Robotics: Science and Systems 2020
Corvalis, Oregon, USA, July 12-16, 2020

Nonparametric Motion Retargeting for Humanoid Robots on Shared Latent Space

Sungjoon Choi
Disney Research
Los Angeles, California
Email: sungjoon.choi@disney.com

Matt Pan
Disney Research
Los Angeles, California
Email: matt.k.pan@disney.com

Joohyung Kim
University of Illinois Urbana-Champaign
Champaign, Illinois
Email: joohyung@illinois.edu

***Abstract*—In this work, we present a semi-supervised learning method to transfer human motion data to humanoid robots with its emphasis on the feasibility of transferred robot motions. To this end, we propose a data-driven motion retargeting method named locally weighted latent learning (LWL^2) which possesses the benefits of both nonparametric regression and deep latent variable modeling. The method can leverage both paired and domain-specific datasets and can maintain robot motion feasibility owing to the nonparametric regression and graph-based heuristics it uses. The proposed method is evaluated using two different humanoid robots, the Robotis ThorMang and COMAN, in simulation environments with diverse motion capture datasets. Furthermore, the online puppeteering of a real humanoid robot is implemented.**

I. Introduction

Motion retargeting is the process of transferring motions from motion capture (mocap) or character animation data to humanoid robots or virtual character rigs where there may be disparate morphologies [1]. This process is relevant to several fields where lifelike character animation is of importance; this includes animatronics, computer-generated characters for films, and interactive avatars in virtual environments. The motion retargeting problem consists of finding a mapping between two different skeletal structures. Although this mapping can be performed manually, procedural methods using optimization or machine learning are often advantageous as they can be flexibly applied to different skeletal morphologies.

Although using procedural methods have demonstrated success, these methods often suffer from issues such as requiring large data sets that cover the input domain in a balanced manner, and not guaranteeing feasibility of the generated motions (e.g., self-collisions, exceeding joint limits, etc.). In this paper, we present a semi-supervised learning method for data-driven motion retargeting which places emphasis on feasibility (i.e., self-collision avoidance) of transferred robotic motions from mocap data. To this end, we propose a nonparametric embedding method combined with a deep latent variable model [2] which we refer to as Locally Weighted Latent Learning (LWL^2).

To better ensure the feasibility of the motion retargeted results as well as the shortcomings mentioned above over prior work, our LWL^2 method leverages recently-developed shared latent space modeling [3]. Specifically, the encoder and decoder networks of both mocap and robot pose domains are learned, where it can leverage both paired and domain-specific datasets. Once the mappings from each domain (mocap data and humanoid joint poses) to the shared latent space are constructed, motion transfer is performed via locally weighted regression [4] on the latent space. We show that LWL^2 is advantageous for establishing the feasibility of transferred motions on the robot as careful selection of its parameters causes the LWL^2 to become a table look-up method.

Graph search heuristics are also proposed to find a practicable transition between current and target poses which resemble E-Graphs [5]. Furthermore, to handle the possible imbalance in the training datasets while constructing the latent space, we present a computationally-efficient subset sampling method which approximates determinantal point process (DPP) sampling [6] under mild assumptions. The benefits of using these methods are empirically shown in Sec. V.

The main contributions of this paper are threefold: 1) we propose a motion retargeting method (LWL^2) that enjoys the benefits of both nonparametric regression and a deep latent variable model for motion retargeting with emphasis on the feasibility (that is, avoidance of self-collisions), 2) the use of graph search heuristics to enable achievable, smooth, and efficiently computed transitions between robot poses, and, 3) an efficient subset sampling process that approximates DPP sampling [6], which we term locally approximated-DPP (LA-DPP), to mitigate the effects of the data imbalance issues during mini-batch training.

The remainder of this paper is structured as follows: Sec. II summarizes the existing work in motion retargeting and modeling of shared latent spaces as well as nonparametric methods in robotics. Sec. III introduces a collision-handling method that is used to obtain collision-free poses of a humanoid robot. The proposed motion retargeting method and results are described in Sec. IV and Sec. V, respectively. Concluding remarks follow in Sec. VI.

