Online Domain Adaptation for Occupancy Mapping

Anthony Tompkins*
School of Computer Science
The University of Sydney
NSW, Australia.
anthony.tompkins@sydney.edu.au

Ransalu Senanayake*
Department of Aeronautics
and Astronautics
Stanford University, CA, USA
ransalu@stanford.edu

Fabio Ramos
School of Computer Science
The University of Sydney, Australia.
NVIDIA Research, Seattle, WA, USA.
fabio.ramos@sydney.edu.au

Abstract—Creating accurate spatial representations that take into account uncertainty is critical for autonomous robots to safely navigate in unstructured environments. Although recent LIDAR based mapping techniques can produce robust occupancy maps, learning the parameters of such models demand considerable computational time, discouraging them from being used in real-time and large-scale applications such as autonomous driving. Recognizing the fact that real-world structures exhibit similar geometric features across a variety of urban environments, in this paper, we argue that it is redundant to learn all geometry dependent parameters from scratch. Instead, we propose a theoretical framework building upon the theory of *optimal transport* to adapt model parameters to account for changes in the environment, significantly amortizing the training cost. Further, with the use of high-fidelity driving simulators and real-world datasets, we demonstrate how parameters of 2D and 3D occupancy maps can be automatically adapted to accord with local spatial changes. We validate various domain adaptation paradigms through a series of experiments, ranging from inter-domain feature transfer to simulation-to-real-world feature transfer. Experiments verified the possibility of estimating parameters with a negligible computational and memory cost, enabling large-scale probabilistic mapping in urban environments.

I. INTRODUCTION

The demand for intelligent robots in day-to-day activities is growing as never before. However, one of the main reasons hindering the deployment of robots in real-world environments is the challenge of reliably adapting to continuously changing environments. Since a robot typically represents its environment and itself using mathematical models, it is indispensable to adjust these models to accommodate changes in the environment the robot operates in. For instance, if the model is represented as a parameterized statistical model, its parameters should be regularly redetermined to adjust for changes to new environments and data.

If the learning procedure is computationally expensive, frequently updating the model parameters in real-time is a significant challenge. This is indeed the case in deep learning as well as in many Bayesian inference techniques. While there are many methods to adapt deep neural networks to varying domains [1, 2, 3, 4, 5], such adaptation techniques are under-explored for Bayesian models [6] despite their extensive applications in robotics [7, 8, 9, 10, 11]. As uncertainty is represented as probability distributions in Bayesian models, entire distributions need to be adapted when changing to a new domain. The question remains: how do we solve the problem of efficient adaptation without retraining models from scratch? In this paper, we focus on learning the uncertainty of occupancy in an unknown environment by transferring model parameters associated with a source dataset to a target dataset in a zero-shot fashion [12]. This transfer procedure significantly reduces the time to estimate the model parameters, as opposed to learning them from scratch.

Even though the fundamental techniques developed in this paper have great potential to be used in a variety of data-efficient robot perception and planning applications, our focus is to build an online continuous mapping method for arbi-
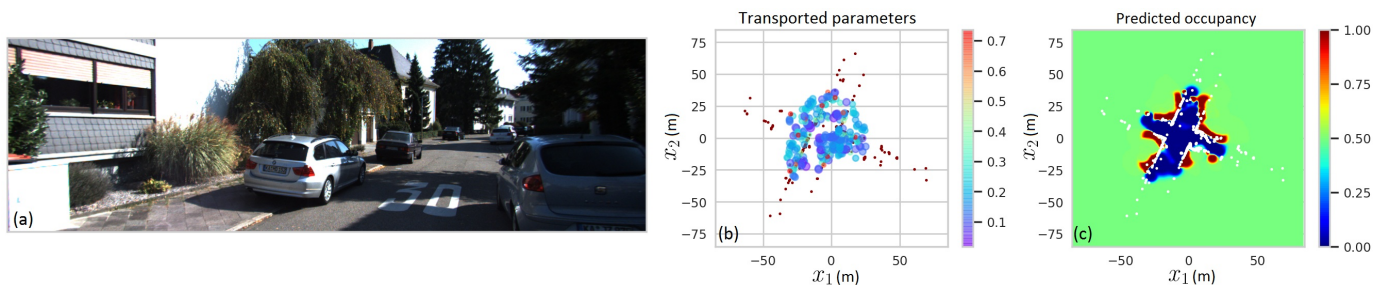
Fig. 1: (a) Forward camera-view from a car that has just passed an urban intersection (KITTI dataset). (b) A set of occupancy model parameters estimated using the proposed Parameter Optimal Transport (POT) method. Values of these parameters depend on the geometry of the environment. Note that these parameters were *transferred online* from a simulated environment and were *never learned from scratch*. (c) Mean occupancy map obtained from the transferred parameters.

trarily large environments. Our formulation builds upon the state-of-the-art Bayesian occupancy mapping technique named automorphing Bayesian Hilbert maps (ABHMs) [13]. By developing a novel parameter transfer learning technique, we make this theoretically rich, yet practically less scalable offline mapping technique, run online in large-scale unknown urban environments. Since ABHM explicitly provides uncertainty estimates of which areas of the environment are occupied, it can be utilized in safety-critical robotics applications [14] such as autonomous driving. For instance, they can be integrated into safe-motion planning algorithms and risk-aware decision-making in cluttered and dynamic real-world urban environments [15, 16]. The main reason that hinders the use of ABHM in real-world applications is the run-time cost of learning parameters as it relies on an expensive black-box variational inference technique. Because these parameters are spatially local and depend on the geometrical features of the objects in the environment, parameters in one location of the environment are completely different from another. Therefore, ABHM requires learning these spatially variant parameters for every location of the environment. Moreover, in dynamic environments, these parameters need to be swiftly adjusted to the changing occupancy level. Taking into account these limitations, it is essential to quickly estimate the parameters in an alternative and more efficient manner.

As an alternative to relearning parameters in a new scene, we propose to transfer "geometry-dependent spatial features" of the ABHM model from a training data pool to the current scene. We show that this can be efficiently done using the theory of Optimal Transport [17], which recently regained popularity due to its successful application to several machine learning algorithms [18, 19]. The proposed approach completely bypasses explicitly learning parameters of the statistical model which are typically learned through a complicated log-likelihood loss. In essence, as shown in Figure 1, the algorithm "transports" location and geometry-dependent parameters of the model from one place to another place by examining the similarities among LIDAR scans. This parameter transport procedure exploits geometry-dependent kernels with less computational cost, resulting in a higher quality maps. With this, we bring the following contributions,

1) a theoretical framework for parameter transfer in robotics;
2) intra-domain transfer: sequentially building a map based on features learned in previous time frames;
3) inter-domain transfer: mapping an environment with features learned from another environment. This includes parameter transfer from one town to another, static to dynamic environments, and simulation to real-world; and
4) online and efficient mapping of large-scale 2D and 3D environments.

