Tensor-Generative Adversarial Network with Two-dimensional Sparse Coding: Application to Real-time Indoor Localization

Chenxiao Zhu[1], Lingqing Xu[2], Xiao-Yang Liu[2,3], and Feng Qian[1]
[1]The Center for Information Geoscience, University of Electronic Science and Technology of China
[2]Dept. of Electrical Engineering, Columbia University
[3]Dept. of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract— **Localization technology is important for the development of indoor location-based services (LBS). Global Positioning System (GPS) becomes invalid in indoor environments due to the non-line-of-sight issue, so it is urgent to develop a real-time high-accuracy localization approach for smartphones. However, accurate localization is challenging due to issues such as real-time response requirements, limited fingerprint samples and mobile device storage. To address these problems, we propose a novel deep learning architecture: Tensor-Generative Adversarial Network (TGAN).**

We first introduce a transform-based 3D tensor to model fingerprint samples. Instead of those passive methods that construct a fingerprint database as a prior, our model applies artificial neural network with deep learning to train network classifiers and then gives out estimations. Then we propose a novel tensor-based super-resolution scheme using the generative adversarial network (GAN) that adopts sparse coding as the generator network and a residual learning network as the discriminator. Further, we analyze the performance of tensor-GAN and implement a trace-based localization experiment, which achieves better performance. Compared to existing methods for smartphones indoor positioning, that are energy-consuming and high demands on devices, TGAN can give out an improved solution in localization accuracy, response time and implementation complexity.

Keywords*—*Indoor localization, RF fingerprint, tensor-based generative adversarial network, two-dimensional sparse coding, smartphones

I. Introduction

In recent years, the world has witnessed a booming number of smartphones, which apparently has had a profound influence on personal lives. Meanwhile, the application of Internet of Things (IoT) has promoted the development of information technology. The real leap forward comes through the combination of IoT and mobility, which offers us opportunities to re-imagine the ideal life. One promising application of IoT on mobile devices is indoor positioning. More recently, location-based services (LBS) have been used in airports, shopping malls, supermarkets, stadiums, office buildings, and homes [1][2]. Generally, there are three approaches for indoor localization systems, viz., cell-based approach, model-based approach, and fingerprint-based approach. In this paper, we will focus on radio frequency (RF) fingerprint-based localization, which is one of the most promising approaches.

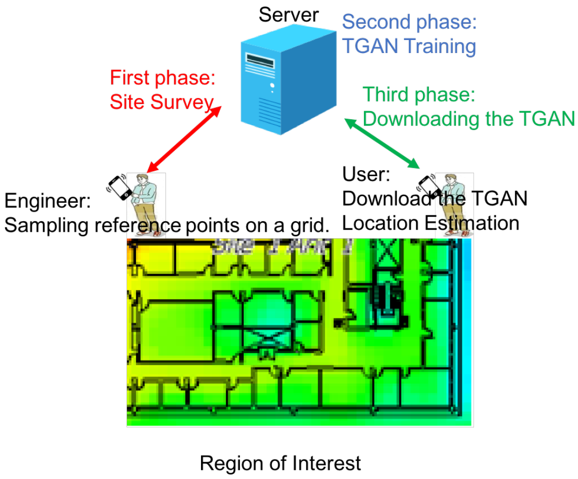

Fig. 1: Our indoor localization scenario. The engineer samples RF fingerprints in ROI on a grid, then the server uses samples to train a classification TGAN. The user downloads the trained TGAN and uses a newly sampled fingerprint to estimate current location.

RF fingerprint-based indoor localization [3][4] consists of two phases: training and operating. In the training phase, radio signal strength (RSS) fingerprints are collected from different access points (APs) at reference points (RPs) in the region of interest (ROI). The server uses the collected fingerprint samples to train a TGAN, i.e., taking the fingerprint samples as feature vectors while RPs' coordinates as labels. Then in the operating phase, a user downloads the trained GAN and uses the newly sampled fingerprint to obtain the location estimation. Fig 1 demonstrates the schema of the indoor localization scenario.

Existing works on RF fingerprint-based indoor localization have the following limitations. First, for the training phase, constructing a fingerprint database as a prior is unpractical due to lack of high-granularity fingerprint samples we can collect [3][4]. Secondly, for a mobile device, storage space may not meet the minimum requirement of storing fingerprint database, and the computing capability, usually at tens or hundreds of Million Floating-point Operations per Second (MFLOPs), is also insufficient when encounter such data. Relying on server for storage and computation raises the challenge of communication delay for real-time localization application. In

