Modality-Agnostic Topology Aware Localization

Anonymous Author(s)
Affiliation
Address
email

Abstract

This work presents a data-driven approach for the localization of an observer on a 2D topological map of the environment. State-of-the-art techniques may yield accurate estimates only when they are tailor-made for a specific data modality that prevents their applicability to broader domains. Here, we establish a model-agnostic framework and formulate the localization problem in the context of parametric manifold learning while leveraging optimal transportation. Our algorithm allows jointly learning a low-dimensional embedding as well as correspondences within a topological map. We examine the generalizability of the proposed algorithm by applying it to data from diverse modalities such as image sequences and radio frequency signals. The experimental results demonstrate decimeter-level accuracy for localization using different sensory inputs.

1 Introduction

Self-localization or localizing objects are primary tasks in navigation and surveillance systems. This problem has been the subject of many studies in the machine learning community. It has been addressed in several areas such as visual odometry [Engel et al., 2017, Brahmbhatt et al., 2018], visual simultaneous mapping and localization (VSLAM) [Davison et al., 2007, Mur-Artal et al., 2015], self-localization [Arth et al., 2011, Sattler et al., 2019, Sarlin et al., 2021], etc., and many approaches have been proposed.

Recent localization methods achieve decimeter precision in the positioning of a moving camera in an indoor environment by leveraging the advances in neural networks [Brahmbhatt et al., 2018, Kendall et al., 2015]. However, such techniques are highly entangled with the modality of data in use; thus, applying such specialized algorithms to other modalities is often not possible. For example, the existing visual odometry and VSLAM techniques that rely on visual features or camera projection models are incompatible with different sensory systems like radio frequency (RF) or audio signals. Still, all instantiate the same problem. In contrast to existing solutions tailored for a particular modality, we formulate the localization problem in terms of low-dimensional manifold learning to represent the input samples in their intrinsic space and transport them to a given topological map by inferring correspondences between intrinsic space and map. This approach allows a broader and more generalizable solution that can be used with data from different modalities.

Here, by localization, we aim to pinpoint the location of a single observer at each time instance on a topological map by analyzing the collected sensory data. Unlike VSLAM, our method does not build a map; we assume that it is given as a prior. Nevertheless, the topological map can have different forms. For example, it can be an approximate sketch of the 2D floorplan or a precise 2D CAD map. While the observer moves in the environment and visits different locations, the measured signals such as video frames, depth images, or WiFi channel states encode the 2D location of the observer as well as the geometry of the space and the modality-specific factors, depending on the type of the sensory system in its high-dimensional ambient space. However, the *intrinsic* manifold that encodes the positional information (coordinates) lies only either in a 2D or 3D space (3D in case the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

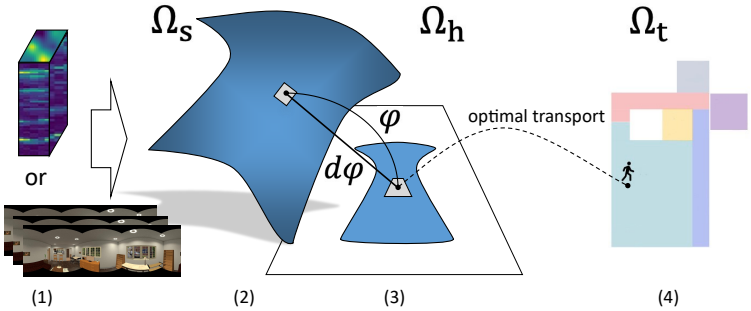

Figure 1: Schematic of the proposed modality-agnostic localization method; (1) different input modalities, for instance, radio frequency (RF) signals or video sequences; (2) the input signals are assumed to change gradually, thus are on a smooth manifold in input space Ω_s; (3) the positions are encoded in an intrinsic space Ω_h. Please refer to Sec. 3.3 for details about the mapping; (4) correspondences between Ω_h and the positions on the floormap is estimated by means of optimal transportation. Please refer to Sec. 3.2 for details.

altitude of the observer varies). Hence, a dimensionality-reducing nonlinear transformation between the input sample and their intrinsic 2D/3D embedding yields a solution to the localization problem. Incorporating domain knowledge (camera projection models for visual sensing or wave propagation equations for RF sensing) may reduce the task of finding the transformation to a parametric regression problem; however, the obtained solution would again be modality-specific. Instead, we build on our intuition that the transformation from high-dimensional input to the intrinsic space can be modeled as a parametric map learnable through a neural network between data manifold and coordinates.

