Improving Channel Charting using a Split Triplet Loss and an Inertial Regularizer

Brian Rappaport[1], Emre Gönültaş[1], Jakob Hoydis[2], Maximilian Arnold[3], Pavan Koteshwar Srinath[4], and Christoph Studer[5]

[1]*School of Electrical and Computer Engineering, Cornell University; e-mail: br434@cornell.edu, eg566@cornell.edu*
[2]*NVIDIA, France; e-mail: jhoydis@nvidia.com*
[3]*Nokia Bell Labs Stuttgart; e-mail: maximilian.arnold@nokia-bell-labs.com*
[4]*Nokia Bell Labs France; e-mail: pavan.koteshwar_srinath@nokia-bell-labs.com*
[5]*Department of Information Technology and Electrical Engineering, ETH Zürich; e-mail: studer@ethz.ch*

***Abstract*—Channel charting is an emerging technology that enables self-supervised pseudo-localization of user equipments by performing dimensionality reduction on large channel-state information (CSI) databases that are passively collected at infrastructure base stations or access points. In this paper, we introduce a new dimensionality reduction method specifically designed for channel charting using a novel split triplet loss, which utilizes physical information available during the CSI acquisition process. In addition, we propose a novel regularizer that exploits the physical concept of inertia, which significantly improves the quality of the learned channel charts. We provide an experimental verification of our methods using synthetic and real-world measured CSI datasets, and we demonstrate that our methods are able to outperform the state-of-the-art in channel charting based on the triplet loss.**

I. Introduction

Hardly any area of modern research remains untouched by deep-learning-based machine learning. Wireless communications is no exception; in recent years, machine learning has been considered for many applications, including network routing, data detection, and others [1]–[3]. The rapid pace of data collection in modern wireless systems allows for data-driven approaches to address many pressing problems, including localization, the problem of estimating the position of a wireless transmitter given its channel state information (CSI). Wireless localization is a well-studied problem; CSI fingerprinting [4], [5] is one approach, but fingerprinting requires large CSI databases and access to ground-truth location, which is challenging to acquire as the channels change over time. To address such issues, we turn to self-supervised methods, which do not require ground truth location information.

A. Channel Charting: Main Idea

Of particular interest here is *channel charting* [6], a self-supervised framework for pseudo-localization and tracking of user equipments (UEs) using dimensionality reduction in the CSI domain. This technique learns a charting function over a specific channel, which maps high-dimensional CSI to low-dimensional pseudo-locations. Moreover, the function is learned in an self-supervised manner, without any need for ground truth location information. Successful channel charting relies on two aspects: (i) carefully-designed CSI features, as CSI measurements are prone to a variety of system and hardware impairments, such as noise, small-scale fading, time synchronization errors, and residual sampling rate and frequency offsets; and (ii) the dimensionality reduction method [6]–[10]. This paper focuses on the second aspect.

We consider one or more mobile UEs moving in the vicinity of an infrastructure base station (BS) or access point (AP), which regularly transmit pilots that are used to acquire large CSI datasets. The time-varying information can be used to build a channel chart with data from the same UE but taken at different time instants, rather than CSI measurements at the same time from different transmitters. The additional information that two consecutive samples are temporally adjacent also improves the quality of dimensionality reduction. Similar ideas have been considered in the past, e.g., the SM+ algorithm proposed in [6], but have been limited in scope.

B. Contributions

In this paper, we strive to take full advantage of all information available at the BS or AP when acquiring CSI measurements and learning the channel chart, both explicit and implicit. Concretely, we propose a new dimensionality reduction method we call the *split triplet loss*, which takes into account physical properties of the transmitting UEs during CSI acquisition. In addition, we augment our loss function using a novel *inertial regularizer*, which further improves the quality of the learned channel charting function. We demonstrate the efficacy of our methods using a synthetic and a real-world dataset, and compare them to Sammon's mapping from [8] and the recently-proposed triplet loss approach from [9].

C. Notation

We denote column vectors and matrices by boldface lowercase and capital letters, respectively. The ith element of a vector $\mathbf{a}$ is denoted by a_i, the ith column of a matrix $\mathbf{A}$ by $\mathbf{a}_i$, and the (i,j)th entry of $\mathbf{A}$ by A_{ij}. The transpose of $\mathbf{A}$ is $\mathbf{A}^T$,

The work of CS was supported in part by ComSenTer, one of six centers in JUMP, a SRC program sponsored by DARPA, and by an ETH Research Grant. The work of BR, EG, and CS was also supported by the US National Science Foundation (NSF) under grants CNS-1717559 and ECCS-1824379.

the conjugate (Hermitian) transpose is $\mathbf{A}^H$, and the (element-wise) conjugate is $\mathbf{A}^*$. We use the Euclidean norm for vectors and Frobenius norm for matrices. Element-wise powers on vectors and matrices are denoted using $\circ$, e.g., $(\mathbf{A}^{\circ 2})_{ij} = A_{ij}^2$.