Notation given in Table I will be used throughout the paper.

TABLE I: Table of notations and terminology

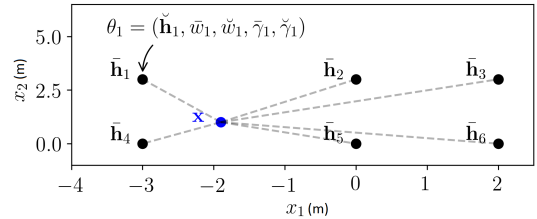| Notation	Description
$\bar{}$ and $\check{}$	Mean and variance of Gaussian; shape and scale of Gamma
$\mathbf{x}$ and y	LIDAR data positions and labels
N and M	Number of data points and number of parameters
$\bar{\mathbf{h}}$	Kernel positions
θ	Parameter set except $\bar{\mathbf{h}}$
$^{(\mathcal{S})}$ and $^{(\mathcal{T})}$	Source and target
P	Coupling matrix
$a \rightarrow b$	transport = transfer = domain adaptation = transform = map = convert (from a to b)

Fig. 2: Kernel positioning. Kernels are placed in different locations $\bar{\mathbf{h}}$. For instance, here, the distance between each data point $\mathbf{x}$ and $\{\bar{\mathbf{h}}_m\}_{m=1}^{M=6}$ has to be evaluated as in eq. 1.

II. PRELIMINARIES

A. Uncertainty of Occupancy

An occupancy model is typically represented as a parameterized function that models the occupancy probability of each location in the environment. The objective is to learn the model parameters θ given a set of observations from LIDAR beams. Once the parameters are estimated, it is possible to query $y_* = p(\text{occupied}|\mathbf{x}_*, \theta) \in [0, 1]$ anywhere in the 2D space[1] $\mathbf{x}_* \in \mathbb{R}^2 := (x_1, x_2)$. Labeling LIDAR hits as $y = 1 = \text{occupied}$ and randomly sampled points between each LIDAR hit and the LIDAR sensor as $y = 0 = \text{free}$, a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$ can be generated. Here, $\mathbf{x}_n \in \mathbb{R}^2$ are the corresponding spatial locations of $y_n \in \{0, 1\}$.

Various models have been proposed for the occupancy function. Gaussian process occupancy maps (GPOMs) [20, 21] have been presented as an alternative to improve occupancy grid mapping (OGM) [22, 23] and Hilbert maps [24]. In addition to considering neighborhood information for accurate occupancy predictions, kernel methods used in GPOMs come with the flexibility of incorporating other aspects such as dynamics into occupancy mapping [25, 26]. On the other hand, GPOMs account for uncertainty as they are based on a Bayesian nonparametric model. Regardless of their attractive theoretical properties, GPOMs are impractical for real-world usage because of the $\mathcal{O}(N^3)$ run-time and memory complexity. Recently proposed Bayesian Hilbert maps (BHMs) [27], on the other hand, encompass all positive traits of GPOMs but at a cost of $\mathcal{O}(M^3)$ where $M \ll N$ is the number of features that correlates with the accuracy. Since ABHM considers the full

[1]We limit our discussion to 2D for simplicity. All theory are readily extensible to 3D.
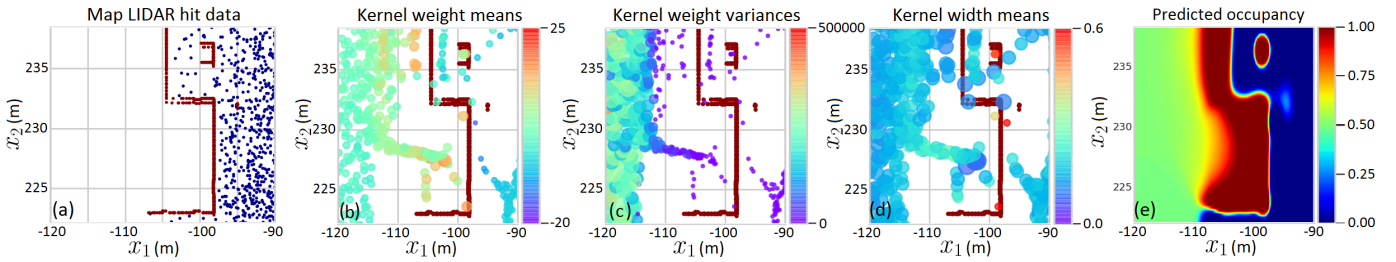
Fig. 3: Spatial correlation among obstacles in the environment and some ABHM parameters. (a) LIDAR data: $y = 1$ (hits) in red and $y = 0$ in blue. (b)-(d) kernel weight means $\bar{w}_m$, weight variances $\check{w}_m$, and width means $\bar{\gamma}_m$. Each point is a kernel placed in the shown location $\bar{\mathbf{h}}$ in the x_1-x_2 space. Refer Observation 1 for further interpretation. (e) Predicted occupancy.

Bayesian treatment over parameters of [27] to account for local spatial changes in the environment, it achieves a significantly higher accuracy.

BHM can be summarized as performing Bayesian logistic regression in a high-dimensional feature space $\mathbb{R}^M$ using *k*ernels [28, 29]. BHM uses the same kernel for the entire map. ABHM is an extension to BHM to learn all location-dependent nonstationary kernel parameters (Appendix I-C). While BHM can be run in near real-time in an online fashion, ABHM is computationally expensive as it requires learning thousands of parameters offline. In ABHM, the occupancy probability of a point $\mathbf{x}_*$ is given by,

$$p(y_* = 1|\mathbf{x}_*) = \text{sigmoid}\left(\sum_{m=1}^{M} w_m \underbrace{\exp\left(-\gamma_m \|\mathbf{x}_* - \mathbf{h}_m\|_2^2 \right)}_{m^{\text{th}} \text{ SE kernel}} \right),$$

(1)

where w, $\mathbf{h}$, and γ are parameters learned from data $\mathcal{D}$. The inner part of the equation is a w weighted sum of M kernels placed in 2D spatial locations $\mathbf{h}$. In areas where there are more LIDAR hits in the locality of a kernel, then its associated weight w_m will be higher, and vice versa. This is because, as illustrated in Figure 3, here, M squared-exponential (SE) kernels positioned at mean locations $(\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \ldots, \bar{\mathbf{h}}_M)$ are used to project 2D data into an M dimensional vector such that each kernel has more effect from data in its locality. γ are positive parameters that control the width of each kernel. Probability distributions $w_m \sim \mathcal{N}(\bar{w}_m, \check{w}_m)$, $\mathbf{h}_m \sim \mathcal{N}(\bar{\mathbf{h}}_m, \check{\mathbf{h}}_m)$, and $\gamma_m \sim \text{Gamma}(\bar{\gamma}_m, \check{\gamma}_m)$ are induced on the parameters to naturally encode uncertainty. Here, slightly abusing standard notations, $\bar{}$ and $\check{}$ symbols are used to represent the mean and dispersion parameters, respectively (Table I).