In addition, we observe that the temporal data only changes gradually as the observer moves; for instance, consecutive images in a video stream are similar. This observation implies the data reside in a smooth, locally connected manifold. We further postulate that an embedded latent manifold of much lower dimensionality represents the locations of the observer in the environment. Based on this, we revisit the localization problem and reformulate it as a learning task of the embedding space of data. Since we are given a topological map yet not the correspondences between the embedding space and the map, we adopt optimal transportation (OT) to estimate them.

The proposed method does not rely on modality-specific priors, instead it builds on correlation between the observer location and the measured signal. Since we make no assumption on the transformation, the proposed method is applicable to a large spectrum of different sensor modalities. Figure (1) illustrates the proposed approach.

2 Related Work

Localization has been the subject of many studies in robotics, computer vision, and RF sensing. As such, classical VSLAM methods [Davison et al., 2007, Mur-Artal et al., 2015] mainly build a 3D point cloud map of the scene while estimating the motion of the camera in the video data. Modern VSLAM approaches are usually composed of several carefully engineered modules, such as tracking, mapping, re-localization, loop closure, graph optimization, key-frame selection, or post-processing by bundle adjustment [Henriques and Vedaldi, 2018]. Many of these modules make strong assumptions on the visual features or a camera projection model. Even recent efforts to replace some of these modules with learnable architectures borrow those assumptions in their methodology [Chen et al., 2020]. Employing visual features and camera projection models can also be seen in visual odometry [Engel et al., 2017]. While these advances report decimeter-level accuracy in indoor visual localization, they are not applicable to other sensory systems such as RF or audio signals. Their success demands major engineering efforts in transferring similar ideas to a new data modality.

Manifold learning is being used extensively in fields like computer vision, natural language processing, and data mining. In manifold learning, the data points are assumed to lie on a smooth manifold $\mathcal{M} \subset \mathbb{R}^n$ in n-dimensional measured ambient space, while they are sampled from a distribution, near or on lower-dimensional sub-manifold $\mathcal{N} \subset \mathbb{R}^s$, where $n > s$. The minimum number of variables

needed to describe such a distribution is known as the intrinsic dimension. The task of manifold learning is to find a smooth map $\Phi : \mathcal{M} \rightarrow \mathcal{N}$. If the data has intrinsic dimensionality of m, according to the Whitney Embedding Theorem [Lee, 2013], we know that $\mathcal{M}$ can be embedded smoothly into $s = 2m$, using a homeomorphism Φ. However, it is often impossible to obtain an isometric embedding directly in the intrinsic space [McQueen et al., 2016]. As far as the smooth embedding preserves the topology of $\mathcal{M}$, this might be sufficient for many dimension reduction purposes but when preserving the geometry of embedding is desirable, finding an isometric embedding is required. Several previous works in non-linear dimension reduction [Tenenbaum et al., 2000, Verma, 2012, Weinberger and Saul, 2006, McQueen et al., 2016] have been driven by the desire to find a smooth embedding of low dimension that is isometric. Since almost all embedding algorithms suffer from distortion, there are several works that propose a different technique for mitigating this issue by filtering out the border samples [Rosman et al., 2010], introducing a Riemannian relaxation [McQueen et al., 2016], or setting some constraints on the embedding [Agrawal et al., 2021]. Depending on the amount of imposed distortion in the intrinsic embedding, the performance of a downstream task that is applied on the representation of the sample in the embedding space would degrade. In localization problems, when there is no ground truth positions available, mapping from a distorted intrinsic space into the target map can be difficult and prone to a high error.

Optimal transportation has received considerable interest from the topology community [Peyré et al., 2019]. When the data is associated with geometrical properties, optimal transport metrics (also called Wasserstein distance or Earth Mover distance) measure the spatial variations between probability distributions of source and target domains. Correspondence matching is one of the successful applications of optimal transport [Mémoli, 2011, Peyré et al., 2016, Solomon et al., 2016]. Given a transport cost function, the Wasserstein distance computes the optimal transportation plan between two measures. Recent progress on efficient computation of the optimal transport by introducing entropy regularization and Sinkhorn's matrix scaling algorithms reduced the computational cost of optimal transport several orders of magnitude than of the original transport solver [Cuturi, 2013]. As shown in [Genevay et al., 2018], computing the OT loss and its gradient can be tractable by using Sinkhorn fixed point iterations.

Finding the transformation for representing the data points on a given topological 2D map requires knowing a set of correspondences in the localization problem. In case the correspondences are available, learning a transformation is straightforward. However, in an unsupervised setting, estimating such a transformation can be very difficult.

In what we propose, the correspondences (i.e. coupling matrix) between the 2D embedding and the target topological map is learned by using the OT algorithm. We find a coupling matrix (also called transport plan) representing the correspondences between two domains by minimizing the cost of transportation. In the next section, we explain how the transportation cost is parameterized by the representation of a neural network in the intrinsic embedding of data.