II. Related Work

In this section, we introduce a summary of existing work related to motion retargeting, shared latent space approaches and nonparametric methods in robotics, forming the foundation of this work.

One of the earliest motion retargeting works [7] focused on finding a mapping between two different skeletons with similar kinematics structures but with varying lengths of limbs by solving a constrained spatiotemporal optimization. However, the lack of physical constraints often led to physically infeasible motions not suited for robotics applications - motion retargeting between human mocap data and robots requires consideration of kinodynamic constraints. In earlier work of

motion retargeting for robots, the mapping between human and robot joints was often manually defined with additional considerations such as balancing or collision-free constraints [8, 9, 10]. For example, Peneo et al. [10] manually defined a mapping between human skeletal joints and an iCub humanoid robot. Choi and Kim [11] proposed an optimization-based motion retargeting method, which first optimizes limb lengths of the source mocap skeleton and solves inverse kinematics to get joint trajectories.

In addition to manual mapping, data-driven methods have also been widely used for motion retargeting [12, 13, 14, 15]. Owing to the flexibility of such methods, Yamane et al. [16] was able to generate convincing motions of non-humanoid characters (i.e., a lamp and a penguin) from human mocap data. In many efforts involving data-driven techniques, Gaussian process latent variable models (GPLVM) have been relied upon to construct common shared latent spaces between the two motion domains [12, 13]. More recently, Yin et al. [3] proposed a new method called associate latent encoding (ALE). ALE uses two different variational auto-encoders (VAEs) with a single shared latent space to transform sensory perception inputs into a sequence of joint motion commands. In Sec. V-C, we compare our method with ALE in a comprehensive set of test motions. Despite the flexibility and scalability of data-driven motion retargeting, such methods often suffer from the inability to establish feasibility of output motions on robot platforms.

The proposed LWL^2 is based on a local nonparametric method named locally weighted regression [4, 17] combined with a deep latent variable model [2, 18]. Nonparametric methods [19] are statistical methods where no parametric assumptions are made about the predictive model. They have been widely used in robotics [17, 20, 21, 22] owing to its sample-efficiency and flexibility in adapting new samples. Especially, local methods in nonparametric statistics such as k-nearest-neighbors regression or locally weighted learning (LWL) [4] have been successfully used in a number of different applications, even with high dimensional problems, including learning pole balancing and modeling inverse dynamics of a 30 DOFs humanoid robot [17].

III. Self-Collision Handling Method

Avoiding self-collision is perhaps one of the most critical physical constraints while performing motion retargeting as unchecked self-collisions can lead to unplanned disassembly of the robot in extreme cases. Unfortunately, general self-collision avoidance poses a non-trivial problem owing to the complex geometries of link meshes of many robots. Here, we present an effective self-collision avoidance method that leverages the need to collect a sufficient number of collision-free poses for the establishment of the latent space.

To efficiently handle self-collisions, we model a capsule surrounding each link. We use capsules for two reasons: 1) determining collisions (i.e., intersection) of a capsule with another is computationally efficient, and 2) poses of self-colliding capsules and their links can be adjusted procedurally with ease (see below). Each capsule and its pose are defined by four parameters: position $\mathbf{p}$, orientation R, height h, and radius r, i.e., $\text{cap} = \{\mathbf{p}, R, h, r\}$.

Given two capsules $\text{cap}_1 = \{\mathbf{p}_1, R_1, h_1, r_1\}$ and $\text{cap}_2 = \{\mathbf{p}_2, R_2, h_2, r_2\}$, we define the signed distance between them as the minimum distance between two line segments described by $\mathbf{p}_1$ to $\mathbf{p}_1 + hR_1(:, 3)$ and $\mathbf{p}_2$ to $\mathbf{p}_2 + hR_2(:, 3)$ subtracted by $(r_1 + r_2)$. Capsules, and thus the encapsulated links, deemed to be colliding when the signed distance is negative.