The parameters of the model are learned using variational inference [13]. See Figure 3 for some of the estimated parameters. Since there are 8 parameters ($\bar{w}_m, \check{w}_m, \bar{\gamma}_m, \check{\gamma}_m \in \mathbb{R}$ and $\bar{\mathbf{h}}_m, \check{\mathbf{h}}_m \in \mathbb{R}^2$) associated with each kernel, it is required to learn $8M$ parameters. In order to achieve a practically satisfactory accuracy to cover a 100 m^2 area, it is necessary to have over 10000 kernels which would take around 10 minutes on a GPU. On the other hand, although ABHM provides high-quality maps, it is required to first collect the entire dataset as it does not support sequential training, making it practically unsuitable for mobile robotics applications.

B. Domain Adaptation

The learned model parameters for a sample environment can be visualized in Figure 3.

Observation 1: Once the full ABHM model is learned, the following can be observed: topsep=0pt,itemsep=-1ex,partopsep=1ex,parsep=1ex

1) As shown in Figure 3 (b), the mean values of weights $\bar{w}$ are higher in areas where there are LIDAR hits, and vice versa. In areas where there are no observations at all ($x_1 \lesssim -105$ in Figure 3 (a)), the variance values $\check{w}$ are high as shown in Figure 3 (c).
2) The mean widths $\bar{\gamma}$, as can be onserved in Figure 3 (d), are higher close to the obstacles, indicating sharp edges.
3) The mean positions of kernels $\bar{\mathbf{h}}$ align according to the geometry of the obstacles (Figure 3 (b)-(d)).

Premise 1: Based on Observation 1, there is geometric correspondence between parameter values and obstacles observed by the LIDAR. Therefore, we argue that spatially dependent parameters for a new environment, defined as the target domain, can be estimated by discovering correspondence between the target (new) LIDAR data and source (known) LIDAR data with associated parameters. Here, the source is an environment whose parameters are known or pre-estimated using a method such ABHM in a simple environment, and the target is a complex and large environment whose parameters are not known and challenging to estimate. This requires transferring features from source to target domains.

Transferring knowledge obtained from one domain to the other has been widely discussed in the machine learning literature [1, 30]. The broader class of transferring from one type of domain to the other, e.g. images to text, is known as transfer learning. If the type of source and target domains are the same, as in occupancy mapping, the transfer process is called domain adaptation (DA). Applications in robotics include transferring control policies from simulation to real-world [2, 31], and making image processing tasks invariant to lighting and other changes [5, 32].

Variations of generative adversarial networks (GANs) such as DTN [33], CycleGAN [34], DiscoGAN [35], UNIT [36], DART [4] have been widely used for domain adaptation of RGB images. However, not only do these methods require a large amount of data but also it is not immediately clear

how to use these techniques with sparse LIDAR data nor transferring probability distributions. In the next section, we consider an alternative domain adaptation method based on optimal transport (OT) [17] to transfer parameters of the Bayesian occupancy model using sparse LIDAR data.

III. OPTIMAL PARAMETER TRANSPORT

In this section, we present the proposed algorithm. Transferring parameters is a two-step procedure: creating a source dataset offline (Section III-A) and transferring them to a target domain online (Section III-B). Section III-C is a generalization and is the actual algorithm used in experiments. Section III-D is an extension to further improve the map quality.
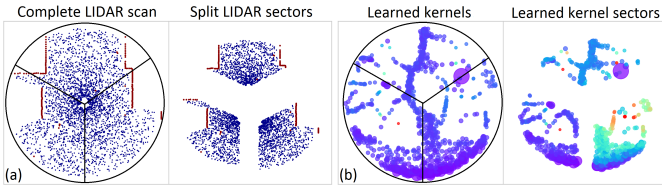
A. Preparing the Source Dictionary of Atoms

Fig. 4: Extracting source data. (a) Splitting source LIDAR scans into 3 sectors. (b) Corresponding kernels parameters are also split the same way. Only kernel position means and weight means are shown here.

In order to take advantage of domain adaptation we must have accurately pre-trained maps from which we can extract spatially relevant features. In the context of our problem, we must extract LIDAR scans (hits and free) with their corresponding model parameters including kernel weights, positions, and widths. To provide high-quality training data we extract learned model parameters from ABHM maps. Since ABHM can only be used on small areas due to the high computational cost, we learn separate ABHM maps for different areas and construct a dictionary of source atoms which we call a *dictionary of atoms*.

To construct the dictionary, as illustrated in Figure 4, we split each LIDAR scan into circular sectors with radii equal to the specified maximum LIDAR distance. Rather than using the entire LIDAR scan as the source dataset, this split not only results in a diverse set of geometric primitives but also provides simpler sources for the transfer procedure presented in the following section. The corresponding learned model parameters for each sector are considered as source parameters that we wish to transfer to the target domain. For each sector, we have $M^{(\mathcal{S})}$ parameters $\{\theta_m^{(\mathcal{S})}\}_{m=1}^{M^{(\mathcal{S})}}$ associated with $N^{(\mathcal{S})}$ LIDAR hits or free points $\{(\mathbf{x}_n^{(\mathcal{S})}, y_n^{(\mathcal{S})})\}_{n=1}^{N^{(\mathcal{S})}}$. The collection of these different LIDAR sectors constitutes the dictionary of source atoms $\mathcal{X}^{(\mathcal{S})}$.

B. Source to Target Parameter Transport

Until we present the general transfer procedure that we used in experiments in Section III-C, for the sake of simplicity of the following discussion, let us assume that the dictionary

of atoms contains only one LIDAR sector and associated parameters.