II. Channel Charting Basics

Channel charting is a self-supervised pseudo-localization framework that applies dimensionality reduction to measured CSI [6]. In brief, channel charting takes a dataset of measured CSI, converts the CSI measurements into CSI features that are resilient to small-scale fading and hardware impairments [9], [11], and applies dimensionality reduction in order to learn a low-dimensional embedding: the *channel chart*. Our goal is to learn a channel chart that preserves local geometry: two nearby charted points should also be close in real space.

A. System Model

We consider a single-input multiple-output (SIMO) orthogonal frequency-division multiplexing system (OFDM) system in which one or multiple single-antenna UEs transmit pilots to a B-antenna BS or AP in time-division duplexing (TDD) fashion. While the UEs continuously broadcast pilot sequences for all W OFDM subcarriers, the BS or AP collects the SIMO-OFDM CSI of an UE in the form of a CSI matrix $\mathbf{H}_i \in \mathbb{C}^{B\times W}$, where i refers to the index of the measured CSI. In what follows, we assume that we are able to track the CSI associated with a specific UE over time, i.e., we can associate each CSI measurement index i with a particular UE u and corresponding time stamp t_i. The collection of CSI measurements $\{\mathbf{H}_i\}$ forms the dataset that is used to learn the channel chart. In what follows, we furthermore assume that there is an *unknown* ground-truth UE location $\mathbf{x}_i \in \mathbb{R}^D$ (with D being either two or three) associated with each measured CSI matrix $\mathbf{H}_i$.

B. CSI Feature Generation

Learning a channel chart directly from the CSI measurement dataset $\{\mathbf{H}_i\}$ is possible, but has several drawbacks that can be mitigated by carefully-crafted CSI features [6]. In particular, wireless channels are subject to (i) system impairments, mostly small-scale fading caused by moving objects between the transmitter and receiver, and (ii) hardware impairments, caused by varying transmit power, timing synchronization errors, as well as residual carrier frequency and sampling rate offsets. Following the work of [9], [11], we use an autocorrelation-based strategy to extract a feature from a raw CSI matrix $\mathbf{H}_i$.

1) Normalization: We normalize the CSI matrix $\mathbf{H}_i$ via $\mathbf{H}_i^{\text{norm}} = \mathbf{H}_i/\|\mathbf{H}_i\|$, which makes the CSI features independent of transmit-side power variations and path loss.

2) Delay-Domain Transform: We convert the normalized CSI matrix into the delay domain by applying an inverse discrete Fourier transform (DFT) to each row of $\mathbf{H}_i^{\text{norm}}$. Concretely, $\mathbf{H}_i^{\text{delay}} = \mathbf{H}_i^{\text{norm}}\mathbf{F}^H$, where $\mathbf{F}$ is the $W \times W$ unitary DFT matrix. In an OFDM system with many subcarriers and a short cyclic prefix length C, we can truncate the delay-domain CSI matrix by keeping only the C columns of $\mathbf{H}_i^{\text{delay}}$ corresponding to the impulse response's nonzero coefficients.

3) Beamspace Transform: We convert the delay-domain CSI matrix into the beamspace domain [12]. With a BS/AP utilizing a 1D uniform linear array (ULA) or a 2D uniform rectangular array (URA), we can apply a 1D or 2D DFT for each delay index. For the 1D case, we compute $\mathbf{H}_i^{\text{beamspace}} = \mathbf{F}\mathbf{H}_i^{\text{delay}}$; for the 2D case, we apply the 2D DFT to each column of $\mathbf{H}_i^{\text{delay}}$.

4) Autocorrelation: We compute an "instantaneous" 2D autocorrelation of the beamspace CSI matrix according to $\mathbf{H}_i^{\text{acorr}} = \mathbf{H}_i^{\text{beamspace}} * \mathbf{H}_i^{\text{beamspace}}$, where $*$ is the discrete-time convolution operator. This step renders our CSI features independent of time synchronization offset and global phase changes [11]. Note that the 2D autocorrelation of a matrix is rotationally symmetric, so we can truncate the redundant second half of the rows of $\mathbf{H}^{\text{acorr}}$.

5) Vectorization: We finish by vectorizing the entry-wise absolute values of the truncated autocorrelation matrix in order to extract our final CSI features: $\mathbf{f}_i = \text{vec}(|\mathbf{H}_i^{\text{acorr}}|)$.