As mentioned earlier, another essential benefit of using capsules is that finding a collision-free pose from a self-collided pose is straightforward. Specifically, suppose we have two links in collision with each other. Let $\mathbf{q}_1$ and $\mathbf{q}_2$ represent the parent joint positions of two links and $\text{cap}_1 = \{\mathbf{p}_1, R_1, h_1, r_1\}$ and $\text{cap}_2 = \{\mathbf{p}_2, R_2, h_2, r_2\}$ represent the surrounding capsules of each link. The augmented joint target positions can be computed numerically with $\mathbf{q}_1 - \epsilon\mathbf{v}$ where $\mathbf{v} = \mathbf{p}_2 + \frac{h_2}{2}R_2(:, 3) - \mathbf{p}_1 - \frac{h_1}{2}R_1(:, 3)$ and ϵ is a small constant resembling a step size. Using the augmented joint target positions, corresponding joint velocities can be computed using an augmented Jacobian method and subsequently used to update the model until the sum of negative signed distances becomes zero.

IV. Motion Retargeting Method

A. Background

Data-driven motion retargeting can be formulated as finding a function f that maps a mocap input vector $\mathbf{x}_{mocap}$ to an output robot pose $\mathbf{x}_{robot}$ (i.e., $f : \mathbf{x}_{mocap} \mapsto \mathbf{x}_{robot}$). In many cases, this function is found using nonparametric methods, which are statistical approaches that make no assumptions about the predictive model [19].

One such nonparametric method, locally weighted regression, which is used in this work, is described here. The method maintains N input and output pairs, $\mathcal{D}_{pair} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, and makes an output prediction of an input $\mathbf{x}_*$, by first selecting k closest inputs from $\mathcal{D}_{pair}$, then computing a weighted sum of k corresponding outputs where the weights are determined as a function of a distance measure between two inputs, $d_{\mathcal{X}}(\mathbf{x}, \mathbf{x}')$. This method, however, suffers from two major drawbacks: the necessity of collecting a sufficient number of *paired* data $\mathcal{D}_{pair}$ and the requirement of an input distance measure $d_{\mathcal{X}}(\cdot, \cdot)$. The severity of these issues often increases as input dimension increases (i.e., the curse of dimensionality).

In this paper, we advocate for the use of locally weighted regression combined with deep latent variable modeling for learning shared latent space between two different domains [2, 18]. This combination allows for semi-supervised learning as both paired and domain-specific datasets can be used for optimizing the latent space, alleviating the necessity of collecting a sufficient number of *paired* data. Furthermore, the Euclidean norm on the latent space naturally provides a distance measure. Also, the proposed method is able to offer guarantees on retargeted motion feasibility by adjusting the localness of locally weighted regression. We also present an efficient subset sampling method approximating determinantal point process (DPP) based sampling [6] guarding against the possible imbalance among datasets used for the construction of the latent space named locally approximated DPP (LA-DPP).

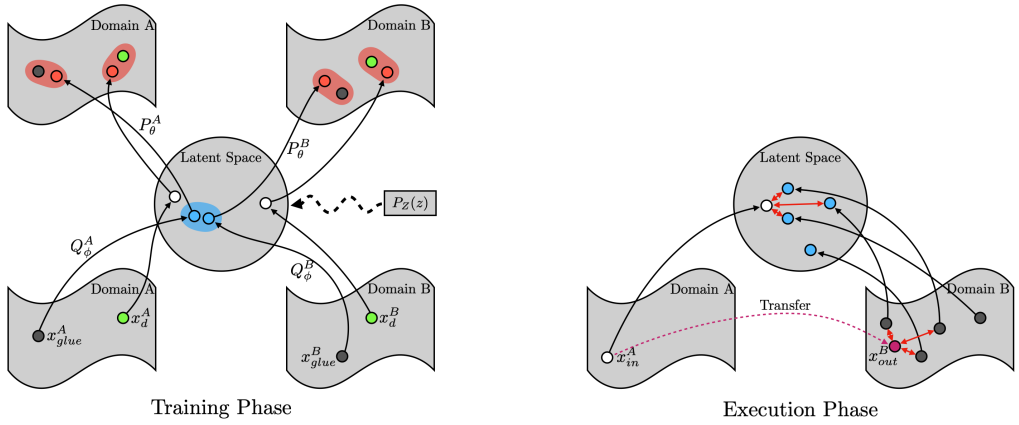

Fig. 1. Overview of the training and execution phases of the proposed domain transfer method using shared latent space modeling and locally weighted regression.