3 Embedding learning and transportation to the intrinsic space

In this work, we formulate the localization problem in the context of manifold learning and optimal transportation. Our contributions are as follows:

1. We propose a model that jointly learns the intrinsic embedding and its transportation into a topological map of the environment in a weakly-supervised style by using gradient descent optimization. Such a joint optimization mitigates the distortion of the intrinsic embedding as the model constrains it to resemble the topology of the target map.
2. The proposed localization method does not make any assumptions about the data modality in use. In that sense, it is modality-agnostic and can be applied to a large family of sensory systems. From the system setup point of view, our method can extend to both active and passive positioning problems.

3.1 Problem and theoretical motivations

Let $\Omega_s \in \mathbb{R}^n$ be the ambient space of some measured signal and $\Omega_h \in \mathbb{R}^m$ with $m \subset \{2, 3\}$ be its intrinsic space. We are interested in finding the representations in Ω_h of the discrete samples

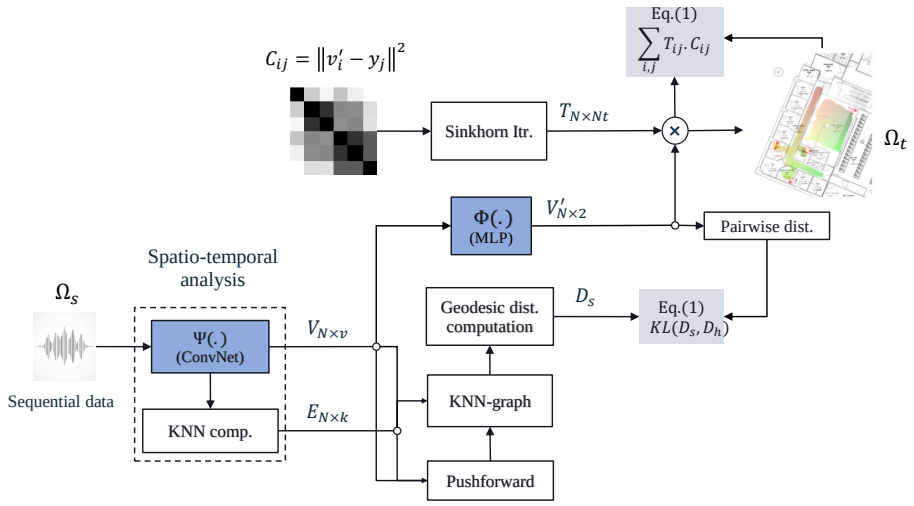

Figure 2: Block diagram of proposed method. A set of prototype vectors and their KNN are computed by spatio-temporal analysis of the sequential input data. The pairwise distances between the KNN samples in the embedding are estimated and used for jointly learning the embedding and the correspondences with a given target topological map.

$X_s = \{x_i^s\}_{i=1}^{N_s} \in \Omega_s$. We assume the measured samples X_s lie on a smooth (Riemannian) manifold in Ω_s; thus, the manifold is locally connected. This assumption intuitively holds since the data is a temporal sequence that does not change much between two consecutive instances in time. We also assume that the topological map $\Omega_t \in \mathbb{R}^m$ that represents the geometry of the environment is known, yet no correspondences between these two spaces are available. This map can be in the form of a 2D sketch or a floorplan of a building. Localizing the observer in Ω_t requires finding a mapping between Ω_s and Ω_t.

Using manifold learning techniques, one can estimate the embedding in $\mathbb{R}^m$ and consequently seek a transformation to map the embedding into Ω_t. When we do not have access to correspondences between the embedding and target domains, finding an isometric embedding that preserves the pairwise distances between samples is desirable as the transformation to the target would be less complex. Hence, methods like Isomap [Tenenbaum et al., 2000] which preserve the local and global distances, have an advantage. However, due to the non-convexity of Ω_s, it is often impossible to obtain an isometric embedding directly in the intrinsic space [McQueen et al., 2016]. Since walls usually partition interior spaces, there may not be a direct path between any two arbitrary points on the floorplan. Therefore, due to the non-convexity of the sample set, usually, the embedding suffers from a severe distortion that a simple isometric transformation is not sufficient for alignment with the target domain. As an alternative, we propose finding a map between the input and the intrinsic spaces that both preserves the pairwise distances between samples and aligns Ω_h and Ω_t.