Objective: Having determined source LIDAR data $\{(\mathbf{x}_n^{(\mathcal{S})}, y_n^{(\mathcal{S})})\}_{n=1}^{N^{(\mathcal{S})}}$ and corresponding parameters $\{\theta_m^{(\mathcal{S})}\}_{m=1}^{M^{(\mathcal{S})}}$, our objective is to determine the new set of parameters $\{\theta^{(\mathcal{T})}\}_{m=1}^{M^{(\mathcal{T})}}$ for a new LIDAR dataset $\{(\mathbf{x}_n^{(\mathcal{T})}, y_n^{(\mathcal{T})})\}_{n=1}^{N^{(\mathcal{T})}}$. This problem is illustrated in Figure 5 (a) and (b). In other words, we are looking for a nonlinear mapping technique to convert a source $(\mathcal{S})$ to a target $(\mathcal{T})$. We recognize this as an *optimal transport (OT) problem* given in Theorem 1.

Theorem 1: (Monge-Kantorovich) [17] Let $\Omega^{(\mathcal{S})}$ and $\Omega^{(\mathcal{T})}$ be two separable metric spaces such that probability measures $\boldsymbol{\mu}^{(\mathcal{S})}$ and $\boldsymbol{\mu}^{(\mathcal{T})}$ on $\Omega^{(\mathcal{S})}$ and $\Omega^{(\mathcal{T})}$, respectively, are Radon measures. The optimal coupling,

$$P_* = \underset{P \in \Gamma(\boldsymbol{\mu}^{(\mathcal{S})}, \boldsymbol{\mu}^{(\mathcal{T})})}{\mathrm{arginf}} \int_{\Omega^{(\mathcal{S})} \times \Omega^{(\mathcal{T})}} D(\boldsymbol{\mu}^{(\mathcal{S})}, \boldsymbol{\mu}^{(\mathcal{T})}) \mathrm{d}P(\boldsymbol{\mu}^{(\mathcal{S})}, \boldsymbol{\mu}^{(\mathcal{T})}), \tag{2}$$

always exists for a distance function $D : \Omega^{(\mathcal{S})} \times \Omega^{(\mathcal{T})} \to [0, \infty)$, where Γ is the set of all couplings (probability measures) on $\Omega^{(\mathcal{S})}$ and $\Omega^{(\mathcal{T})}$ with marginals $\boldsymbol{\mu}^{(\mathcal{S})}$ and $\boldsymbol{\mu}^{(\mathcal{T})}$, respectively.

Intuitively, as illustrated in Figures 5 (a) and 6, the OT problem attempts to determine the optimal way to move one probability distribution to another. If $\boldsymbol{\mu}^{(\mathcal{S})}$ and $\boldsymbol{\mu}^{(\mathcal{T})}$ constitute two datasets of size $N^{(\mathcal{S})}$ and $N^{(\mathcal{T})}$, respectively, there always exists an optimal probabilistic coupling $P_* \in \mathbb{R}^{N^{(\mathcal{S})} \times N^{(\mathcal{T})}}$ between the two datasets [37]. Here, as shown in Figures 6 where the source and target samples are assumed to separately follow bivariate distributions, P_* is a doubly stochastic matrix—each row and column sums to one—that indicates the probability of a sample in the source match with all other points in the target. In occupancy mapping, $\boldsymbol{\mu}$ is computed as Dirac measures from LIDAR data (Appendix I-A).

With source data obtained in Section III-A, for a new target dataset, we attempt to obtain the optimal coupling,

$$P_* = \underset{P \in \Gamma(\mathbf{x}^{(\mathcal{S})}, \mathbf{x}^{(\mathcal{T})})}{\mathrm{argmin}} \sum_{ij} P_{ij} D_{ij} - \lambda^{-1} r(P), \tag{3}$$

for a given $D \in \mathbb{R}^{N^{(\mathcal{S})} \times N^{(\mathcal{T})}}$ distance matrix (e.g. squared Euclidean distance between source-target pairs) with the information entropy of P,

$$r(P) = -\sum_{ij} P_{ij} \log P_{ij}. \tag{4}$$

This entropic regularization, commonly known as the Sinkhorn distance [38, 39], enables solving the otherwise hard integer programming problem using an efficient iterative algorithm [40]. Here, λ controls the amount of regularization[2].

Having obtained the optimal coupling between source and target LIDAR, as illustrated in Figures 5 (b)-(c) and 7, now it is possible to transport source parameters $\theta^{(\mathcal{S})}$ to the target domain. This is done by associating the source parameter positions $\bar{\mathbf{h}}^{(\mathcal{S})}$ with source samples $\mathbf{x}^{(\mathcal{S})}$ as a linear map [41], and

---

[2]λ can be set to a large number depending on the machine precision of the computer.

transporting them to the target domain $\bar{\mathbf{h}}^{(\mathcal{S})} \to \bar{\mathbf{h}}^{(\mathcal{T})}$ according to the coupling matrix P_* learned from LIDAR matching. All other $\theta^{(\mathcal{S})}$ parameters associated with the kernels positioned at $\bar{\mathbf{h}}^{(\mathcal{S})}$ will also be transported to the target domain. This implicit transfer process is depicted in Figure 7.

C. Transport from a Dictionary of Atoms

Although we created a dictionary of atoms consisting of diverse geometric primitives in Section III-A, the transfer procedure introduced in Section III-B was limited to a single LIDAR sector. In order to effectively make use of the entire dictionary, it is required to find the optimal coupling matrix over all elements in the dictionary $\mathbf{x}^{(\mathcal{S})} \in \mathcal{X}^{(\mathcal{S})}$.

As another fact, although eq. 3 can be used to obtain a translation and scale invariant solution, it is not robust enough against large rotation variations. However, we can rotate data about the centroid of each atom using the rotation matrix,

$$R(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad (5)$$

for a discrete set of rotations $\alpha \in \mathcal{A}$.

Overall, we obtain a candidate optimal coupling set of size $|\mathcal{X}^{(\mathcal{S})}| \times |\mathcal{A}|$ by minimizing eq. 3 over all rotations and atoms,

$$\mathcal{P}_* = \left\{ \underset{P \in \Gamma(\mathbf{x}^{(\mathcal{S})}, R(\alpha)\mathbf{x}^{(\mathcal{T})})}{\operatorname{argmin}} \sum_{ij} P_{ij} D_{ij} - \lambda^{-1} r(P) \right\}_{\substack{\mathbf{x}^{(\mathcal{S})} \in \mathcal{X}^{(\mathcal{S})} \\ \alpha \in \mathcal{A}}}. \quad (6)$$

Ultimately, we select the overall best coupling matrix from the candidate set $\mathcal{P}_*$ as the candidate that has the minimum 2-Wasserstein distance (refer Appendix I-B) to the target,

$$P_* = \underset{P \in \mathcal{P}_*}{\operatorname{argmin}} \sum_{ij} P_{ij} D_{ij}. \quad (7)$$

This P_* can now be used to transfer parameters using the same method explained in Figure 7. As a result of the

computation procedure introduced in this section, as depicted in Figure 8, atoms from various domains will be transferred to the target. Because atoms only consist of a few hundred LIDAR points, this transfer can be performed in real-time. Unlike in BHM or ABHM, we can now introduce thousands of kernels. The increasing number of pre-learned kernels as well as the nonstationarity help to improve the accuracy. The entire Parameter Optimal Transport (POT) algorithm is summarized in Algorithm 1.

