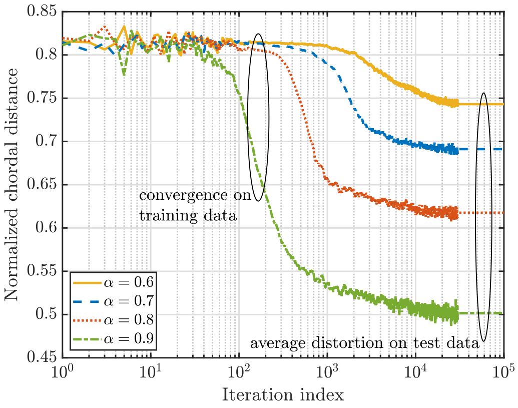

Fig. 2. Convergence behavior of codebook learning for correlated Gaussian channels of size $n \times m = 64 \times 4$ with correlation coefficient $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$. The total number of quantization bits is $b = 60$ bits.

$\mathbb{C}^{64\times 4}$ assuming a Kronecker correlation model [28]

$$\mathbf{H} = \mathbf{C}^{1/2}\tilde{\mathbf{H}}, \tag{19}$$

where $\tilde{\mathbf{H}} \sim \mathcal{CN}(0, 1)$ is an independent and identically distributed (i.i.d.) Rayleigh fading matrix and $\mathbf{C}$ is the correlation matrix, parametrized as $\mathrm{diag}(\mathbf{C}, d) = \alpha^{|d|}\mathbf{1}_{n-|d|}$. We train the trellis on a training data set consisting of $3e4$ samples and we evaluate the performance on a test data set of size $1e3$ samples.

1) Convergence: In Fig. 2, we exhibit the convergence behavior of the learning process for correlation coefficient $\alpha \in \{0.6, 0.7, 0.8, 0.9\}$. We consider a trellis with $R = 20$ layers and a sub-codebook size of $D_i = 2^{b_i} = 8, \forall i$; hence, the total number of quantization bits is $b = b_i R = 60$. During training, we gradually reduce the gradient step-size t from initially 0.1 down to 0.01. As we can see in Fig. 2, with lower correlation α learning only occurs later, when the step size is sufficiently small. Furthermore, we observe that the steady-state convergence value is the same as the average distortion achieved on test data.

2) Benchmark Comparison: We benchmark our scheme against k-means vector quantization as provided by MATLAB, since we are not aware of another applicable scheme that can handle large-dimensional correlated data. Applying k-means clustering directly on the 64×4 dimensional training samples is computationally not feasible for larger numbers of bits. To reduce complexity, we therefore partition $\mathbf{U}$ into non-overlapping vectors and apply k-means clustering on each vector individually. Specifically, we partition $\mathbf{U}$ into $\{4, 8, 16\}$ vectors of length $\{64, 32, 16\}$ and utilize a total of $b \in \{52, 104, 160\}$ bits to quantize the vectors. For reconstruction, we concatenate the quantized vectors and apply an SVD to obtain an orthogonal basis $\hat{\mathbf{U}}$ (post-orthogonalization).

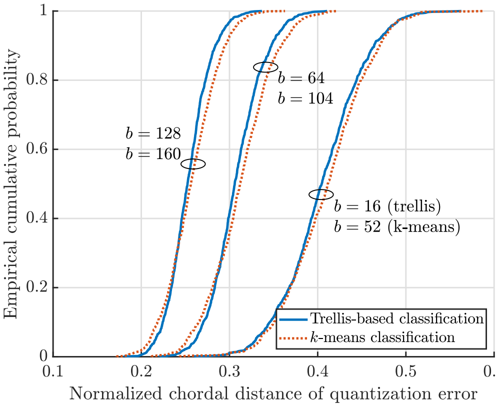

Fig. 3. Empirical cumulative distribution of the quantization error with trellis-based classification and k-means classification for correlated Gaussian channels of size $n \times m = 64 \times 4$ with correlation coefficient $\alpha = 0.9$.