We formulate the localization problem as follows. Consider $\Phi : \Omega_s \to \Omega_h$ be a smooth map between input and intrinsic embedding. The map Φ can be implemented by a neural network e.g. an MLP. We define $D_s \in \mathbb{R}^{N_s \times N_s}$ and $D_h \in \mathbb{R}^{N_t \times N_t}$ as distance matrices between samples in Ω_s and Ω_h, respectively. So the entries of D_h can be computed like $d_{ij}^2 = \|\Phi(x_i) - \Phi(x_j)\|^2$. For now, let's assume we have access to the geodesic distance-matrix D_s, which contains all pairwise geodesic distances of X_s on the input manifold. Later, in section 3.3, we explain how D_s can be approximated. It is worth mentioning that training the Φ by minimizing the $\|D_s - D_h\|^2$ and using gradient descent optimization leads to a parametric approximation of the MDS algorithm [Pai et al., 2019]. However, this formulation is ill-posed when X_s is a non-convex set since comparing the geodesic distances with Euclidean distances by using $\|\cdot\|^2$ is only valid inside a convex region.

For solving the correspondence problem, the Gromov-Wasserstein discrepancy between two distance matrices was used [Peyré et al., 2016]. There, the coupling between two sets of distances is found

by a regularized optimal transport. In this work, our proposed model learns *both* the map Φ *and* simultaneously by estimating a coupling matrix ($T \in \mathbb{R}^{N_s \times N_t}$). Training the model therefore amounts to minimizing the loss

$$\min_{\Phi,T} L(D_s, D_h) + \mu \sum_{ij} T_{ij} \cdot C_{ij} \quad \text{with} \quad C_{ij} = \|\Phi(x_i) - y_j\|^2, \tag{1}$$

where the first term $L(.,.)$ is a dissimilarity measure between two distance matrices and the second term is the Sinkhorn distance between the samples in the embedding and the target topological map. The scalar μ is a contribution weight. The $C \in \mathbb{R}^{Ns \times Nt}$ and $T \in \mathbb{R}^{Ns \times Nt}$ are the cost of transportation and the coupling matrix. Choosing Eucidean distances for L is ill-posed since D_s contains geodesic distances on the input manifold and the matrix D_h contains $L2$ distances between samples on the embedding. We propose to use the Kullback-Leibler divergence (KL) for L [Cuturi, 2013]. Figure (2) shows the block diagram of the proposed model.

3.2 Regularized Transport with Differentiable Sinkhorn Distance

One major advantage of regularizing the optimal transport problem is that it becomes solvable efficiently using Sinkhorn's algorithm [Zhou et al., 2018]. In computing the entropy constraint of the Sinkhorn distance as proposed in [Cuturi, 2013], we are looking for a coupling matrix T that satisfies

$$T(C, p, q) = \underset{T \in \gamma(p,q)}{\text{argmin}} \langle T, C \rangle - \frac{1}{\lambda} H(T). \tag{2}$$

Here, p and q are probability distributions of samples in source Ω_s and target Ω_t domains and $\gamma(p, q)$ is their joint probability. The $C \in \mathbb{R}^{Ns \times Nt}$ is cost matrix for transporting mass between two domains. In Eq. (2), the term $H(T) = -\sum T \log(T)$ is the entropy of coupling T. The solution of (2) is [Cuturi, 2013]:

$$T(C, p, q) = \text{diag}(a) K \text{diag}(b), \tag{3}$$

where $K = e^{-\lambda C} \in {\mathbb{R}_+}^{Ns \times Nt}$ is the so-called Gibbs kernel associated to C and $(a, b) \in {\mathbb{R}_+}^{N_s} \times {\mathbb{R}_+}^{N_t}$ can be computed using the Sinkhorn-Knopp iterative algorithm:

$$a \leftarrow \frac{p}{Kb} \quad \text{and} \quad b \leftarrow \frac{q}{K^\top a}, \tag{4}$$

where $\top$ denotes the transpose of a matrix and the division is element-wise. When there is no prior knowledge about the location of the observer in the environment, a uniform distribution can be assigned to p and q. As proposed in [Zhou et al., 2018], using the Sinkhorn iteration is an efficient way for differentiating Eq. (2) [Zhou et al., 2018]. By running Eq. (4) for M iteration, and putting the obtained (a, b) in Eq. (3), we can compute the derivative $\frac{\partial T}{\partial K}$ and consequently $\frac{\partial T}{\partial C}$. Hence, Eq. (1) is differentiable. As the cost matrix C depends on Φ, the gradient can be back-propagated to optimize the parameters of the neural network Φ. The pseudo code in Algorithm (1) describes the procedure of jointly learning the embedding and the transportation.