D. POT Maps and Refined POT Maps

Transporting parameters can be performed in two different ways. It is possible to transport parameters for each LIDAR scan separately, and immediately build the occupancy map. This results in an *instantaneous map* which is useful for understanding the occupancy of the surrounding at present. Such maps can be used for safe decision-making and control in the locality of the robot. On the other hand, it is also possible to build the *overall map* by sequentially aggregating the transported parameters as the robot moves. The overall map model completely discards training LIDAR data after transporting the parameters. This enables mapping large areas at a constant cost.

Once the parameters are transported with the intention of building an instantaneous or overall map, an occupancy map can be generated by plugging in the transported parameters to eq. (1) and querying occupancy probabilities. It will not only provide the mean occupancy map, but also the uncertainty as the variance estimate. Since only the parameters of the continuous mapping function eq.1 are stored, the occupancy map can later be queried *at any time at any resolution*.

Learning kernel parameters γ and $\mathbf{h}$ in real-time is not feasible with ABHM. However, learning weights w, assuming other parameters are given, we have a fast approximation given by Bayesian Hilbert maps (BHMs) [27]. As an additional step to further improve the map quality, we propose to *use*
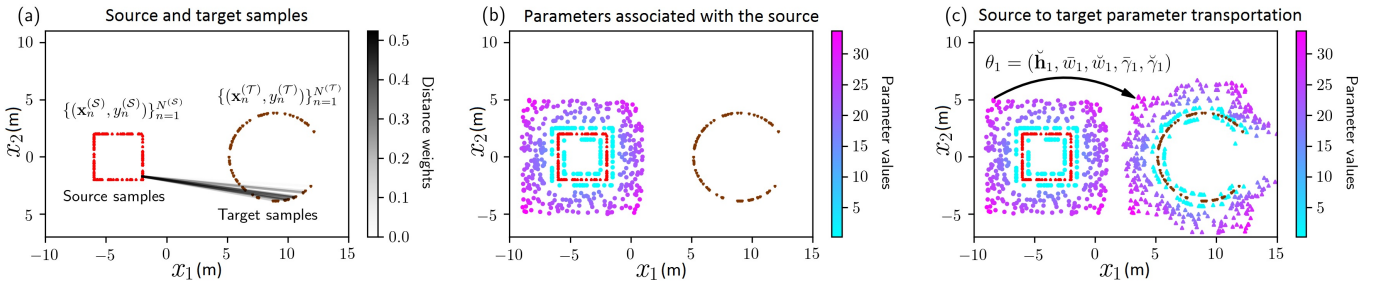

Fig. 5: Optimal transport from a square to an arc. (a) If there are $N^{(\mathcal{S})}$ and $N^{(\mathcal{T})}$ number of data points in the source (red) and target (brown) datasets, the coupling matrix γ is size $N^{(\mathcal{S})} \times N^{(\mathcal{T})}$ where any column or any row sums to 1. A given row in γ indicates the probabilities of the sample associated with that row could be coupled to all samples in the target dataset. Probabilities associated with one such source point to target matches are shown in white-black color scale. Note that only the 10 highest matches are shown for clarity. (b) For a given set of LIDAR hits (red) spatial parameters can be learned using ABHM. Here we see kernel parameters spread across the environment. However, for another set of LIDAR hits (brown) we would prefer not re-learning parameters because it is expensive. (c) Based on the coupling matrix between the source and the target, we transport (move from the target area to the source area) the parameters around each point. Note that how the small lengthscales (cyan) stays close to the LIDAR hits and larger lengthscales (magenta) move away from the LIDAR.
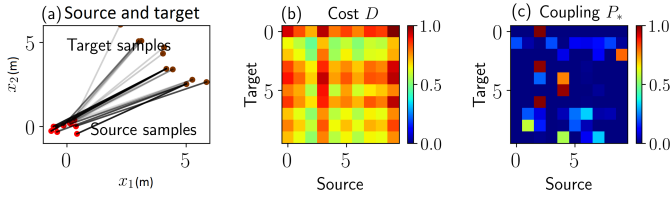
Fig. 6: (a) 10 red and 10 brown dots indicate samples in $\mathbb{R}^2$ from the bivariate source and target distributions, respectively. The higher the transparency of gray lines, the lower the probability of couplings (matches) obtained after solving eq. 3. (b) 10×10 pairwise cost matrix D between the positions of samples. (c) 10×10 coupling matrix P_* indicates the optimal coupling probability of source points and all other target points. Determining this matrix (and gray lines in (a)) is the goal of optimal transport.

$$\text{learn } P_* \text{ for } \mathbf{x}^{(\mathcal{S})} \xrightarrow[\text{transport}]{\text{explicit}} \mathbf{x}^{(\mathcal{T})}$$

$$\text{predict} \quad \bar{\mathbf{h}}^{(\mathcal{S})} \xrightarrow[\text{transport}]{\text{explicit}} \bar{\mathbf{h}}^{(\mathcal{T})} \text{ using } P_*$$
$$\vdots \qquad \qquad \vdots$$
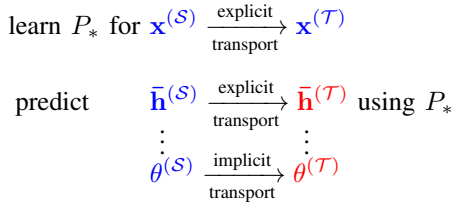$$\theta^{(\mathcal{S})} \xrightarrow[\text{transport}]{\text{implicit}} \theta^{(\mathcal{T})}$$

Fig. 7: Parameter optimal transport. Known and unknown quantities are in blue and red, respectively. We learn an optimal coupling matrix P_* using source and target LIDAR. Then we use this coupling matrix to predict target kernel positions corresponding to the source kernel position. By doing this, the other parameters associated with each kernel are also implicitly transported by treating them as labels.

transported parameters as prior distributions of the BHM and simply update the weights w by using [27]. We call this improved map, the refined POT (RePOT) map.