B. Shared Latent Space Modeling for Domain Transfer

One of the aims of this work is to develop a shared latent space modeling method for data-driven motion retargeting that maintains the feasibility of the transferred robot pose. Furthermore, the proposed framework leverages both paired and domain-specific datasets for training and augmentation. This is particularly beneficial as collecting domain-specific data is generally more accessible than collecting paired data[1].

In our LWL[2] method, a shared latent space is constructed between the mocap space (represented as joint positions in Cartesian coordinates) and the robot joint space (represented as joint angles) using a deep latent variable model: Wasserstein auto-encoder (WAE)[2]. While training the shared latent space, we incorporate an efficient subset sampling method to handle an imbalance in the training dataset (see Sec. IV-D) [18].

Once the shared latent space is properly optimized by learning domain-specific encoder and decoder networks - $Q_\phi(\cdot)$ and $P_\theta(\cdot)$ respectively - motion retargeting is performed by first mapping the given mocap input $\mathbf{x}_{mocap}$ to the latent space $\mathbf{z}_{mocap}$ and find the k-closest neighbor of $\mathbf{z}_{mocap}$ among all the domain-specific robot pose data $D_{robot}^{\mathbf{x}} = \{\mathbf{x}_i\}_{i=1}^{N_{robot}}$ mapped to the latent space $D_{robot}^{\mathbf{z}} = \{\mathbf{z}_i\}_{i=1}^{N_{robot}}$. Note that the robot-specific pose dataset $D_{robot}^{\mathbf{x}}$ does not necessarily need to be identical to the data used for training; thus, we can always incorporate more feasible robot pose data even after the training phase. The training and execution phases of this proposed method are illustrated in Fig. 1.

The loss functions of WAE consist of two sub-loss functions for matching the empirical encoded distribution to the latent prior distribution - a reconstruction loss $\mathcal{L}_{\text{rec}}$ and an adversarial loss $\mathcal{L}_{\text{adv}}$. Letting $p_{\mathbf{x}}$ and $p_{\mathbf{z}}$ represent the input distribution and a prior distribution over the latent space:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}}\mathbb{E}_{\mathbf{z}\sim Q_\phi(\mathbf{x})}\left[c(\mathbf{x}, P_\theta(\mathbf{z}))\right] \tag{1}$$

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}} D_{\mathbf{z}}(Q_\phi(\mathbf{x}),\, p_{\mathbf{z}}) \tag{2}$$

where $c(\cdot)$ is a distance function defined in the input space and $D_z(\cdot)$ is a divergence function defined in the latent space. We deploy l_1-norm for the distance function, $c(\cdot)$, and a Jensen-Shannon divergence for $D_{\mathbf{z}}(\cdot)$ where we use an adversarial discriminator $d_\psi(\cdot)$ in the training phase (see [18] for more details). The total loss function for WAE becomes:

$$\begin{aligned}\mathcal{L}_{\text{WAE}} = &\frac{1}{N}\sum_{i=1}^{N} c(\mathbf{x}_i, P_\theta(Q_\phi(\mathbf{x}_i))) \\ &- \frac{\beta}{2}\left[\log(\rho(d_\psi(\mathbf{z}_i))) + \log\left(1-\rho(d_\psi(Q_\phi(\mathbf{x}_i)))\right)\right]|_\psi \\ &- \beta\left[\log(\rho(d_\psi(Q_\phi(\mathbf{x}_i))))\right]|_\phi\end{aligned} \tag{3}$$

where N is the total number data, β is a tunable parameter to control the effects latent prior fitting (we default to $\beta = 1$), $\rho(\cdot)$ is a sigmoid function, and $|_\phi$ and $|_\psi$ indicate the trainable variables, e.g., $|_\phi$ indicates only ϕ is being updated while training, respectively.