The value of the regularization coefficient ($\frac{1}{\lambda}$) in Eq. (2) controls the distribution of T. A high entropy on the coupling matrix T means a uniform distribution over its entries that makes it a smooth assignment matrix. In contrast, a low entropy means the T makes a hard assignment. Requlariziing the Eq. (2) with high entropy makes the optimization stable [Zhou et al., 2018], at the cost of a higher error in finding the correct correspondences. When training the model, we apply a linear annealing

schedule to the weight of the entropy regularization from an initial high value to a final small value. This range of λ is a hyper-parameter of the model and is fixed in all our experiments.

Algorithm 1: Jointly learning the embedding and the transportation plan.

Input : $X_s \in \Omega_s$, $Y \in \Omega_t$, and $D_s \in \mathbb{R}^{N_s \times N_s}$
Output : model parameters Φ and coupling matrix $T \in \mathbb{R}^{N_s \times N_t}$
Initialize: Φ, T, and $\lambda = \{\lambda^{(0)}, ..., \lambda^{(max)}\}$;
while $t < Max_{itr}$ **do**
  $\hat{D}_{ij} = \|\Phi(x_i) - \Phi(x_j)\|^2$;
  $C_{ij} = \|\Phi(x_i) - y_j\|^2$;
  $T = Sinkhorn(C, \lambda^{(t)})$ `// Eq. (3) and Eq. (4)`
  $\mathcal{L} = D_{KL}(D_s \| \hat{D}) + \mu \cdot \langle T, C \rangle$ `// Eq. (1)`
  $\Phi^{(t+1)} \leftarrow \Phi^{(t)} + \eta \frac{\partial \mathcal{L}}{\partial \Phi}$ `//` η `is learning rate`
end

3.3 Computing the geodesic distance-matrix

The objective function in Eq. (1), requires pre-computing the distance matrix D_s that represents the pair-wise distances in the training set X_s. Since the inputs X_s are distributed on a non-convex manifold in input space, the pairwise geodesic distances between non-neighboring samples should be measured. For an efficient computation of the geodesic distances, three steps are taken: (1) Reducing the size of X_s by finding a set of representative samples (also called prototype vectors or landmarks) and computing their k-nearest neighbors; (2) estimating the push-forward metric for estimating the Euclidean distances between neighbor prototypes in the embedding; (3) estimating the pairwise geodesic distances between non-neighbor prototypes, using a shortest path algorithm; In the following, we explain each step.

3.3.1 Finding a set of prototypes and their KNN

Similar to landmark Isomap [De Silva and Tenenbaum, 2003], instead of computing all pairwise distances in a large training set of X_s, only the pairwise distances in a set of N_s representative samples – called prototype vectors – are computed. Finding a set of prototypes by using farthest-point sampling [Pai et al., 2019] and consequently measuring the Euclidean distance between them in the high-dimensional input space have some drawbacks. Even for the 2D manifolds in $\mathbb{R}^3$, such as surfaces with holes or self-intersection, finding the KNN, by using Euclidean distance is susceptible to short-circuiting on the manifold.

Instead, we propose to learn a metric space for computing both the prototype vectors and their neighbor indices by training a neural network on the triplet sampled data [Hadsell et al., 2006]. Each triplet set contains two samples that are temporally close, and a third one that is distant. We apply an upper bound to the maximal temporal distance. The network learns to produce similar feature vectors for samples, based on their temporal vicinity by minimizing its triplet margin loss:

$$\mathcal{L}(x_i^a, x_i^p, x_i^n) = max\Big(0, d(h_i^a, h_i^p) - d(h_i^a, h_i^n) + \alpha\Big) \quad \text{where} \quad h_i = \Psi(x_i). \tag{5}$$

In Eq.(5), the symbol Ψ denotes the function of the neural network, d measures the Euclidean distances, and $(h_i^a, h_i^p, h_i^n) \in \mathbb{R}^\nu$, $\nu < n$, are the output vectors of the network, produced from the ith set of anchor (x_i^a), positive (x_i^p) and negative (x_i^n) instances and the scalar α is a constant margin.

After convergence, the data samples (X_s) are partitioned into N_s clusters by applying a standard kmeans clustering on the obtained features set h. As prototypes, we use the centroid vectors of each class. Furthermore, by measuring the pairwise Euclidean distances between the feature set, the K nearest neighbors of each prototype are found.