IV. EXPERIMENTS

Both simulated and real-world datasets were used to assess the quality of POT. To generate simulated data, Carla v.0.9.2 simulator [42] was used as it closely resembles real-world towns. As a real-world dataset, we used the KITTI benchmark dataset [43]. All datasets are listed in Table II and each of these environments is considered as a domain. More details are provided in Appendix II-A. As evaluation metrics, we used accuracy (ACC), area under ROC curve (AUC) and negative log-likelihood (NLL) [44]. Unlike ACC and AUC, NLL takes into account uncertainty of predictions. The higher the AUC or lower the NLL, the better the model is.

A. Intra-domain and Inter-domain Adaptation

In this experiment, we consider two paradigms: intra-domain and inter-domain transfer. In intra-domain transfer, the source atoms are generated from the first 10 frames of a particular dataset and parameters are transferred to the rest of the same dataset while they are transferred to a completely different domain in inter-domain transfer. Based

---

Algorithm 1: Transferring parameters to a new domain

Input: New LIDAR scans, Source dictionary of atoms
while *new scan in new domain* **do**
    $\mathcal{P}_* = \{\}$;
    for *each atom in $\mathcal{X}^{(\mathcal{S})}$* **do**
        for *each rotation in $\mathcal{A}$* **do**
            $\mathcal{P}_*$.insert(Compute the coupling matrix)
            (Eq. 6);
        end
    end
    $P_* \leftarrow$ Determine the best coupling matrix (Eq. 7);
    $\theta^{(\mathcal{T})} \leftarrow$ Transfer the source parameters to the
    target domain using P_* (Figure 7);
end
Output: Parameters $\theta^{(\mathcal{T})}$

TABLE II: Description of domains

Domains (Datasets)	Description
Carla Town 1	a 2D dataset in town 1 in Carla (3.7 km).
Carla Town 2	a 2D dataset in town 1 in Carla (1.5 km).
Carla Town 3	a 2D dataset in town 1 in Carla (8.6 km).
Carla Town 1 3D	a 3D dataset in town 1 in Carla.
Carla Town 1 Dyna	Carla Town 1 with 120 vehicles running around.
KITTI Dyna	a 2D dataset (the middle LIDAR channel).

on results reported in Table III with 20% randomly sampled test LIDAR beams from each town, it is possible to accurately transfer parameters using POT. This enables mapping large scale towns in real-time. All parameters are aggregated over time to build occupancy maps of the entire environments as visualized in Figure 9 and Appendix II-C. Using the Town 1 3D dataset, we demonstrate the possibility of extending POT to 3D environments. In this case, source atoms described in Section III-A, were circular cylindrical sectors (i.e. pie slice shaped). The post-hoc refinement procedure, RePOT, introduced in Section III-D, further improved the map significantly. A visualization of RePOT is shown in Figure 10 and performance improvement, in direct comparison with results in Table III, is reported in Table IV.

B. Building Instantaneous Maps

This experiment was designed to demonstrate how parameters can be instantaneously transported to build the instantaneous map of the surrounding. For this purpose, we used the two dynamic environments: Town 1 Dyna and KITTI Dyna. The source dictionary of atoms was prepared similar to the intra/inter-domain adaptation experiment. Such a map is shown in Figure 1. The performance of the model was evaluated on 20% of data that were not used for optimal transport. Table V shows the performance of transferring features extracted from each town to the dynamic datasets.

C. Performance Comparison

In this experiment, we compared various occupancy mapping algorithms in terms of accuracy and speed. Since these
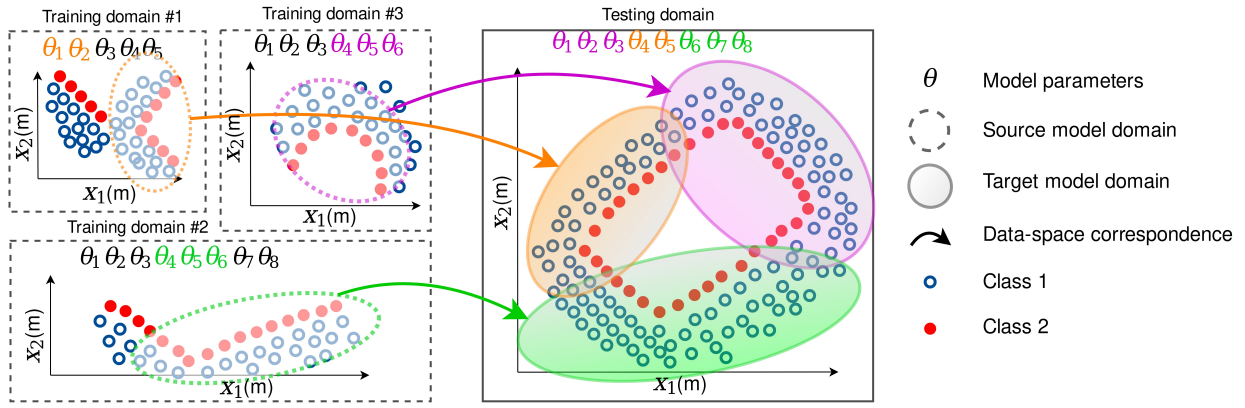
Fig. 8: A high-level overview of the proposed method: Parameter Optimal Transport (POT). *Training domains* correspond to potentially independent, data-intensive, expensive, yet small-scale pre-learned models. After storing in a dictionary of atoms, representative data-space and model-parameter tuples from the pre-learned set of models, we find data-space correspondences using optimal transport. These correspondences are then used to transport pre-learned parameters to out-of-sample *test domains*.

TABLE III: Performance of intra-domain (diagonal entries of the table) and inter-domain (off-diagonal entries of the table) transfer.

			Target		
			Town1	Town2	Town3
Source	ACC	Town1	0.79	0.82	0.76
		Town2	0.70	0.72	0.58
		Town3	0.85	0.83	0.84
	AUC	Town1	0.88	0.88	0.90
		Town2	0.85	0.83	0.83
		Town3	0.92	0.92	0.93
	NLL	Town1	1.14	0.97	1.40
		Town2	3.30	3.23	5.98
		Town3	1.64	1.69	1.79

TABLE IV: Performance metrics of Refined POT (Re-POT) across both intra- and inter-domain transfers.