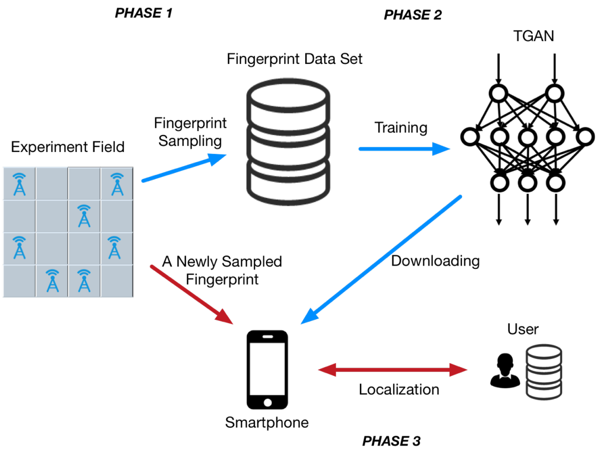

Fig. 2: Our localization process can be separated into three phases; phase 1 is the fingerprint sampling as [3] and phase 2 is the training process of our TGAN. In phase 3, the user downloads the trained TGAN, inputs a newly sampled fingerprint and obtains the location estimation.

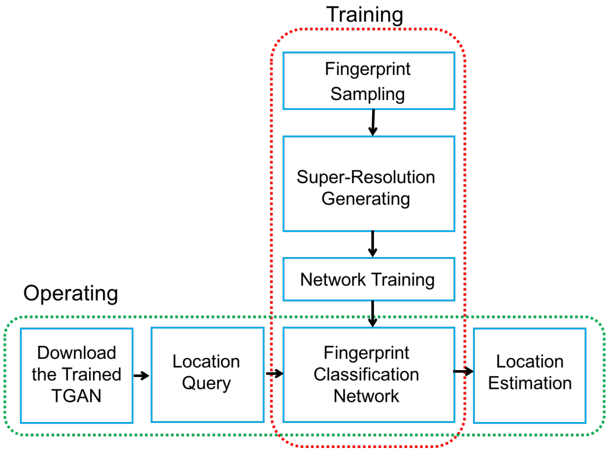
Training
Fingerprint Sampling
Super-Resolution Generating
Network Training
Operating
Download the Trained TGAN
Location Query
Fingerprint Classification Network
Location Estimation

Fig. 3: The training phase is completed by engineers and the operation phase by users.

this context, we exploit a novel indoor localization approach based on deep learning.

Inspired by the powerful performance of generative adversarial networks (GAN) [5][6], we propose a tensor-based generative adversarial network (TGAN) to achieve super-resolution of the fingerprint tensor. The advantage of GAN lies in recovering the texture of input data, which is better than other super-resolution approaches. We also adopt transform-based approach introduced in [7] to construct a new tensor space by defining a new multiplication operation and tensor products. We believe with these multilinear modeling tool, we can process multidimensional-featured data more efficient with our tensor model.

Our paper is organized as follows. Section II presents details about transform-based tensor model with real-world trace verification. Section III describes the architecture of TGAN and gives two algorithms for high-resolution tensor derivation and the training process. Section IV introduces a evaluation of TGAN and implements a trace-based localization experiment. Conclusions are made in Section V.

II. Transform-Based Tensor Model

In this section, we model RF fingerprints as a 3D low-tubal-rank tensor. We begin by first outlining the notations and taking a transform-based approach to define a new multiplication operation and tensor products under multilinear tensor space as in [3]. We suppose, in the multilinear tensor space, multidimensional-order tensors act as linear operators which is alike to conventional matrix space. In this paper, we use a third-order tensor $\mathcal{T}^{n_1\times n_2\times n_3}$ to represent the RSS map. Specifying the discrete transform of interest to be a discrete Fourier transform, the $\bullet$ operation represents circular convolution.

Notations- In this paper, scalars are denoted by lowercase letters. e.g. n; vectors are denoted by boldface lowercase letters e.g. $\boldsymbol{a}$ and the transpose is denoted as $\boldsymbol{a}^\top$; matrices are denoted by boldface capital letters e.g. $\boldsymbol{A}$; and tensors are denoted by calligraphic letters e.g. $\mathcal{T}$ and the transpose is denoted by $\mathcal{T}^\top$. We use $[n]$ to denote the set $\{1,2,...,n\}$. The ℓ_1 norm of tensor is denoted as $\|\mathcal{T}\|_1 = \sum_{i,j,k}|\mathcal{T}(i,j,k)|$. The Frobenius norm of a tensor is defined as $\|\mathcal{T}\|_F = \left(\sum_{i,j,k}\mathcal{T}(i,j,k)^2\right)^{1/2}$.