3.3.2 Approximating the distances in embedding

For estimating the small distances between neighbor samples in the embedding, measuring the Euclidean distance in the input space is not informative [Dsilva et al., 2015]. For more reliable estimation, the unimportant sources of variability need to be suppressed. As in [Dsilva et al., 2015],

the local Mahalanobis distances can be used as an approximation of small Euclidean distances in the embedding:

$$\|\Phi(x_i) - \Phi(x_j)\|^2 \approx \frac{1}{2}[x_j - x_i]^\top \cdot [C^\dagger(x_i) + C^\dagger(x_j)] \cdot [x_j - x_i], \tag{6}$$

where $C(x_i)$ is the measured local covariance matrix of data at the location of sample x_i, and $\dagger$ denotes the Moore-Penrose pseudoinverse. For computing Eq. (6), we estimate the covariance matrix of samples in each cluster. By using Eq. (6), the pairwise distances between neighboring prototype vectors in the embedding space can be approximated. When the distances are small enough, such an estimation is similar to the *pushforward* in differential geometry that is an approximation to the differential of map ($\partial\Phi$) between tangent spaces of two manifolds.

After computing all pairwise distances between each prototype and its KNN, we create a KNN-graph similar to the procedure in Isomap [De Silva and Tenenbaum, 2003] and the distances between non-neighboring samples are estimated by using the Dijkstra's shortest path algorithm [Dijkstra et al., 1959]. Then, the geodesic matrix D_s in the embedding space is known and Eq. (1) can be evaluated for training the model.

4 Experiments

We applied the proposed model on three different data modalities. For details about the implementation, hyper-parameters and training please refer to the supplementary material.

4.1 Synthetic data - 2D maze environment

To validate our approach without the added complexities of data, we first setup an experiment with a simple 2D Maze environment[1] [Henriques and Vedaldi, 2018, Parisotto and Salakhutdinov, 2017]. In this simulation, an agent walks through a maze from a starting point to the end point. The starting and end points are chosen randomly on the outer walls of the maze (see Figure (3)). The solution path between these two points is considered as a trajectory. We create many trajectories by randomly choosing several different starting and end points. A square window, centered at the location of agent is used for sampling a image patch from the environment. A collection of sampled patches is used as the training set. The model predicts the position of observer with 0.7 pixel error for the maze environment of size 24×24 pixels. More inference results for number of trajectories are shown in the supplementary (Figure (**??**)).

Figure 3: Synthetic maze environment of size 24×24 pixels; (left) An example of a path and the sampling window; (middle) extracted patches for the shown path; (right) an example of the ground truth (in green) and the predicted (in red) trajectories

4.2 Camera-based localization

We setup an experiment by using the *iGibson* dataset [Shen et al., 2020]. It consists of 15 high-quality 3D scenes which can be explored interactively [Xia et al., 2020]. The scenes are created from real homes and offices. The iGibson allows us to create a set of trajectories in each environment that simulate a moving agent equipped with a camera in a building. We create a video from each

[1] https://github.com/theJollySin/mazelib (GNU General Public License v3.0)

Figure 4: An example of the iGibson environment; (left) the quadruple-view at the location of observer and the constructed panorama image; (middle) a ground truth trajectory; (right) prediction

	Environments														
index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ϵ	0.62	1.05	0.73	0.64	0.69	0.83	1.02	1.03	0.73	0.62	0.74	0.71	0.98	1.04	1.14

Table 1: First row: indexes of the 15 scenes of iGibson (*Rs*, *Merom-0*, *Merom-1*, *Pomaria-0*, *Pomaria-1*, *Pomaria-2*, *Ihlen-0*, *Ihlen-1*, *Benevolence-0*, *Benevolence-1*, *Benevolence-2*, *Beechwood-0*, *Beechwood-1*, *Wainscott-0*, *Wainscott-1*). Second row: Localization errors (in meters).

environment by traversing all rooms/zones. We exclude some of the images from training. At each location of agent, we capture four orthogonal-view images by rotating the camera with multiple of 90° with the up-direction being the rotation axis. These four images are used for constructing a panorama image. The sampled images and the bird-eye view of floor plan are the inputs to the model. Figure 4 shows an example of four images, taken at the location of observer, a ground-truth trajectory, and its prediction. Table 1 reports the localization errors for different environments in the dataset.

4.3 Passive WiFi localization

Localization of a moving target in the pervasive WiFi signal has received some attention of the machine learning community [Fan et al., 2020]. We consider *passive* positioning, i.e. the only source of information are the reflections of the electromagnetic waves from the body of the moving target. We collected a data set by using four commercial IEEE 802.11 access points (AP), operating in the 5GHz band, and deployed them in the test environment of size 16×20 meters in a building. The environment contains three rooms, two long aisles and a large lab (see Figure 5). We use three receivers with eight antennas each, and a transmitter with a single antenna. Each receiver collects the *Channel State Information* (CSI) at periodic intervals. The CSI represents the channel between the transmitter antenna and each of its 8 receiver antennas, across 208 frequency tones that span the transmission bandwidth. Hence, the CSI is represented as a multidimensional tensor of complex numbers of dimension 8x1x208 per each WiFi packet. Here, we only use the magnitude of CSI signal. We recorded about five minutes of CSI data from all three receivers, while a person freely walks through different locations in the environment. For ground-truth positions, we used several 3D cameras, installed in the environment. The algorithm achieved an average error of 1.2 meters for locating the person on the 2D floor-plan image of the environment. Figure 5 visualizes the predicted locations of the person in the building.