C. Dimensionality Reduction: Sammon's Mapping

After generating CSI features, the next step is to learn a low-dimensional embedding. Dimensionality reduction is a well-studied problem [13] and linear dimensionality reduction, e.g., using principal component analysis (PCA), is widely used for many problems. Unfortunately, such methods do not work well for channel charting and nonlinear methods have been proposed instead [6]. Sammon's mapping [14] learns an embedding vector $\hat{\mathbf{x}}_i$ for each CSI feature $\mathbf{f}_i$ in a low (D') dimensional space. The embedding $\{\hat{\mathbf{x}}_i\}$, the set of points in the channel chart, can be calculated by minimizing the non-convex loss function

$$\mathcal{L}_{\text{Sammon}}(\{\hat{\mathbf{x}}_i\}) \triangleq \sum_{i,j} \frac{(\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\| - \|\mathbf{f}_i - \mathbf{f}_j\|)^2}{\|\mathbf{f}_i - \mathbf{f}_j\|}, \quad (1)$$

summed over all pairs of CSI features. Sammon's mapping attempts to learn an embedding $\{\hat{\mathbf{x}}_n\}$ in which pairs of CSI features with small Euclidean distance $\|\mathbf{f}_i - \mathbf{f}_j\|$ match the embedding distance $\|\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j\|$; pairs of points with large feature distances are deweighted. The loss can be approximately minimized efficiently using gradient-descent methods [6].

Although Sammon's mapping is appealingly simple, it has several drawbacks. It is non-parametric and suffers from the out-of-sample problem: given a new CSI feature $\mathbf{f}'$, finding the corresponding point $\mathbf{x}'$ in the channel chart requires either re-solving the entire optimization problem or using approximate out-of-sample extensions [15]. In [8], the authors address this issue using Siamese neural networks [16], two parallel and identical neural networks with shared parameter vector $\boldsymbol{\theta}$. They replace the embedded points $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_j$ in (1) with the outputs of a neural network $g_{\boldsymbol{\theta}}$ that takes in CSI features and produces points in the channel chart by setting $\hat{\mathbf{x}}_i = g_{\boldsymbol{\theta}}(\mathbf{f}_i)$ and $\hat{\mathbf{x}}_j = g_{\boldsymbol{\theta}}(\mathbf{f}_j)$. It can be formulated using the following loss function:

$$\mathcal{L}_{\text{Siamese}}(\boldsymbol{\theta}) \triangleq \sum_{i,j} \frac{(\hat{d}_{ij}^{\boldsymbol{\theta}} - \|\mathbf{f}_i - \mathbf{f}_j\|)^2}{\|\mathbf{f}_i - \mathbf{f}_j\|}, \quad (2)$$

where $\hat{d}_{ij}^{\boldsymbol{\theta}} \triangleq \|\hat{x}_i^{\boldsymbol{\theta}} - \hat{x}_j^{\boldsymbol{\theta}}\|$, and the network is optimized over the weights and biases subsumed in the vector $\boldsymbol{\theta}$. We note that the

embedding points $\hat{\mathbf{x}}_i$ and distances $\hat{d}_{ij}$ are functions of $\boldsymbol{\theta}$, but we omit these dependencies for ease of notation. Unfortunately, solving (2) relies heavily on the precept that the Euclidean distance between two CSI features is an accurate proxy for real pairwise distances. Although distance information is contained within the CSI features, the Euclidean distance is a poor choice to uncover it—a more appropriate CSI feature dissimilarity would allow us to learn more faithful channel charts.

D. Dimensionality Reduction: Triplet Loss

The triplet loss, originally from [17] and applied to channel charting in [9], avoids Euclidean pairwise distances altogether by learning a dissimilarity directly from the CSI features. This method is based on the concept of triplets: suppose we have access to a set $\mathcal{T} = \{(i,j,k) : d_{ij} \leq d_{ik}\}$, where $d_{ij} \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|$ is the true pairwise distance. For each triplet (i,j,k), we call the vector $\mathbf{x}_i$ the anchor point, $\mathbf{x}_j$ the positive (or close) point, and $\mathbf{x}_k$ the negative (or far) point. By comparing the distances from positive and negative points, we can learn an embedding which ensures that the ordering of real pairwise distances is maintained in the channel chart. As before, this method avoids the out-of-sample problem by learning a function $g_{\boldsymbol{\theta}}$ that maps CSI features to points in the channel chart. Reference [9] proposed the following triplet-based loss function for channel charting:

$$\mathcal{L}_{\text{Triplet}}(\boldsymbol{\theta}) \triangleq \sum_{(i,j,k)\in\mathcal{T}} [\hat{d}^{\boldsymbol{\theta}}_{ij} - \hat{d}^{\boldsymbol{\theta}}_{ik} + \lambda]^+. \tag{3}$$

The function $[x]^+ = \max\{x, 0\}$ for $x \in \mathbb{R}$ is the rectified linear unit (ReLU) and $\lambda \in \mathbb{R}$ is a user-defined margin parameter that determines the global scale of the learned embedding. Minimizing this loss function pulls negative points away from and positive points towards the anchor point, which means the ordering of pairwise distances in the embedding will match the true ordering. In some ways, this approach replaces one problem with another: feature distances are no longer required but one has to determine which points should be positive or negative with respect to an anchor point. In [9], the authors provide a solution based on sampling instants. Using the sampling rate of CSI measurements, they set a temporal bound on triplets: for a given anchor point, points close in time to it are labelled positive and those far in time to it are labelled negative.