Definition 1. Tensor product*: For two tensors* $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$*, and* $\mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$*. The tensor product* $\mathcal{C} = \mathcal{A} * \mathcal{B} \in \mathbb{R}^{n_1\times n_4\times n_3}$*. The* $\mathcal{C}(i,j,:)$ *is a tube given by* $\mathcal{C}(i,j,:) = \sum_{l=1}^{n_2}\mathcal{A}(i,l,:)\bullet\mathcal{B}(l,j,:)$*, for* $i\in[n_1]$ *and* $j=[n_4]$*.*

Definition 2. Identity tensor*: A tensor* $\mathcal{I} \in \mathbb{R}^{n_1\times n_2\times n_3}$ *is identity if* $\mathcal{I}(:,:,1)$ *is identity matrix of the size* $n_1\times n_1$ *and other tensor* $\mathcal{I}(:,:,i), i\in[n_3]$ *are all zeros.*

Definition 3. Orthogonal tensor*:* $\mathcal{T}\in\mathbb{R}^{n_1\times n_2\times n_3}$ *is orthogonal if* $\mathcal{T}*\mathcal{T}^\top = \mathcal{T}^\top*\mathcal{T} = \mathcal{I}$*.*

Definition 4. f-diagonal tensor*: A tensor is f-diagonal if all* $\mathcal{T}(:,:,i), i\in[n_3]$ *are diagonal matrices.*

Definition 5. t-SVD*:* $\mathcal{T}\in\mathbb{R}^{n_1\times n_2\times n_3}$*, can be decomposed as* $\mathcal{T} = \mathcal{U} * \boldsymbol{\Theta} * \mathcal{V}^\top$*, where* $\mathcal{U}$ *and* $\mathcal{V}$ *are orthogonal tensors of sizes* $n_1\times n_1\times n_3$*,* $n_2\times n_2\times n_3$ *individually and* $\boldsymbol{\Theta}$ *is a rectangular f-diagonal tensor of size* $n_1\times n_2\times n_3$*.*

Definition 6. Tensor tubal-rank*: The tensor tubal-rank of a third-order tensor is the number of non-zero fibers of* $\boldsymbol{\Theta}$ *in t-SVD.*

In this framework, we can apply t-SVD [3][7], which indicates that the indoor RF fingerprint samples can be modeled into a low tubal-rank tensor. [7] gives out the decomposition measurement of t-SVD regarding normal matrix-SVD, which explains t-SVD may be a suitable approach for further processing. To verify this low-tubal-rank property, we obtain a real-world data set from [8] in an indoor region of size 20m × 80m represented as Fig. 4, located in a college building.

The region is divided into 64×268 grid map with gird size $0.3\text{m}\times 0.3\text{m}$, and there are 21 APs randomly deployed. Therefore, our ground truth tensor is of size $64\times 256\times 21$. The Fig. 5 demonstrates that compared with matrix-SVD and CP decomposition [9], the t-SVD shows RF fingerprint data

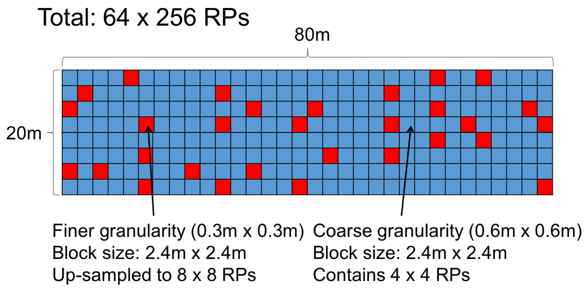

Fig. 4: There are 64×256 RPs uniformly distributed in the $20\text{m} \times 80\text{m}$ region. The blue blocks represent the finer granularity ones and the red blocks represent the coarse granularity ones

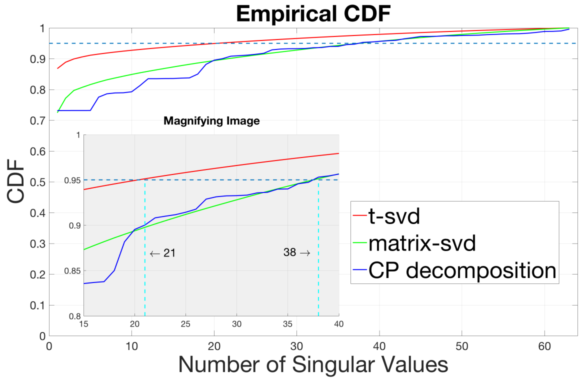

Fig. 5: The CDF of singular values for t-SVD, matrix-SVD and CP decomposition. It shows that the low-tubal-rank tensor model better captures the correlations of RF fingerprints.

are more likely to be in the form of a low tubal-rank structure. For t-SVD, 21 out of the total 64 singular values capture 95% energy of the fingerprint tensor, and the amount is increased to 38 for the matrix-SVD method and CP decomposition method. Therefore, the low tubal-rank property of transform-based tensor model is more suitable for the RSS fingerprint data than matrix.

In this section, we compare three decomposition methods and decide to utilize the transform-based tensor model that are constructed under a new tensor space with 2D sparse coding. With the tensor calculations defined [7], we are able to extend the conventional matrix space to third-order tensors. This framework enables a wider tensor application. In the next section, we will introduce generative adversarial networks (GAN) to implement super-resolution based on the tensor model we propose.