In constructing shared latent space to *glue* the two domains, we use a paired dataset, which will be referred to as a glue dataset. This glue data set is fed through two loss functions to achieve the shared latent space: a latent consensus loss in (4) and a cross-domain reconstruction loss in (5). We use superscripts to denote the domain information, e.g., Q_ϕ^1 is the encoder of the first domain. The latent consensus loss is defined as

$$\mathcal{L}_{lat} = \sum_{j=1}^{m}\sum_{l=j+1}^{m}\frac{1}{N}\sum_{i=1}^{N}\|Q_\phi^j(\mathbf{x}_i^j) - Q_\phi^l(\mathbf{x}_i^l)\|_2^2 \tag{4}$$

and the cross-domain reconstruction loss is defined as

$$\mathcal{L}_{cro} = \sum_{j=1}^{m}\sum_{l=1,j\neq l}^{m}\frac{1}{N}\sum_{i=1}^{N}\|P_\theta^l(Q_\phi^j(\mathbf{x}_i^j)) - \mathbf{x}_i^l\|_2^2 \tag{5}$$

[1]In the context of motion retargeting, collecting precisely matched pairing of mocap skeleton poses and corresponding robot joint positions which could be done using optimization of joint angles and solving inverse kinematics whereas obtaining collision-free joint positions simply requires checking collision between linkages.

[2]Here, we use Wasserstein auto-encoders (WAEs) but other latent variable modeling methods such as variational auto-encoders (VAEs) [2] can also be used.

where m is the number of domains. While only two domains ($m = 2$) are being used in this paper, this method can naturally be extended to multiple domains.

Once an encoder/decoder pair is constructed for each domain, we deploy locally weighted regression on the latent space to find a mapping from one domain to the other. In particular, we vary the localness parameter k from a positive integer (e.g., 3) to 1, and check the feasibility (collision-free) and use the largest feasible k. Note that, even in the worst case, the output can always be guaranteed to be feasible by setting $k = 1$, as the proposed LWL^2 becomes a table look-up method.

C. Graph Search Heuristics and Limitation

As the shared latent space is modeled using a nonparametric method, the retargeted result can be constrained to be feasible. However, *transitioning* between these poses using simple linear interpolation (our primary method of finding transition poses) may yield infeasible/self-colliding actions. To handle such situations, we implement a fallback method which uses a graph search approach to determine feasible transitions that resembles E-Graphs [5]. We construct an undirected graph over the subset of poses D_{graph} where the connectivity of poses indicates that the linear interpolation between two poses is close enough to be reached within a single control period and no self-collision will occur.

When linearly interpolated poses between current and target poses are detected to be in self-collision (using the system described in Sec. III), we simply match the current and target poses to the closest poses in the D_{graph}. From there, the shortest path between the poses is found using the breadth-first search (BFS), and the path is traversed. A new target pose is continuously updated with the next pose in the path. Once the end of the path is reached, a linear interpolation is performed to reach the original target pose. Note that the computational complexity of BFS is $O(|V|+|E|)$ where $|V|$ is the number of poses and $|E|$ is the number of edges. The critical limitation of this graph search is that there needs to exist at least one pose in the graph that connects the current and target robot poses. If such a pose does not exist, the algorithm can fail to find a feasible transition, and no motion is performed.

D. Efficient Subset Sampling Method

In the problem of supervised learning tasks such as classification or regression, it is well-known that label imbalance may cause harmful effects on the learned model. While several resampling methods have been proposed (e.g., [23]), there have been fewer solutions to this problem concerning latent space modeling.

To demonstrate the imbalance problem for latent space modeling, we refer to an example illustrated in Fig. 2 showing synthetic latent space modeling results using WAEs. The input and latent spaces are both two-dimensional: the input domain is $[0, 5] \times [0, 5]$ and the latent prior distribution $p(\mathbf{z})$ is a two-dimensional uniform distribution between -1 and $+1$, i.e., $U[-1, +1] \times U[-1, +1]$. Each point in the training data is color-coded to visualize the data as it is encoded into the latent space (encoded data), and decoded back into the input space (reconstructed data) .