For trellis-based quantization, we consider $b \in \{16, 64, 128\}$ bits, $R = 60$ layers and a dimension step-size of $\Delta_i = 1, \forall i$. Small Δ_i provides the lowest quantization complexity, because the total number of b bits is divided amongst many layers. Equal partitioning of the b bits amongst all layers does not achieve the lowest possible distortion. Thus, a hyper-parameter optimization over the number of b_i bits per layer is generally required. As a guideline for this hyper-parameter search we can utilize a result of [21]: the recursive multi-layer quantizer of [21] achieves the lowest distortion, when the bits are partitioned such that each layer contributes equal distortion. Although we have not yet been able to generalize this result to trellis-based quantization, it still provides a good starting point for a hyper-parameter search. For the relatively small number of bits considered in our simulation, compared to the dimensions of the quantization problem, it turns out that the bits should mostly be assigned to the latter layers of the trellis, whereas preceding layers get only single-element sub-codebooks assigned, which are trained to fit the mean value of the training samples $\mathbf{U}$. In our simulation, this leads to assigning bits only to the last 3, 10 and 18 layers for $b \in \{16, 64, 128\}$, respectively.

In Fig. 3, we show the results of our benchmark comparison. As can be seen the number of bits used for k-means has been selected such as to match the performance of trellis-based classification. Especially for smaller number of bits the proposed trellis quantizer is much more efficient than k-means with post-orthogonalization.

B. Non-Coherent Multi-Resolution Transmission

For non-coherent data transmissions, the minimum distance properties of the symbol constellation govern the achievable performance. We thus investigate the average pairwise minimum distance of our trained constellations, as a function of the number of b bits encoded in the constellation. As a benchmark, we utilize another numerically optimized symbol constellation, namely the direct constellation design of [29], in which the

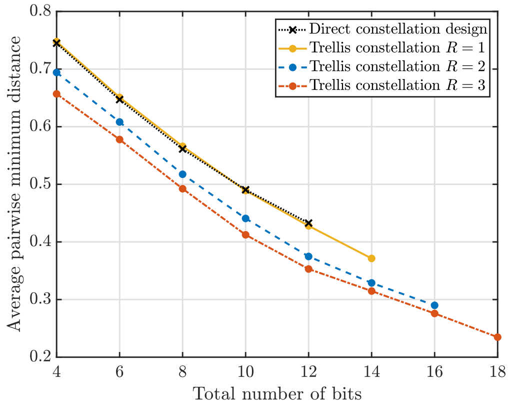

Fig. 4. Average pairwise minimum distance of our trellis-based symbol constellations compared to the direct design of [29]. The constellations are designed for transmission over $n = 8$ time instances and $m = 2$ antennas, with equal bit-partitioning amongst the R multi-resolution streams.

points of the symbol constellation are jointly optimized by applying a smooth approximation to maximizing the minimum distance. This optimization can be solved by stochastic gradient techniques, similar to our approach when the number of layers $R = 1$, with the difference that the gradient in [29] is jointly calculated from all constellation points, whereas we consider in each step only a single pair of constellation points, as described in Section II-E, to reduce the complexity.

The results are shown in Fig. 4 for different numbers R of non-coherent multi-resolution streams. For $R = 1$ our approach achieves the same performance as the direct optimization approach. However, the training complexity is lower as it is based only on a single pair of constellation points and therefore larger constellations can be trained. By increasing the number R of layers, the large product constellation is split into smaller sub-constellations per layer, which allows to train even larger symbol constellations, yet with slightly worse distances.

IV. Conclusion

We have presented a trellis-based hierarchical subspace classification network, which can be trained to perform different tasks. Specifically, we have applied the classifier in the contexts of Grassmannian CSI quantization and non-coherent multi-resolution transmissions. The main advantage of the proposed classifier is that it provides substantially reduced complexity compared to a single layer classifier and can therefore support classification on higher dimensional Grassmann manifolds.

Acknowledgment

The authors acknowledge TU Wien Bibliothek for financial support through its Open Access Funding Programme.

References

[1] B. M. Hochwald and T. L. Marzetta, "Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 543–564, Mar. 2000.

[2] L. Zheng and D. N. C. Tse, "Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 359–383, Feb. 2002.

[3] M. Beko, J. Xavier, and V. A. N. Barroso, "Noncoherent communication in multiple-antenna systems: Receiver design and codebook construction," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5703–5715, Dec. 2007.

[4] I. Kammoun, A. M. Cipriano, and J.-C. Belfiore, "Non-coherent codes over the Grassmannian," *IEEE Trans. Wireless Commun.*, vol. 6, no. 10, pp. 3657–3667, Oct. 2007.

[5] W. Yang, G. Durisi, and E. Riegler, "On the capacity of large-MIMO block-fading channels," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 117–132, Feb. 2013.