III. Tensor Generative Adversarial Network

TGAN is used to generate extra samples from coarse RF fingerprint samples. The ultimate goal is to train a generator G that estimates a finer granularity tensor $\mathcal{T}^{'}$ from a coarse granularity input tensor $\mathcal{T}$. TGAN consists of three parts: the generator named tensor sparse coding network (TSCN), the discriminator and the localization network named localizer. In this section, we introduce whole architecture of TGAN and illustrates the data delivery and parameter updating process in TGAN as show in Fig. 6.

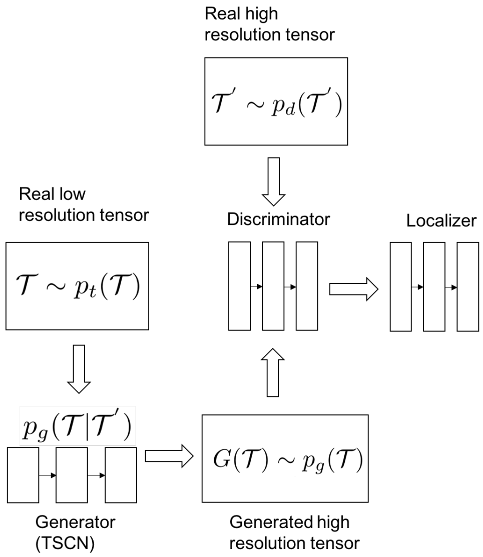
Real high resolution tensor
$\mathcal{T}^{'} \sim p_d(\mathcal{T}^{'})$
Real low resolution tensor
$\mathcal{T} \sim p_t(\mathcal{T})$
Discriminator
Localizer
$p_g(\mathcal{T}|\mathcal{T}^{'})$
$G(\mathcal{T}) \sim p_g(\mathcal{T})$
Generator (TSCN)
Generated high resolution tensor

Fig. 6: The architecture, data delivery and parameter updating of TGAN.

TSCN is used to encode the input coarse granularity tensor $\mathcal{T}$ into a sparse representation $\mathcal{A}$ and its dictionary $\mathcal{D}$ with feedback from discriminator. Therefore, TSCN can be regarded as two parts. The first step is figuring out the sparse representation $\mathcal{A}$ and the dictionary $\mathcal{D}$ by implementing Learned Iterative Shrinkage and Thresholding Algorithm based on Tensor (LISTA-T) and three fully-connection layers of network to receive feedback from discriminator. Then the output of the LISTA-T algorithm is modified. This operation scheme is similar to the function of sparse autoencoder, since the TSCN's target function (2) has the same format with autoencoder.

Let $\mathcal{T}^{'}$ be the input, $p_g(\mathcal{T}^{'})$ be the model distribution and $p_t(\mathcal{T}^{'})$ be the data distribution. $p_g(\mathcal{T}|\mathcal{T}^{'})$ can be regard as encoding distribution of auto encoder, or as TSCN correspondingly. Therefore, we can define that $p_g(\mathcal{T}^{'}) = \int_{\mathcal{T}} p_g(\mathcal{T}|\mathcal{T}^{'})\ p_t(\mathcal{T})d\mathcal{T}$. Let $p_d(\mathcal{T})$ be the arbitrary prior distribution of $\mathcal{T}$. By matching the aggregated posterior $p_g(\mathcal{T})$ to an arbitrary prior distribution $p_d(\mathcal{T}^{'})$, the TSCN can be regularized. In the following work, TSCN parametrized by θ_G is denoted as G_{θ_G}. Then, finding a suitable TSCN for generating a finer granularity tensor $T^{'}$ out of coarse tensor T can be represent as: $\widehat{\theta}_G = \arg\min \frac{1}{L}\sum_{l=1}^{L} \|G_{\theta_G}(\mathcal{T}_l), \mathcal{T}_l^{'}\|_F^2$. The target adversarial min-max problem is:

$$\begin{aligned} \min\nolimits_{\theta_G} \max\nolimits_{\theta_D} & \mathbb{E}_{\mathcal{T}' p_d(\mathcal{T}')} \log D_{\theta_d}(\mathcal{T}^{'}) + \\ & \mathbb{E}_{\mathcal{T} p_g(\mathcal{T})} \log\left(1 - D_{\theta_d}(G_{\theta_G}(\mathcal{T}))\right), \end{aligned} \tag{1}$$

The encoding procedure from a coarse granularity tensor $\mathcal{T}$ to a higher resolution tensor $\mathcal{T}^{'}$ of a tensor can be regarded as a super-resolution problem. The problem is ill-posed and we model two constraints to solve it. The first one is that

Algorithm 1 Super-resolution via Sparse Coding

```
Input: trained dictionaries $\mathcal{D}^{'}$ and $\mathcal{D}$, a coarse granularity fingerprint image $\mathcal{Y}$.
Output: Super-Resolution fingerprint image $\mathcal{X}$.
Solve $\mathcal{D}$ and $\mathcal{D}^{'}$ via algorithm 2.
for Each patch of $\mathcal{Y}$ do
    Use $\mathcal{D}$ to solve the optimization problem for $\mathcal{A}$ in (2),
    Generate the finer granularity patch $\mathcal{X} = \mathcal{D}^{'} * \mathcal{A}$.
```