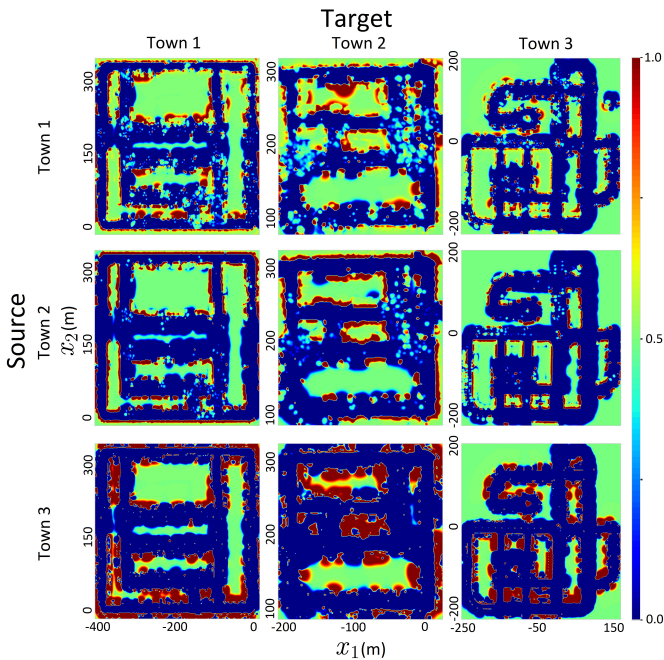|			Target		
			Town1	Town2	Town3
Source	ACC	Town1	0.95	0.93	0.95
		Town2	0.91	0.91	0.92
		Town3	0.95	0.92	0.93
	AUC	Town1	0.99	0.98	0.98
		Town2	0.98	0.98	0.98
		Town3	0.99	0.97	0.97
	NLL	Town1	0.71	1.4	1.12
		Town2	1.40	1.74	1.85
		Town3	0.96	1.62	1.44

Fig. 9: Transported occupancy maps for the inter and intra domain adaptation experiments using the town datasets. From top to bottom and left to right are towns 1, 2, and 3.

TABLE V: Instantaneous maps in dynamic environments: Experiments for sim2sim and sim2real with mean and SD.

			Target	
			Town 1 Dyna	KITTI Dyna
Source	ACC	Town 1	0.74 ± 0.10	0.69 ± 0.06
		Town 2	0.70 ± 0.10	0.58 ± 0.06
		Town 3	0.74 ± 0.11	0.71 ± 0.07
	AUC	Town 1	0.81 ±0.11	0.77 ±0.06
		Town 2	0.77 ±0.12	0.73 ±0.06
		Town 3	0.78 ±0.15	0.73 ±0.09
	NLL	Town 1	1.06 ± 0.56	1.42 ± 0.38
		Town 2	1.90 ± 0.79	3.63 ± 1.04
		Town 3	1.89 ± 1.30	2.30 ± 0.83

algorithms cannot be trained or queried in a similar fashion, we measured the per time unit performance. For instance, ABHM can only be trained in small environments although our datasets consist of large towns. Firstly, we measure the time for running POT per LIDAR scan. Then we decide the number of kernels to match the same runtime for BHM and ABHM. Results are reported in Table VI. Though OGM cannot be computed per time basis, we report the results for reference (See Appendix II-B). GPOM cannot be executed for datasets this large. As expected, ABHM outperforms BHM in all metrics because ABHM is a nonstationary model that takes into
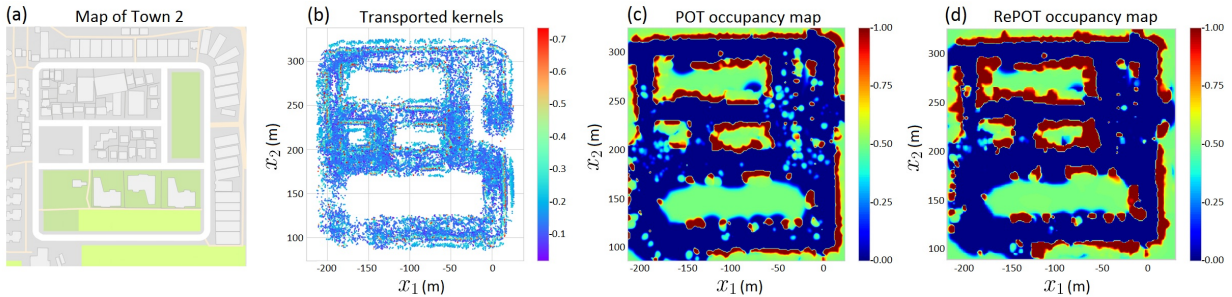
Fig. 10: Large-scale map building with POT and RePOT. (a) Carla Town 2 plan. (b) Transferred kernel mean width and position parameters. (c) Occupancy prediction with POT. (d) Occupancy prediction with RePOT.

TABLE VI: Performance per time unit for RePOT, POT, ABHM, and BHM. Though OGM results are reported for reference purposes, unlike other methods, OGM cannot be computed for per time unit basis.

	Method	Town1	Town2	Town3
			Target	
ACC	RePOT	0.95	0.93	0.95
	POT	0.85	0.83	0.84
	ABHM	0.77	0.59	0.86
	BHM	0.66	0.61	0.71
	OGM	0.78	0.78	0.77
AUC	RePOT	0.99	0.98	0.98
	POT	0.92	0.92	0.93
	ABHM	0.95	0.96	0.96
	BHM	0.94	0.92	0.91
	OGM	0.89	0.91	0.90
NLL	RePOT	0.71	1.41	1.12
	POT	1.64	1.69	1.79
	ABHM	0.58	0.71	0.41
	BHM	0.63	0.69	0.61
	OGM	2.00	1.34	1.13

account local geometry. Theoretically, in the infinite memory and computation time limit, ABHM should outperform all methods. Nonetheless, practically, POT has a higher ACC and AUC compared to ABHM as POT can transfer kernels online to accommodate the complexity of the environment. However, the increase in NLL in POT compared to ABHM, indicates the inherent uncertainties of the transfer procedure. Once the weights were refined using RePOT, NLL has dropped as the weight distributions can be optimized to reduce the uncertainty giving better predictions.

Runtime: With a laptop with 4 cores and 8 GB RAM, on average, POT, programmed in Python, takes around 1 s update time. This is without parallelizing any part of the code. Note that eq. (6) is highly parallelizable making the algorithm $|\mathcal{X}^{(\mathcal{S})}| \times |\mathcal{A}|$ faster (approx. 25 times). This is a significant improvement to algorithms such as BHM and ABHM which would take several hours to build a large-scale map as they rely on complicated variational inference procedures. POT runtime increases with increasing λ in the Sinkhorn algorithm we used in POT. As $\lambda \to \infty$ the convergence is guaranteed.