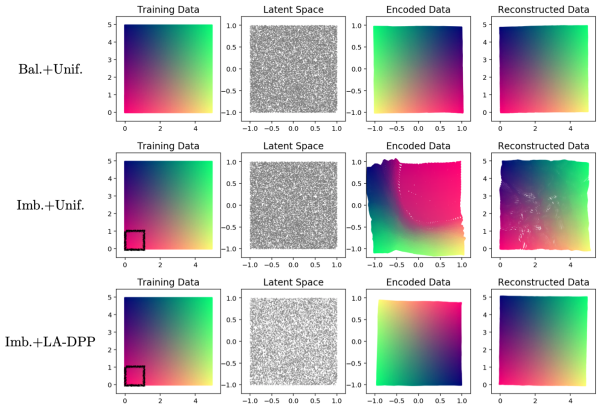

Fig. 2. Latent space modeling of a single domain with and without the proposed subsampling method.

The first row in Fig. 2 shows the case where training data is sampled uniformly from the input domain. As observed by the uniform color distribution, the resulting latent embedding is smooth. However, when the input training data is sampled unevenly, as shown in the second row, where half of the samples are obtained from a small region enclosed by the black-outlined square, the mapping becomes irregular. The majority of the latent space is mapped from the small concentrated region, as the prior fitting objective in (2) is only able to match the latent prior distribution with the empirical density of the encoded training points. However, the third row of Fig. 2 suggests that if when we apply the subsampling method which evenly subsamples from the sample set, the latent embedding becomes smooth again. Thus, to ensure good latent space generation, we require a regular subsampling method.

Such methods are usually formulated as selecting a finite subset among the whole set to maximize a certain measure of information criterion. One widely-used method is the determinantal point process (DPP) based sampling, which utilizes a determinant of a kernel matrix constructed by a subset as a measure [6]. However, one critical drawback of DPP-based algorithms is its substantial computational complexity.

Alternatively, we consider a more efficient subset sampling method. Suppose that a set of n inputs, $\{\mathbf{x}_i\}_{i=1}^n$, are given in the input space $\mathcal{X}$ equipped with a distance measure $d_{\mathcal{X}}(\cdot, \cdot)$, and a valid kernel function, $k : \mathbf{x} \times \mathbf{x} \mapsto \mathbb{R}$, is given where we deploy a commonly-used, squared exponential (SE) kernel function, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\beta d_{\mathcal{X}}(\mathbf{x}_i, \mathbf{x}_j)^2)$. Suppose that we have selected m inputs, $X_m = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m\}$, and an $m \times m$ kernel matrix K. Then, the one-step objective of DPP sampling is to find $\mathbf{x}_*$ such that the determinant of an $(m+1)\times(m+1)$ extended kernel matrix $\widetilde{W}$ constructed from $X_{m+1} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m, \mathbf{x}_*\}$ is maximized where we can rewrite $\det(\widetilde{W})$ as:

$$\det \widetilde{W} = \det \left(\begin{bmatrix} K & \mathbf{k} \\ \mathbf{k}^T & 1 \end{bmatrix} \right) \tag{6}$$

$$= \det(K - \mathbf{k}\mathbf{k}^T) \tag{7}$$

$$= (1 - \mathbf{k}^T K^{-1} \mathbf{k}) \det(K). \tag{8}$$

(6) to (7) is from a simple property of determinant and (7) to

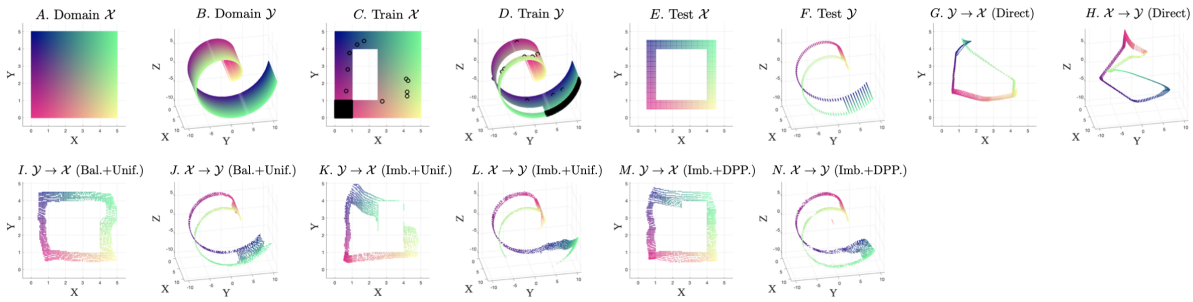

Fig. 3. Synthetic examples of finding a mapping between two and three dimensional spaces using different methods (see text for details).