Algorithm 2 Dictionary Training based on Tensor

```
Input: finer granularity tensor $\mathcal{T}^{'}$ and the maximum iterations: num.
Output: Trained dictionary $\mathcal{D}^{'}$.
Initialize $\mathcal{D}^{'}$ with a Gaussian random tensor,
for $t = 1$ to num do
    Fix $\mathcal{D}^{'}$ and solve $\mathcal{A}$ via (4) to (7),
    Fix $\mathcal{A}$ and set $\widehat{\mathcal{T}'} = \text{fft}(\mathcal{T}^{'}, [\,], 3), \widehat{\mathcal{A}} = \text{fft}(\mathcal{A}, [\,], 3)$,
for $k = 1$ to $n_3$ do
    Solving (22) for $\Lambda$ by Newton's method,
    Calculate $\widehat{\mathcal{D}}^{'}$ from (12),
    $\mathcal{D}^{'} = \text{ifft}(\widehat{\mathcal{D}}^{'}, [\,], 3)$.
```

the finer granularity patches can be sparsely represented in an appropriate dictionary and that their sparse representations can be recovered from the coarse granularity observation. The other one is the recovered tensor $\mathcal{T}^{'}$ must be consistent with the input tensor $\mathcal{T}$ for reconstruction. Corresponding to $\mathcal{T}$ and $\mathcal{T}^{'}$, dictionaries are defined $\mathcal{D}$ and $\mathcal{D}^{'}$. In our algorithm, the approach to recover $\mathcal{T}^{'}$ from $\mathcal{T}$ is to keep $\mathcal{D}^{'}$ and $\mathcal{D}$ as a same sparse representation $\mathcal{A}$. To simplify the tensor model, we should minimize the total number of coefficients of sparse representation, which can be represent as $\min \|\mathcal{A}\|_0$, such that $\mathcal{T}^{'} \approx \mathcal{D} * \mathcal{A}$, where higher resolution tensor $\mathcal{T}^{'} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, sparse representation coefficient $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and dictionary $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

Suggested by [10], in order to make $\mathcal{A}$ sparse, we present the optimization equation: equation (2). It can be simplified by Lagrange multiplier λ to equation (3). Here, λ balances sparsity of the solution and fidelity of the approximation to $\mathcal{T}$.

$$\min \ \|\mathcal{A}\|_1 \ \text{ s.t. } \|\mathcal{D} * \mathcal{A} - \mathcal{T}\|_F^2 \leq \epsilon. \tag{2}$$

$$\min_{\mathcal{A}} \ \|\mathcal{D} * \mathcal{A} - \mathcal{T}\|_F^2 + \lambda \|\mathcal{A}\|_1, \tag{3}$$

Recent researches [11][12][13] indicate that there is an intimate connection between sparse coding and neural network. In order to fit in our tensor situation, we propose a back-propagation algorithm called learned iterative shrinkage and thresholding algorithm based on tensor (LISTA-T) to efficiently approximate the sparse representation (sparse coefficient matrix) $\mathcal{A}$ of input $\mathcal{T}$ via using trained dictionary $\mathcal{D}$ to solve equation (2). The entire super-resolution algorithm described above is illustrated in Algorithm 1.

Accompanied with the resolution recovering procedure to obtain pairs of high and coarse granularity tensor patches that have the same sparse representations, we should solve the respected two dictionaries $\mathcal{D}^{'}$ and $\mathcal{D}$. We present an efficient way to learn a compact dictionary by training single dictionary.

Sparse coding is a problem of finding sparse representations of the signals with respect to an overcomplete dictionary $\mathcal{D}$. The dictionary is usually learned from a set of training examples $\mathcal{T}^{'} = \{\mathcal{T}_1^{'}, \mathcal{T}_2^{'}, ..., \mathcal{T}_l^{'}\}$. To optimize coefficient matrix $\mathcal{A}$, we first fix $\mathcal{D}$ and utilize ISTA-T to solve the optimization problem. That is:

$$\min_{\mathcal{A}} f(\mathcal{A}) + \lambda g(\mathcal{A}), \tag{4}$$

where $f(\mathcal{A})$ stands for the data reconstruction term $\|\mathcal{T}^{'} - \mathcal{D}^{'} * \mathcal{A}^{'}\|_F^2$, the Forbenius norm constraints on the term remove the scaling ambiguity, and $g(\mathcal{A})$ stands for the sparsity constraint term $\|\mathcal{A}^{'}\|_1$.