The triplet loss has the following shortcomings: it does not make full use of the information available during CSI acquisition. Its generality makes it hard to attach physical significance to results; in particular, the scale of the embedding is entirely dependent on the parameter λ in (3). Additionally, it is difficult to tune this loss to make physical sense; the margin parameter λ has no physical significance, and it is difficult to select an appropriate margin value in a principled manner.

III. Split Triplet Loss with Inertial Regularizer

To address these concerns, we propose a novel loss function for channel charting, which we call the *split triplet loss*. We also propose an refined method of selecting triplets and a novel regularizer which encapsulates physical constraints on consecutive positions of a moving UE.

A. Split Triplet Loss

The triplet loss is most effective when pairwise distances are totally ordered: for any triple (i,j,k), it is possible to determine if $d_{ij} \leq d_{ik}$. For channel charting, pairwise distances are only partially ordered, especially when using the temporal triplet selection method described in Section II-D. It is generally difficult to determine whether $d_{ij} \leq d_{ik}$ for an arbitrary triple (i,j,k). However, for every anchor index i, it is possible to categorize all other points as either positive, negative, or neither. Consider what information we have about the triplets, given the temporal selection method of section Section II-D. Let $(i,j,k) \in \mathcal{T}$ be a triplet. If we know the maximum and minimum speeds the UEs are moving in space, v_{max} and v_{min}, then in T_c seconds, we know that the UE can move no further than $b_{\text{pos}} \triangleq v_{\text{max}} \cdot T_c$ and at least $b_{\text{neg}} \triangleq v_{\text{min}} \cdot T_c$. If the UE is moving in a straight line, then for the triplet loss, $d_{ij} \leq b_{\text{pos}}$ and $d_{ik} \geq b_{\text{neg}}$; if the maximum and minimum speeds are similar, it is quite likely that $d_{ik} \geq d_{ij}$, as required by the triplet loss. However, this information can be encoded more efficiently using two loss terms. The first term determines whether the positive point is close enough to the anchor; the second determines whether the negative point is far enough away. Instead of using constraints, we formulate these properties as regularizers: the first term is $[\hat{d}^{\boldsymbol{\theta}}_{ij} - b_{\text{pos}}]^+$, which is zero if the distance is less than the bound b_{pos} and increases linearly if not; the second term is $[b_{\text{neg}} - \hat{d}^{\boldsymbol{\theta}}_{ik}]^+$, which is zero if the distance is greater than b_{neg} and increases linearly as distance decreases. The overall loss function, the *split triplet loss*, is the sum of these two regularizers:

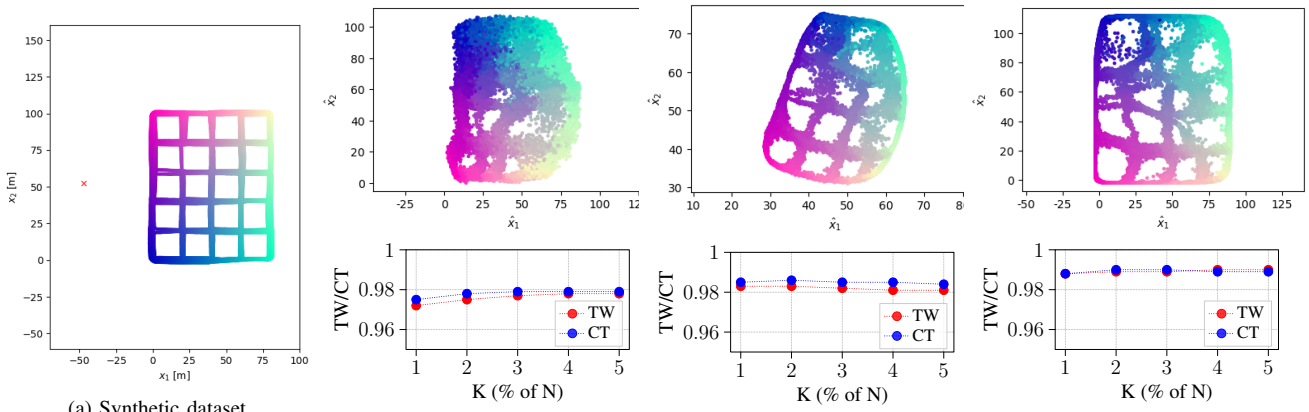
$$\mathcal{L}_{\text{Split Triplet}}(\boldsymbol{\theta}) \triangleq \sum_{(i,j,k)\in\mathcal{T}} [\hat{d}^{\boldsymbol{\theta}}_{ij} - b_{\text{pos}}]^+ + [b_{\text{neg}} - \hat{d}^{\boldsymbol{\theta}}_{ik}]^+. \tag{4}$$

This loss function considers physical constraints from the UE movement and measurement process. In contrast, the basic triplet loss does not reflect concepts like speed and does not use available time-stamp information during training.