4.4 Ablation Studies

We did an ablation study on the robustness of the camera-based localization by altering the visual features and object instances in the environment. This has been done by randomly exchanging both texture and appearance of objects in the scene. Figure 6 shows several examples of such variations with respect to the original environment that has been used for training. By measuring the accuracy the algorithm achieves on the altered scenes in a validation run, we are able to determine how susceptible the algorithm is when it is tested on the perturbed images. To estimate mean and standard deviation, we repeated this test ten times with ten differently randomized scenes. The results can be seen in table 2.

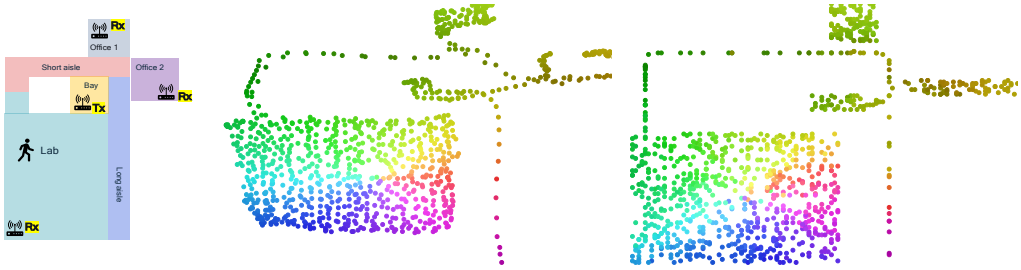

Figure 5: WiFi localization; (left) floorplan topological map; (middle) ground-truth positions of observer; (right) inference. The points color encodes the x-y locations of person.

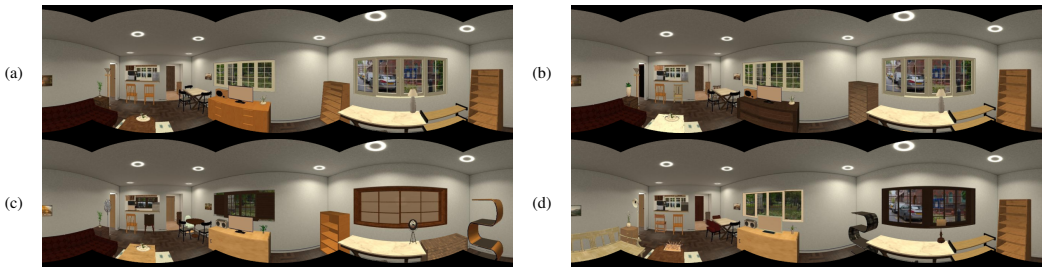

Figure 6: Example of perturbation in the environment; (a) shows a reference environment that has been used for the training; (b-d) show panoramas which correspond to 70%, 50% and 30% of the objects identical with those in (a).

	Perturbation rate					
Rate	0%	10%	20%	30%	50%	70%
Mean of error	61.9	61.87 ± 0.09	62.2 ± 0.09	62.82 ± 0.17	63.77 ± 0.14	64.07 ± 0.26

Table 2: Mean and standard deviation of localization error (cm) under different amount of changes in the environment.

5 Discussion

The localization problem includes a large group of systems and methods that by using a sensory system and a computational algorithm aims to find the location of observer on a given topological map. Depending on the type of sensory system, several approaches have been presented in the literature. However, the existing solutions, even the data-driven methods, mostly work for a specific modality. This is mainly because these algorithms are built based on some strong assumptions about the characteristic of input data like image features or sensory system like the projection model of the camera. These assumptions can easily be violated if the algorithm applies on a different data modality like RF signals or vice versa. In this work, we formulated the localization problem in the context of manifold learning by finding the 2D intrinsic space of data that leads to a more generic solution for a large family of localization problems. Finding an isometric embedding for the realistic noisy data under limited amount of samples, is highly erroneous and the computed embedding usually suffers from a severe distortion. Such a distortion imposes difficulty in finding a precise transformation to the topological map in an weakly-supervised style. In this work, we proposed to learn the embedding and the transformation to the target topological map jointly by gradient descent optimization. Since the exact position of target on the topological map is unknown for the model, we employed optimal transport for finding the correspondence (i.e. coupling) between the intrinsic embedding and the target geometry. Our experiments on both synthetic and realistic data with different modalities show the effectiveness of the proposed method.