ISTA-T is used to solve this problem, which can be rewritten as a linearized, iterative functional around the previous estimation $\mathcal{A}_p$ with proximal regularization and non-smooth regularization. Thus at the $(p+1)$-th iteration, $\mathcal{A}_{p+1}$ can be update by equation (5). By using L_{p+1} to be Lipschitz constant, and $\nabla f(\mathcal{A})$ to be gradient defined in the tensor space, we can have equation (6)

$$\begin{aligned} \mathcal{A}_{p+1} = \arg\min_{\mathcal{A}} & f(\mathcal{A}_p) + \langle \nabla f(\mathcal{A}_p), \mathcal{A} - \mathcal{A}_p \rangle \\ & + \frac{L_{p+1}}{2} \|\mathcal{A} - \mathcal{A}_p\|_F^2 + \lambda g(\mathcal{A}), \end{aligned} \tag{5}$$

$$\begin{aligned} \mathcal{A}_{p+1} = \arg\min_{\mathcal{A}} \frac{1}{2} \|\mathcal{A} - (\mathcal{A}_p - \frac{1}{L_{p+1}} \nabla f(\mathcal{A}_p))\|_F^2 \\ + \frac{\lambda}{L_{p+1}} \|\mathcal{A}\|_1. \end{aligned} \tag{6}$$

Here, $\nabla f(\mathcal{A}) = \mathcal{D}^\top * \mathcal{D} * \mathcal{A} - \mathcal{D}^\top - \mathcal{T}^{'}$. As we introduced in section II, the transform-based tensor model can also be analyzed in the frequency domain, where the tensor product $\mathcal{T} = \mathcal{D} * \mathcal{B}$. It can be computed as $\widehat{\mathcal{T}}^{(k)} = \widehat{\mathcal{D}}^{(k)} \widehat{\mathcal{B}}^{(k)}$, where $\mathcal{T}^{(k)}$ denotes the k-th element of the expansion of $\mathcal{X}$ in the third dimension and $k \in [n_3]$. For every $\mathcal{A}_{p+1}$ and $\mathcal{A}_p$, we can conduct that:

$$\begin{aligned} & \|\nabla f(\mathcal{A}_{p+1}) - \nabla f(\mathcal{A}_p)\|_F \\ \leq & \sum_{k=1}^{n_3} \|\widehat{\mathcal{D}}^{(k)^\top} \widehat{\mathcal{D}}^{(k)}\|_F^2 \|\mathcal{A}_{p+1} - \mathcal{A}_p\|_F. \end{aligned} \tag{7}$$

Thus Lipschitz constant for $f(\mathcal{A})$ in our algorithm is:

$$L_{p+1} = \sum_{k=1}^{n_3} \|\widehat{\mathcal{D}}^{(k)^\top} \widehat{\mathcal{D}}^{(k)}\|_F^2 \tag{8}$$

Then we can optimized $\mathcal{A}$ by proximal operator $\text{Prox}_{\lambda/L_{p+1}}(\mathcal{A}_p - \frac{1}{L_{p+1}} \nabla f(\mathcal{A}_p))$, where $\text{Prox}_\tau(\cdot) = \text{sign}(\cdot)\text{max}(| \cdot | - \tau, 0)$ is the soft-thresholding operator [8]. After that, we should fix $\mathcal{A}$ to update corresponding

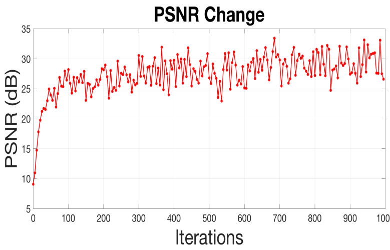

Fig. 7: The PSNR change in training procedure.

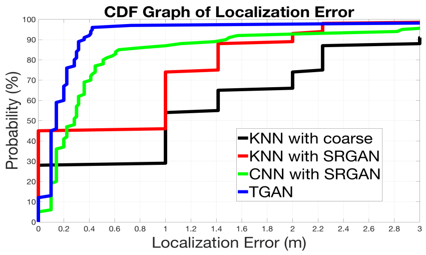

Fig. 8: CDF of location estimation error

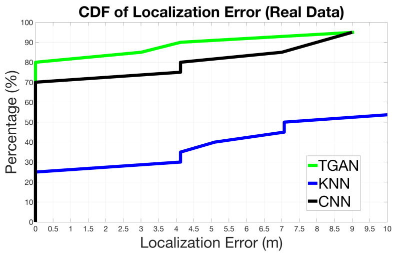

Fig. 9: CDF of localization error of TGAN

dictionary $\mathcal{D}$. We rewrite equation (4) as equation (9). For the reason that, in this paper, we specify the discrete transform of interest to be a discrete Fourier transform, aka., decomposing it into k-nearly-independent problems using DFT like equation (10). Here, $\widehat{\mathcal{T}'}$ denotes the frequency representation of $\mathcal{T}$, i.e. $\widehat{\mathcal{T}'} = \text{fft}(\mathcal{T}, [\,], 3)$.