[6] K. Takeuchi, R. R. Müller, M. Vehkaperä, and T. Tanaka, "On an achievable rate of large Rayleigh block-fading MIMO channels with no CSI," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6517–6541, Oct. 2013.

[7] K. M. Attiah, K. Seddik, R. H. Gohary, and H. Yanikomeroglu, "A systematic design approach for non-coherent Grassmannian constellations," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2016, pp. 2948–2952.

[8] K. G. Seddik, R. H. Gohary, M. T. Hussien, M. Shaqfeh, H. Alnuweiri, and H. Yanikomeroglu, "Multi-resolution multicasting over the Grassmann and Stiefel manifolds," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5296–5310, Aug. 2017.

[9] K. Ngo, A. Decurninge, M. Guillaud, and S. Yang, "Cube-split: A structured Grassmannian constellation for non-coherent SIMO communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1948–1964, Mar. 2020.

[10] S. Schwarz, "Non-coherent broadcasting based on Grassmannian superposition transmission," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, Australia, Jul. 2021, pp. 1–6.

[11] S. Schwarz and B. Tahir, "Non-coherent multi-resolution broadcasting using Grassmannian product codebooks," in *Proc. 25th Int. ITG Workshop Smart Antennas (WSA)*, Nov. 2021, pp. 1–6.

[12] D. Love and R. Heath, "Limited feedback unitary precoding for spatial multiplexing systems," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2967–2976, Aug. 2005.

[13] V. Raghavan, R. W. Heath, and A. M. Sayeed, "Systematic codebook designs for quantized beamforming in correlated MIMO channels," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 7, pp. 1298–1310, Sep. 2007.

[14] O. El Ayach and R. W. Heath, "Grassmannian differential limited feedback for interference alignment," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6481–6494, Dec. 2012.

[15] A. Decurninge and M. Guillaud, "Cube-split: Structured quantizers on the Grassmannian of lines," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[16] S. Schwarz, M. Rupp, and S. Wesemann, "Grassmannian product codebooks for limited feedback massive MIMO with two-tier precoding," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 1119–1135, Sep. 2019.

[17] X. Fu and D. L. Ruyet, "Grassmannian constellation design for noncoherent MIMO systems using autoencoders," Sep. 2021, *arXiv:2109.00621*.

[18] J. Hamm and D. D. Lee, "Grassmann discriminant analysis: A unifying view on subspace-based learning," in *Proc. 25th Int. Conf. Mach. Learn.*, New York, NY, USA, Jul. 2008, pp. 376–383.

[19] M. T. Harandi, C. Sanderson, S. Shirazi, and B. C. Lovell, "Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching," in *Proc. CVPR*, Colorado Springs, CO, USA, Jun. 2011, pp. 2705–2712.

[20] Z. Huang, J. Wu, and L. Van Gool, "Building deep networks on Grassmann manifolds," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, Feb. 2018, pp. 3279–3286.

[21] S. Schwarz and M. Rupp, "Reduced complexity recursive Grassmannian quantization," *IEEE Signal Process. Lett.*, vol. 27, pp. 321–325, Jan. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8970542

[22] S. Schwarz and M. Rupp, "Tree-structured quantization on Grassmann and Stiefel manifolds," in *Proc. Data Compression Conf. (DCC)*, Mar. 2021, pp. 1–10.

[23] W. Dai, Y. Liu, and B. Rider, "Quantization bounds on Grassmann manifolds and applications to MIMO communications," *IEEE Trans. Inf. Theory*, vol. 54, no. 3, pp. 1108–1123, Mar. 2008.

[24] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: Packings in Grassmannian spaces," *Exp. Math.*, vol. 5, no. 2, pp. 139–159, Apr. 1996.

[25] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, Oct. 1998.

[26] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[27] L. Bottou, *On-line Learning and Stochastic Approximations* (Publications of the Newton Institute). Cambridge, U.K.: Cambridge Univ. Press, 1999, pp. 9–42.

[28] C. Oestges, "Validity of the Kronecker model for MIMO correlated channels," in *Proc. IEEE 63rd Veh. Technol. Conf.*, May 2006, pp. 2818–2822.

[29] R. H. Gohary and T. N. Davidson, "Noncoherent MIMO communication: Grassmannian constellations and efficient detection," *IEEE Trans. Inf. Theory*, vol. 55, no. 4, pp. 1176–1205, Mar. 2009.