References

A. Agrawal, A. Ali, and S. Boyd. Minimum-distortion embedding. *arXiv preprint arXiv:2103.02559*, 2021.

C. Arth, M. Klopschitz, G. Reitmayr, and D. Schmalstieg. Real-time self-localization from panoramic images on mobile devices. In *2011 10th ieee international symposium on mixed and augmented reality*, pages 37–46. IEEE, 2011.

S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2018.

C. Chen, B. Wang, C. X. Lu, N. Trigoni, and A. Markham. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567*, 2020.

M. Cuturi. Lightspeed computation of optimal transportation distances. *Advances in Neural Information Processing Systems*, 26, 2013.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007.

V. De Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, pages 721–728, 2003.

E. W. Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1):269–271, 1959.

C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman, and I. G. Kevrekidis. Data-driven reduction for multiscale stochastic dynamical systems. *arXiv preprint arXiv:1501.05195*, 2015.

J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

L. Fan, T. Li, Y. Yuan, and D. Katabi. In-home daily-life captioning using radio signals. In *European Conference on Computer Vision (ECCV)*, 2020.

A. Genevay, G. Peyré, and M. Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR, 2018.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

J. F. Henriques and A. Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018.

A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.

J. M. Lee. Smooth manifolds. In *Introduction to Smooth Manifolds*, pages 1–31. Springer, 2013.

J. McQueen, M. Meila, and D. Joncas. Nearly isometric embedding by relaxation. *Advances in Neural Information Processing Systems*, 29:2631–2639, 2016.

F. Mémoli. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

G. Pai, R. Talmon, A. Bronstein, and R. Kimmel. Dimal: Deep isometric manifold learning using sparse geodesic sampling. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 819–828. IEEE, 2019.

E. Parisotto and R. Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360*, 2017.

G. Peyré, M. Cuturi, and J. Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In *International Conference on Machine Learning*, pages 2664–2672. PMLR, 2016.

G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

G. Rosman, M. M. Bronstein, A. M. Bronstein, and R. Kimmel. Nonlinear dimensionality reduction by topologically constrained isometric embedding. *International Journal of Computer Vision*, 89 (1):56–68, 2010.

P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, et al. Back to the feature: Learning robust camera localization from pixels to pose. *arXiv preprint arXiv:2103.09213*, 2021.

T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe. Understanding the limitations of cnn-based absolute camera pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3302–3312, 2019.

B. Shen, F. Xia, C. Li, R. Martín-Martín*, L. Fan, G. Wang, S. Buch, C. D'Arpino, S. Srivastava, L. P. Tchapmi, K. Vainio, L. Fei-Fei, and S. Savarese. igibson, a simulation environment for interactive tasks in large realistic scenes. *arXiv preprint arXiv:2012.02924*, 2020.

J. Solomon, G. Peyré, V. G. Kim, and S. Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG)*, 35(4):1–13, 2016.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

N. Verma. Distance preserving embeddings for general n-dimensional manifolds. In *Conference on Learning Theory*, pages 32–1. JMLR Workshop and Conference Proceedings, 2012.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1):77–90, 2006.

F. Xia, W. B. Shen, C. Li, P. Kasimbeg, M. E. Tchapmi, A. Toshev, R. Martín-Martín, and S. Savarese. Interactive gibson benchmark: A benchmark for interactive navigation in cluttered environments. *IEEE Robotics and Automation Letters*, 5(2):713–720, 2020.

H. Zhou, B. Ummenhofer, and T. Brox. Learning generative models with sinkhorn divergences. In *European Conference on Computer Vision (ECCV)*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] In section (3), we explained in details how the claims can be justified. Furthermore, we valid our claims by several experiments.
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]

(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Two out of three datasets that have been used are publicly available. The third dataset was collected internally and the authors don't have permission from the owner to release it. Regarding the code, we didn't release the programming code but a pseudo code was provided for the core of the proposed method. In supplementary, the instructions for reproducing the experiments were given.

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We includes them in the supplementary materials

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We reported the statistics (mean and standard deviation) of the obtained results in Table 2.

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We mentioned type of resources and the training time on our GPU device in the supplementary materials.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] we cited the iGibson dataset and its interactive assets. We also mentioned the source code of generating the Maze environment.

   (b) Did you mention the license of the assets? [Yes] we added the license to the library for generating Maze data in the footnote of the experimental section. The iGibson dataset was cited and its license is accessible in the dataset website.

   (c) Did you include any new assets either in the supplemental material or as a URL? [No]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]