$$\begin{aligned} \mathcal{D}', \mathcal{A}' = \ & \arg\min_{\mathcal{D}', \mathcal{A}'} \|\mathcal{T}' - \mathcal{D}' * \mathcal{A}'\|_F^2 + \lambda \|\mathcal{A}'\|_1 \\ & \text{s.t. } \|\mathcal{D}_i(:, j, :)\|_F^2 \leq 1, j \in [r], \end{aligned} \tag{9}$$

$$\begin{aligned} & \min_{\widehat{\mathcal{D}}} \ \sum_{k=1}^{n_3} \|\widehat{\mathcal{T}'}^{(k)} - \widehat{\mathcal{D}}^{(k)} \widehat{\mathcal{A}}^{(k)}\|_F^2 \\ & \text{s.t. } \sum_{k=1}^{n_3} \|\widehat{\mathcal{D}}^{(k)}(:, j)\|_F^2 \leq r, j \in [r], \end{aligned} \tag{10}$$

We can apply Lagrange dual with a dual variable $\lambda_j \geq 0, j \in [r]$ to solve equation (10) in frequency domain as equation (11). By defining $\Lambda = \text{diag}(\lambda)$, the minimization over $\widehat{\mathcal{D}}$ can be write as equation (12). Substituting the equation (12) into Lagrangian $\mathcal{L}(\widehat{\mathcal{D}}, \Lambda)$, we will obtain the Lagrange dual function $\mathcal{L}(\Lambda)$ as equation (13). Notation $\text{Tr}(\cdot)$ represents the trace norm of a matrix.

$$\begin{aligned} \mathcal{L}(\widehat{\mathcal{D}}, \Lambda) = & \sum_{k=1}^{n_3} \|\widehat{\mathcal{T}'}^{(k)} - \widehat{\mathcal{D}}^{(k)} \widehat{\mathcal{A}}^{(k)}\|_F^2 \\ & + \sum_{j=1}^{r} \lambda_j \left(\|\widehat{D}(:, j, :)\|_F^2 - r \right), \end{aligned} \tag{11}$$

$$\widehat{\mathcal{D}}^{(k)} = (\widehat{\mathcal{T}'^{(k)}} \widehat{\mathcal{A}}^{(k)\top})(\widehat{\mathcal{A}}^{(k)} \widehat{\mathcal{A}}^{(k)\top} + \Lambda)^{-1}, k \in [n_3]. \tag{12}$$

$$\mathcal{L}(\Lambda) = -\sum_{k=1}^{n_3} \text{Tr}(\widehat{\mathcal{A}}^{(k)\top} \widehat{\mathcal{T}'}^{(k)} \widehat{\mathcal{D}}^{(k)\top}) - k \sum_{j=1}^{r} \lambda_j, \tag{13}$$

Equation (13) can be solved by Newton's method or conjugate gradient. After we get the dual variables Λ, the dictionary can be recovered. The whole training algorithm of dictionary $\mathcal{D}$ is summarized in Algorithm 2.

With the input finer granularity tensor $\mathcal{T}'$, the algorithm can return the trained dictionary $\mathcal{D}'$ and with the input coarse granularity tensor $\mathcal{T}$, the algorithm returns $\mathcal{D}$. So, this is an efficient way to learn a compact dictionary by training single dictionary.

IV. Performance Evaluation

In this section, we first compare the super-resolution tensor results that obtained by applying TGAN on trace- based fingerprint tensor data with other algorithms. Then we evaluate the localization accuracy and compare the performance of our localization model with other previous conventional models using the same data. Finally we implement our own experiment and collect another ground-truth fingerprint data set to verify our approach.

As illustrated in Fig. 5, the trace data set is collected in the region of size 20m × 80m and the tensor set can be measured as $64 \times 256 \times 21$. Each block have 4×4 RPs, we then set some blocks which have 4×4 RPs of coarse granularity tensor $\mathcal{T}'$ contrasted with the rest which have 8×8 RPs of finer granularity tensor $\mathcal{T}$ by $4\times$ up-sampling (The finer granularity of blocks is $0.3\text{m} \times 0.3\text{m}$). 70% blocks enjoy finer granularity, and 30% blocks ready to be processed by TGAN. For this scenario, to measure the quality of generated finer granularity tensor, we adopt the peak signal-to-noise ratio (PSNR) criteria denoted as:

$$\text{PSNR} = 20 \times \log_{10} \frac{\text{MAX}_I}{\text{MSE}}, \tag{14}$$

where $\text{MSE} = \frac{1}{n_1 n_2 n_3} \|\mathcal{T}' - \mathcal{T}\|_F$

There are several existing super-resolution algorithms with our LISTA-T, as suggested in [6][14][11], Trilinear interpolation, Super-resolution GAN (SRGAN) and the Sparse coding based networks (SCN) we adopt. The measurements are shown in Fig. 7 and Fig. 8.