B. Triplet Selection

As mentioned in Section II-D, triplets can be selected based solely on time stamps. In [9], an anchor point i is chosen at random from the dataset, at time stamp t_i; then the positive point j is chosen from $\{j : 0 < |t_i - t_j| < T_c\}$ and the negative point k is chosen from $\{k : T_c < |t_i - t_k| < T_f\}$, where T_c and T_f are user-defined parameters. Although intuitive, this approach cannot correctly identify self-intersecting tracks. If a UE crosses its own path from some time prior, points could be arbitrarily far in time but close in distance.

To address this problem, we return, somewhat surprisingly, to the feature distance. As mentioned before, feature distance is not generally a suitable proxy for real distance, but if the feature distance is close to zero, then the true distance is typically small. We leverage this observation as a way to detect self-intersections: if a point is far in time but close in feature distance (as measured using a quantile threshold over all feature distances), then it is likely to be a self-intersection point and should be treated as a nearby point instead.



(a) Synthetic dataset (b) Triplet, $\mu = 0$ (c) Triplet, $\mu = 0.2$ (d) Split triplet, $\mu = 0.2$

Fig. 1. Top: learned channel charts from the synthetic dataset (locations shown in Figure 1a); bottom: TW/CT dependent on neighborhood size K (% of $N = 17\,769$ CSI vectors). Left: conventional triplet loss as in [9]; middle: triplet loss with inertial regularizer ($\mu = 0.2$); right: proposed split triplet loss with inertial regularizer ($\mu = 0.2$). We observe that (i) inertial regularization significantly improves the channel chart and (ii) the proposed split triplet loss performs on par with the conventional triplet loss in terms of TW/CT but more resemble real locations as they occupy the entire area.

C. Inertial Regularization

Throughout this paper, we have been guided by the principle that we can and should use all information we have available to us. In a physical system, we can make a simple observation: the track of a body moving through real space is smooth. By this we mean that the UE's path must be continuous, so it cannot jump far away in a short time; this corresponds to the temporally-based triplet selection mechanism mentioned in Section III-B. But we can gleam additional information from this assertion: a body also cannot change its velocity discontinuously, which would imply infinite acceleration. In order to make use of this information, we propose a new regularizer which captures this inertia of the UE. The regularizer, which we unsurprisingly call the *inertial regularizer*, takes the following form:

$$\mathcal{L}_{\text{Inertia}}(\boldsymbol{\theta}) \triangleq \sum_{(i,j,\ell)\in\mathcal{I}} \|(\hat{\mathbf{x}}_j^{\boldsymbol{\theta}} - \hat{\mathbf{x}}_i^{\boldsymbol{\theta}}) - (\hat{\mathbf{x}}_i^{\boldsymbol{\theta}} - \hat{\mathbf{x}}_\ell^{\boldsymbol{\theta}})\|, \tag{5}$$

where $\mathcal{I} = \{(i,j,\ell) : t_i - t_\ell = t_j - t_i, |t_i - t_j| < T_c\}$, temporally linear triplets. The regularizer is the second difference of consecutive points: if i, j, and ℓ are equally spaced and on a straight line, then it will be zero, but it penalizes the UE slowing down, speeding up, or turning. In real motion, with a high enough sampling rate, the UEs move linearly or with gradual turns or changes in acceleration. Forcing this regularizer to be zero would result in linear, evenly spaced points. Instead, we deweight it using a suitably chosen regularization factor μ. We can modify the existing temporal triplet selection procedure to provide inertial points: for an anchor point i with positive point j, the inertial point is at time stamp $t_i - (t_j - t_i)$, for points close in time; we ignore inertial loss on points re-identified by feature distance (see Section III-B for the details).

IV. Results

We now provide results on synthetic and real-world measured CSI datasets: first, we introduce performance metrics and summarize the neural network structure used in the algorithms; second, we present our results compared to two benchmarks, Sammon's mapping via Siamese nets [8] and the triplet loss [9].

A. Performance Metrics

Channel charting is inherently a self-supervised learning problem, which means that standard error metrics, such as the mean-squared error, are not applicable. Instead, we consider the following performance metrics to assess channel chart quality.