V. DISCUSSION

In optimal transport, we consider the problem of transforming one probability measure to another. This also loosely relates to the point cloud registration problem typically addressed by the iterative closest point (ICP) algorithm [45]. However, unlike ICP which only has a single set of translation and rotation parameters, in optimal transport, each data point in the source dataset has a highly nonlinear relationship with every other point in target datapoints through the optimal coupling matrix P_*. Another reason why we cannot resort to a popular algorithm such as ICP is because it only works for slight changes in translation and rotation. When a robot moves in dynamic environments, it is essential to adapt for sudden, potentially large, nonlinear changes in geometry.

One remarkable aspect of being able to transport distributions is that it endows us the ability to adapt Bayesian models in the sense of an *informed prior* [46] enabling expedited parameter tuning. We have demonstrated such a use case in RePOT with significant improvements in overall map quality.

Although our method was presented and demonstrated in the context of occupancy mapping, there are many other potential applications in robotics. For example, the theory can be potentially used for domain adaptation of policy parameters where a policy is trained in one environment and needs to be transferred to another. For example, a particular robotic arm is trained to grasp objects on a table and performs well on this task. One could, in principle adapt policies for use in another arm without retraining the policy from the start. Finally, it can also be used for sim2real where models are learned in simulation and transferred to the physical world, saving significant time and cost in running real robots.

VI. CONCLUSION

This paper introduced parameter optimal transport (POT), an efficient framework for geometric domain adaptation. By combining the formalism of automorphing Bayesian Hilbert maps with optimal transport theory, patterns from one environment can be seamlessly transferred to another in a fraction of a second. We show that this framework can be effectively used to map large urban environments, transferring learned patterns between two cities, between simulated and real environments, and between static and dynamic environments.

REFERENCES

[1] H. B. Ammar, E. Eaton, J. M. Luna, and P. Ruvolo, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015, pp. 3345–3351.

[2] ——, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1126–1135.

[4] X. Fang, H. Bai, Z. Guo, B. Shen, S. Hoi, and Z. Xu, "Dart: Domain-adversarial residual-transfer networks for unsupervised cross-domain image classification," *arXiv preprint arXiv:1812.11478*, 2018.

[5] M. Wulfmeier, A. Bewley, and I. Posner, "Incremental adversarial domain adaptation for continually changing environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1–9.

[6] F. Meier, P. Hennig, and S. Schaal, "Efficient bayesian local model learning for control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2014, pp. 2244–2249.

[7] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *International Conference on Machine Learning (ICML)*, 2011, pp. 465–472.

[8] M. Wüthrich, C. Garcia Cifuentes, S. Trimpe, F. Meier, J. Bohg, J. Issac, and S. Schaal, "Robust gaussian filtering using a pseudo measurement," in *American Control Conference (ACC)*, 2016.

[9] J. Campbell and H. B. Amor, "Bayesian interaction primitives: A slam approach to human-robot interaction," in *Conference on Robot Learning (CoRL)*, 2017, pp. 379–387.

[10] B. Burchfiel and G. Konidaris, "Bayesian eigenobjects: A unified framework for 3d robot perception." in *Robotics: Science and Systems (RSS)*, 2017.

[11] V. V. Unhelkar and J. A. Shah, "Learning models of sequential decision-making without complete state specification using bayesian nonparametric inference and active querying," *Massachusetts Institute of Technology*, 2018.

[12] D. Isele, M. Rostami, and E. Eaton, "Using task features for zero-shot knowledge transfer in lifelong learning." in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016, pp. 1620–1626.

[13] R. Senanayake, A. Tompkins, and F. Ramos, "Automorphing kernels for nonstationarity in mapping unstructured environments," in *Conference on Robot Learning (CoRL)*, 2018, pp. 443–455.

[14] P. A. Lasota, T. Fong, J. A. Shah *et al.*, "A survey of methods for safe human-robot interaction," *Foundations and Trends in Robotics*, vol. 5, no. 4, pp. 261–349, 2017.

[15] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. N. Zeilinger, J. H. Gillula, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes," in *IEEE Conference on Decision and Control (CDC)*, 2014, pp. 443–455.

[16] G. Vallicrosa and P. Ridao, "H-slam: Rao-blackwellized particle filter slam using hilbert maps," *Sensors*, vol. 18, no. 5, 2018.

[17] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[18] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[19] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas, "Convolutional wasserstein distances: Efficient optimal transportation on geometric domains," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–11, 2015.

[20] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *International Journal of Robotics Research (IJRR)*, vol. 31, no. 1, pp. 42–62, 2012.

[21] J. Wang and B. Englot, "Fast, accurate gaussian process occupancy maps via test-data octrees and nested bayesian fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1003–1010.

[22] A. Elfes, "Occupancy grids: a probabilistic framework for robot perception and navigation," Ph.D. dissertation, Carnegie Mellon University, 1989.

[23] D. Arbuckle, A. Howard, and M. Mataric, "Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 2002, pp. 409–414.

[24] F. Ramos and L. Ott, "Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent," in *Robotics: Science and Systems (RSS)*, 2015.

[25] R. Senanayake, S. O'Callaghan, and F. Ramos, "Learning highly dynamic environments with stochastic variational inference," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[26] R. Senanayake, L. Ott, S. O'Callaghan, and F. T. Ramos, "Spatio-temporal hilbert maps for continuous occupancy representation in dynamic environments," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 3925–3933.

[27] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning (CoRL)*, 2017, pp. 458–471.

[28] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, pp. 1171–1220, 2008.

[29] R. Senanayake and F. Ramos, "Building continuous occupancy maps with moving robots," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[30] S. J. Pan and Q. Yang, "A survey on transfer learning,"

IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[31] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*.   IEEE, 2018, pp. 4243–4250.

[32] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[33] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *European Conference on Computer Vision (ECCV)*.   Springer, 2016, pp. 597–613.

[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *International Conference on Computer Vision (ICCV)*, 2017.

[35] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International Conference on Machine Learning (ICML)*, 2017, pp. 1857–1865.

[36] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 700–708.

[37] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 9, pp. 1853–1865, 2017.

[38] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2292–2300.

[39] A. Genevay, G. Peyré, and M. Cuturi, "Learning generative models with sinkhorn divergences," *1608-1617*, 2018.

[40] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[41] M. Perrot, N. Courty, R. Flamary, and A. Habrard, "Mapping estimation for discrete optimal transport," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 4197–4205.

[42] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[43] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.

[44] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[45] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611.   International Society for Optics and Photonics, 1992, pp. 586–606.

[46] Q. F. Gronau, S. Van Erp, D. W. Heck, J. Cesario, K. J. Jonas, and E.-J. Wagenmakers, "A bayesian model-averaged meta-analysis of the power pose effect with informed and default priors: The case of felt power," *Comprehensive Results in Social Psychology*, vol. 2, no. 1, pp. 123–138, 2017.