From Fig. 7, we can tell that the calculation of TGAN converge fast. And we measure the Euclidean distance between the estimated location and the actual location of the testing point as localization error by equation (15).

Fig. 8 describes the cumulative distribution function (CDF) of location estimation error of fingerprint classify network with and without original training samples and KNN algorithm.

$$d_e = \sqrt{(\widehat{x} - x)^2 + (\widehat{y} - y)^2} \tag{15}$$

For trace data in localization performance, in Fig. 8, we adopt KNN and SRGAN to compare with the fingerprint-based indoor localization and draw a CDF graph of localization error. For the location estimation with fingerprint data, both neural networks (TGAN and CNN) perform better than KNN with small variation. The error of localization with KNN will increase stage by stage, while the error in neural network increases smoothly. By comparing with the TGAN and CNN supplied by the SRGAN, the improvement shows that the TGAN performs better than the simple merge of generated extra tensor samples and CNN. The TGAN can catch the interconnection inside the data better and generate more similar data for localization.

In our own experiment, we develop an Android application to collect the indoor Wi-Fi RF fingerprint data in a smaller region. With the native Android WifiManager API introduced in Android developer webpage, we can manage all aspects of Wi-Fi connectivity. To be specific, we select a 6m × 16m region with 14 APs lie inside, and we set the scale of the RPs as 1m × 1m, so RSS fingerprint tensor set is of size $6 \times 16 \times 14$. The raw fingerprint data we collect is 4.2MB.

After collecting data, we choose two representative localization techniques to compare with TGAN, namely, weighted KNN and conventional CNN. Weighted KNN is the most widely used technique and is reported to have good performance for indoor localization systems. The conventional CNN in this experiment is similar to the localization part of TGAN, while TGAN holds an extra step for processing collected coarse data to high granularity ones, which is called Super-resolution via Sparse Coding. The trained TGAN downloaded to smartphone is only 26KB in our experiment. The result is shown in Fig.9. By comparing the localization error of KNN algorithm, CNN and TGAN, the terrific result shows that the neural network is much suitable for indoor RSS fingerprint data than KNN, which is consistant with the result of our simulation.

Based on the results of simulation and experiment we made above, we can summarize the improvement of indoor localization in our work. First, the transform-based tensor model can handle insufficient fingerprint samples well by sparse representation, which means we can obtain a finer granularity fingerprint with sparse coding. Secondly, we adopt CNN like localization step and it is proved to be much suitable for analyzing the indoor RSS fingerprint data than KNN, which enables TGAN to give out an improved solution in localization accuracy. Thirdly, TGAN is easy to use and proved to reduce the storage cost of smartphones by downloading the TGAN rather than raw data, so it is acceptable to implement TGAN on smartphones.

V. Conclusion

We first adopted a transform-based tensor model for RF fingerprint and, based on this model, we raised a novel architecture for real-time indoor localization by combining the architecture of sparse coding algorithm called LISTA-T and neural network implementation called TSCN. In order to solve the insufficiency of training samples, we introduced the TGAN to complete super-resolution for generating new fingerprint samples from collected coarse samples. In future works, we will try to figure out a more efficient and slim network architecture for localization process. We believe that applying neural network on mobile platforms is promising.

References

[1] Iris A Junglas and Richard T Watson, "Location-based services," *Communications of the ACM*, vol. 51, no. 3, pp. 65–69, 2008.

[2] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. ACM, 2014, pp. 617–628.

[3] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, Xiaodong Wang, and Min-You Wu, "Adaptive sampling of rf fingerprints for fine-grained indoor localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2411–2423, 2016.

[4] Zheng Yang, Chenshu Wu, and Yunhao Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*. ACM, 2012, pp. 269–280.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[6] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint arXiv:1609.04802*, 2016.

[7] Xiao-Yang Liu and Xiaodong Wang, "Fourth-order tensors with multidimensional discrete transforms," *CoRR*, vol. abs/1705.01576, 2017.

[8] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, Xiaodong Wang, and Min-You Wu, "Tensor completion via adaptive sampling of tensor fibers: Application to efficient indoor rf fingerprinting," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2529–2533.

[9] Xiao-Yang Liu, Xiaodong Wang, Linghe Kong, Meikang Qiu, and Min-You Wu, "An ls-decomposition approach for robust data recovery in wireless sensor networks," *arXiv preprint arXiv:1509.03723*, 2015.

[10] David L Donoho, "For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[11] Zhaowen Wang, Ding Liu, Jianchao Yang, Wei Han, and Thomas Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 370–378.

[12] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun, "Fast inference in sparse coding algorithms with applications to object recognition," *arXiv preprint arXiv:1010.3467*, 2010.

[13] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 399–406.

[14] Wenzhe Shi, Jose Caballero, Lucas Theis, Ferenc Huszar, Andrew Aitken, Christian Ledig, and Zehan Wang, "Is the deconvolution layer the same as a convolutional layer?," *arXiv preprint arXiv:1609.07009*, 2016.