1) Kruskal Stress: Kruskal stress [18] is commonly used in multidimensional scaling, the problem of finding a low-dimensional set of points which preserve distances in the high-dimensional space. We use the following definition [8]:

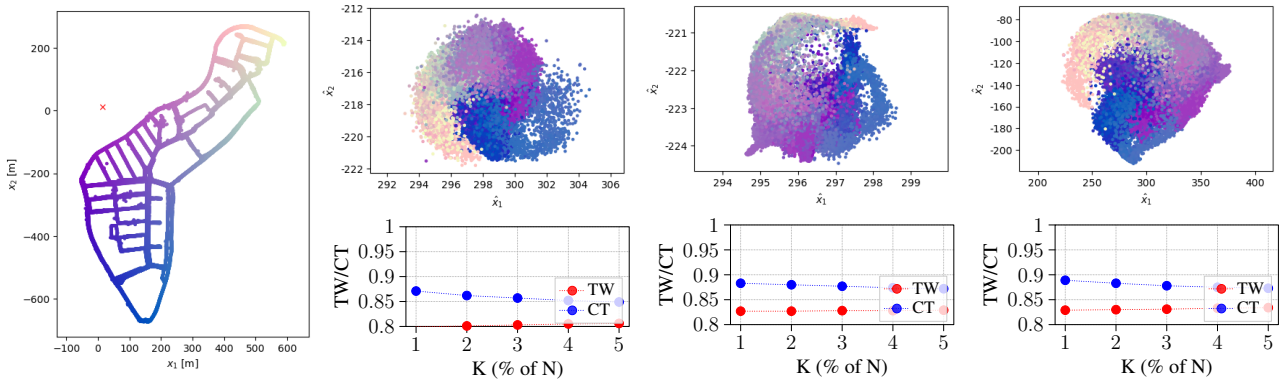
$$KS \triangleq \sqrt{1 - \left(\frac{\langle \mathbf{d}, \hat{\mathbf{d}} \rangle}{\|\mathbf{d}\| \cdot \|\hat{\mathbf{d}}\|}\right)^2}. \tag{6}$$

Here, $\mathbf{d}$ and $\hat{\mathbf{d}}$ are vectorized versions of the pairwise distance matrices formed by $\{d_{ij}\}$ and $\{\hat{d}_{ij}\}$; lower is better. This metric can also be interpreted in terms of the cosine distance between $\mathbf{d}$ and $\hat{\mathbf{d}}$: if $\frac{\langle \mathbf{d}, \hat{\mathbf{d}} \rangle}{\|\mathbf{d}\| \cdot \|\hat{\mathbf{d}}\|} = \cos\phi$, then $KS = \sin\phi$. The Kruskal stress in (6) is independent of scale, rotation, or translation, since it is measured using normalized distances. It is a useful similarity measurement for global distances, but does not specifically measure local embedding quality.

2) Optimal Shifting, Stretching, and Rotating: Since the channel charts are created using relative distances, they are invariant to translations and rotations. In addition, there cannot be a measure of absolute scale since the distances are given by a partial ordering. Accordingly, the optimal shifting, stretching, and rotating metric (SR) [9] is invariant to these parameters:

$$SR \triangleq \frac{1}{N} \sum_i \left\| \mathbf{D}_{\boldsymbol{\sigma}_x}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_x) - \mathbf{W}\mathbf{D}_{\boldsymbol{\sigma}_{\hat{x}}}^{-1}(\hat{\mathbf{x}}_i - \boldsymbol{\mu}_{\hat{x}}) \right\|^2. \tag{7}$$

Here, $\boldsymbol{\mu}_x = \frac{1}{N}\sum_i \mathbf{x}_i$, $\mathbf{D}_{\boldsymbol{\sigma}_x} = \text{diag}((\frac{1}{N}\sum_i (\mathbf{x}_i - \boldsymbol{\mu}_x)^{\circ 2})^{\circ 1/2})$, $\boldsymbol{\mu}_{\hat{x}}$ and $\mathbf{D}_{\boldsymbol{\sigma}_{\hat{x}}}$ are defined analogously, and $\mathbf{W}$ is the optimal projection from normalized embeddings onto normalized ground truth solving the orthogonal Procrustes problem,



(a) Real-world dataset (b) Triplet, $\mu = 0$ (c) Triplet, $\mu = 0.2$ (d) Split triplet, $\mu = 0.2$

Fig. 2. Top: learned channel charts from the real-world IEEE CTW 2020 data competition dataset (locations shown in Figure 2a); bottom: TW/CT dependent on neighborhood size K as a percentage of $N = 45\,784$ CSI vectors. Left: basic triplet loss as in [9]; middle: triplet loss with inertial regularizer ($\mu = 0.2$); right: proposed split triplet loss with inertial regularizer ($\mu = 0.2$). We note that (i) inertial regularization significantly improves the channel chart and (ii) the proposed split triplet loss outperforms the basic triplet loss in terms of TW/CT and more closely resemble real locations as they occupy the full area.

$\mathbf{W} = \mathbf{U}\mathbf{V}^T$ with the singular value decomposition $\mathbf{U}\mathbf{S}\mathbf{V}^T$ of $(\mathbf{D}_{\boldsymbol{\sigma}_{\hat{x}}}^{-1}(\hat{\mathbf{X}} - \mathbf{1}\boldsymbol{\mu}_{\hat{x}}^T))^T \mathbf{D}_{\boldsymbol{\sigma}_x}^{-1}(\mathbf{X} - \mathbf{1}\boldsymbol{\mu}_x^T)$, where $\mathbf{X}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{N \times D}$ are the matrices created by concatenating the N ground truth and learned locations, respectively [19].

3) Trustworthiness and Continuity: Local embedding quality uses a different set of metrics. Trustworthiness (TW) captures whether the neighbors in the embedding are also neighbors in the ground truth; continuity (CT) measures the converse [8]. Each is parametrized over a neighborhood of size K; we consider values of K from 1% to 5% of the total number of samples, N. If the set of K closest neighbors by Euclidean distance to a point i in the embedding is $\mathcal{U}_K(i)$, then

$$TW(K) \triangleq 1 - \frac{1}{(2N - 3K - 1)NK} \sum_i \sum_{j \in \mathcal{U}_K(i)} (r(i,j) - K), \quad (8)$$

where $r(i,j)$ is the order of $j \neq i$, ranked by distance from i in the ground truth. Similarly, if $\mathcal{V}_K(i)$ is the set of K closest neighbors to i in the ground truth, then

$$CT(K) \triangleq 1 - \frac{1}{(2N - 3K - 1)NK} \sum_i \sum_{j \in \mathcal{V}_K(i)} (\hat{r}(i,j) - K), \quad (9)$$

where $\hat{r}(i,j)$ is the analogous rank ordering in the embeddings; the leading coefficient is a normalization factor which ensures that $TW(K)$ and $CT(K)$ are between 0 and 1; higher is better.

B. Neural Network Structure

In the following experiments, we use the same neural network architecture for the function $g_{\boldsymbol{\theta}}$ that maps CSI features to points in the channel chart. We use a neural network with 2048 inputs, three dense layers, halving the number of nodes in each subsequent layer, and 256 outputs. We use ReLU activations except for the last layer, which generates probability maps [11] and uses a softmax function instead. This final layer outputs a probability mass function (PMF) $\mathbf{p}_i = \tilde{g}_{\boldsymbol{\theta}}(\mathbf{f}_i)$ over a set of predefined grid centroids $\{\mathbf{c}_k\}$ with $\mathbf{c}_k \in \mathbb{R}^{D'}$ which span the area, instead of predicting the pseudo-location in $\mathbb{R}^{D'}$ directly; p_k indicates the likelihood of the UE being at grid point c_k.

TABLE I
RESULTS FOR SYNTHETICALLY GENERATED NON-LOS DATA.

Method	μ	*KS*	*SR*	*TW* (5%)	*CT* (5%)
Sammon's mapping	0	0.525	1.132	0.647	0.731
	0.2	0.511	1.063	0.665	0.749
Triplet loss	0	0.182	0.080	0.978	0.979
	0.2	0.158	0.048	0.981	0.984
Split triplet loss	0	0.200	0.085	0.956	0.956
	0.2	**0.119**	**0.028**	**0.990**	**0.989**

We can then calculate the expected pseudo-location using the PMF and the 16×16 set of centroids, $\hat{\mathbf{x}}_i = \sum_k \mathbf{c}_k p_k$.

C. Synthetic Results

We evaluate the performance of our methods on a synthetic dataset with CSI generated using the QuadRiGa urban macro non-line-of-sight Berlin scenario [20]. We created a simple city-block MATLAB simulator in which a mobile UE travels randomly around a 4×4 grid of blocks. Using this simulator, we generated a dataset of $N = 17\,769$ channel vectors using a $B = 32$ antenna BS with a ULA and half wavelength antenna spacing. Thermal noise was added to the CSI vectors dependent on room temperature (300 K) and the bandwidth, and the signals were received using an OFDM system with $W = 64$ subcarriers at a bandwidth of 20 MHz. The BS is located around 50 m away from the UE locations; see Figure 1a for further details.

As shown in Figure 1, both algorithms are able to convincingly approximate the dataset. In Figure 1b, we compare the original algorithm from [9], the triplet loss with a margin $\lambda = 1$ and no inertia: this algorithm is able to recreate some of the city-block topology but contains irregularities. In Figure 1c, we add the inertial regularizer described in Section III-C to the triplet loss, which improves the local accuracy but negatively affects the scaling. Finally, in Figure 1d, we demonstrate the split triplet loss with the inertial regularizer, which has trustworthiness and continuity similar to that of the triplet loss but lower Kruskal stress and optimal stretch/rotation; see Table I for the details.

TABLE II
RESULTS FOR THE MEASURED IEEE CTW 2020 DATASET.

Method	μ	*KS*	*SR*	*TW* (5%)	*CT* (5%)
Sammon's mapping	0	0.610	1.564	0.540	0.576
	0.2	0.582	1.433	0.619	0.669
Triplet loss	0	0.464	0.841	0.806	0.849
	0.2	0.473	0.828	0.829	0.872
Split triplet loss	0	0.459	0.877	0.786	0.831
	0.2	**0.423**	**0.824**	**0.834**	**0.873**

D. Measured Results

We also demonstrate the performance of our methods on a real-world dataset consisting of channel matrices from the published dataset from [21] of the IEEE Communication Theory Workshop (CTW) 2020 data competition. The dataset covers a large suburban area (see Figure 2a). The UE in the field is equipped with a differential global positioning system (GPS), creating tagged CSI with 20 MHz of bandwidth at a carrier frequency of 1.27 GHz. The BS is equipped with a URA consisting of 8×8 antennas mounted on a high building complex. The measurement sanity was verified for massive MIMO precoding, thus capturing the spatial component of the environment correctly. Data was collected using SIMO-OFDM with $W = 924$ subcarriers for about 8 hours, resulting in $N = 45\,784$ CSI measurements. We calculated CSI features and averaged over 5 independent measurements.

It is harder to learn accurate embeddings on the real dataset, since it is larger and more sparsely sampled in space than the synthetic dataset, and the performance is correspondingly lower, as seen in Figure 2. We observe the same patterns of behavior as in the synthetic dataset. The benchmark algorithm with the triplet loss and no inertial regularization in Figure 2b is able to separate points roughly by location, but is poorly scaled and misshapen. When we add the inertia regularizer, Figure 2c shows slightly improved performance (it is able to separate the blue "elbow" of points at the bottom of the ground truth around (150,-600) in Figure 2a from the other points, for example) but still suffers from overlapping points and a bad scale. Finally, in Figure 2d we show the triplet loss combined with the inertial regularizer, which is able to outperform the other methods, once again marginally in terms of SR, TW, and CT and more so in Kruskal stress; see Table II for the details.

V. CONCLUSIONS

We have proposed the split triplet loss for channel charting, which takes into account physical constraints from the CSI measurement process, as well as a suitable triplet selection procedure and a novel inertial regularizer that incorporates the fact that moving UEs must have finite acceleration/deceleration. We have demonstrated the efficacy of our methods on a real-world dataset, which reveals that our methods outperform the Siamese network of [8] and the state-of-the-art triplet loss of [9] in terms of Kruskal stress, optimal stretching/rotating, trustworthiness, and continuity. We leave several problems as future work: an obvious extension is to utilize multiple BSs/APs as well as semi-supervised techniques as in [8] to enable absolute positioning capabilities with channel charting.

ACKNOWLEDGMENTS

The authors would like to thank Olav Tirkkonen and Pengzhi Huang for discussions on channel charting.

REFERENCES

[1] A. Forster, "Machine Learning Techniques Applied to Wireless Ad-Hoc Networks: Guide and Survey," in *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007, pp. 365–370.

[2] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3072–3108, 2019.

[3] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2016.

[4] Y. Zhang, D. Li, and Y. Wang, "An indoor passive positioning method using CSI fingerprint based on adaboost," *IEEE Sensors J.*, vol. 19, no. 14, pp. 5792–5800, Mar. 2019.

[5] V. Savic and E. G. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," in *Proc. IEEE 82nd Vehic. Tech. Conf.*, Sep. 2015, pp. 1–5.

[6] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47 682–47 698, Aug. 2018.

[7] P. Huang, O. Castañeda, E. Gönültaş, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, "Improving channel charting with representation-constrained autoencoders," in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, Aug. 2019, pp. 1–5.

[8] E. Lei, O. Castañeda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *Proc. 57th Ann. Allerton Conf. Commun., Control, and Comput.*, Sep. 2019, pp. 200–207.

[9] P. Ferrand, A. Decurninge, L. G. Ordoñez, and M. Guillaud, "Triplet-based wireless channel charting," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[10] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based localization from channel estimates: Feature design and experimental results," *ArXiv preprint: 2004.00363*, Apr. 2020.

[11] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-based multi-antenna and multi-point indoor positioning using probability fusion," *ArXiv preprint: 2009.02798*, Sep. 2020.

[12] J. Brady, N. Behdad, and A. M. Sayeed, "Beamspace MIMO for millimeter-wave communications: System architecture, modeling, analysis, and measurements," *IEEE Trans. Antennas Propag.*, vol. 61, no. 7, pp. 3814–3827, Mar. 2013.

[13] L. van der Maaten, E. Postma, and H. Herik, "Dimensionality reduction: A comparative review," *J. Mach. Learn. Res.*, vol. 10, Jan 2007.

[14] J. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401–409, 1969.

[15] T. Ponnada, H. Altous, O. Tirkkonen, and C. Studer, *An Out-of-Sample Extension for Wireless Multipoint Channel Charting*, Jul 2019, pp. 208–217.

[16] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, pp. 737–744, 1993.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[18] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[19] P. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, pp. 1–10, 1966.

[20] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, Mar. 2014.

[21] M. Arnold, J. Hoydis, and S. ten Brink, "Novel massive MIMO channel sounding data applied to deep learning-based indoor positioning," in *Intl. ITG Conf. Systems, Commun., Coding*, Feb. 2019, pp. 1–6.