(8) is a special case of $\det(A+\mathbf{u}\mathbf{v}^T) = (1+\mathbf{v}^T A^{-1}\mathbf{u})\det(A)$. Note that for selecting $\mathbf{x}_*$, K is a constant, and if we assume that $\beta \gg 1$,

$$\mathbf{x}_* = \arg\max_{\mathbf{x}\in\mathcal{X}_{\setminus m}} (1 - \mathbf{k}^T K^{-1}\mathbf{k}) \det(K) \tag{9}$$

$$\approx \arg\max_{\mathbf{x}\in\mathcal{X}_{\setminus m}} (1 - \mathbf{k}^T\mathbf{k}) \tag{10}$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}_{\setminus m}} \|\mathbf{k}\|_2^2 \tag{11}$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}_{\setminus m}} \|\mathbf{k}\|_1 \tag{12}$$

$$= \arg\min_{\mathbf{x}\in\mathcal{X}_{\setminus m}} \sum_{i=1}^{m} k(\mathbf{x}, \mathbf{x}_i). \tag{13}$$

(10), (11), (12) are derived from the fact that K is independent from $\mathbf{x}_*$, if $\beta \gg 1$ then $K \approx I_m$ where I_m is an $m \times m$ identity matrix, and the equivalence between $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively. As having $\beta \gg 1$ implies focussing on local structure, we refer to this subset sampling method as locally approximated DPP (LA-DPP) where we set $\beta = 100$ throughout the experiments. LA-DPP is used while selecting the mini-batch per each iteration.

While LA-DPP is computationally efficient compared to the original DPP sampling, it still requires constructing a $N \times N$ kernel matrix which can be prohibitive if N exceeds $50,000$. To handle the square memory complexity, we deploy a divide-and-conquer method where we first randomly shuffle the whole dataset and divide it into m subsets of sizes which are computationally tractable (e.g., $10,000$) to construct m kernel matrices. Then, the sampling procedure is performed hierarchically in that a subset is selected among the m subsets on which the proposed subset sampling process is run.

V. Experiments

In this section, we show the benefit of using the proposed subsampling method (LA-DPP) on domain transfer experiments with a synthetic dataset (in Sec. V-A) and latent space modeling of a robot called Theo[3] (in Sec. V-B). Furthermore, we apply the proposed motion retargeting method on both Theo and COMAN in simulations for quantitative comparisons with the baseline method in Sec. V-C. Finally, we demonstrate online puppeteering of a real-world humanoid, Theo, directly from mocap inputs in Sec. V-D.

Throughout the experiments, the mocap pose is represented as a 15-dimensional vector which concatenates five normalized vectors: hip to neck, right shoulder to elbow, right elbow to hand, left shoulder to elbow, and left elbow to hand. The output pose of the robots are represented as joint angles of the upper-body joints, which number 12 and 13 for COMAN and Theo, respectively. All the inputs of the neural networks are normalized to have zero-mean and unit variance.

A. Domain Transfer with Synthetic Data

Before attempting to retarget motions from mocap to robot, we wish to validate the importance of followings empirically: 1) the ability to incorporate both domain-specific datasets and a paired glue dataset, and 2) the proposed LA-DPP sampling in the case of having an imbalance within the domain-specific datasets. To this end, we conduct domain embedding experiment using synthetic datasets involving a two-dimensional domain $\mathcal{X}$ and three-dimensional domain $\mathcal{Y}$ (A and B in Fig. 3). We color-code each point to visualize the mapping between Domains $\mathcal{X}$ and $\mathcal{Y}$. Notice that the domain-specific datasets shown as diagrams C and D do not fully cover the whole domain - i.e., the blank areas have no samples. The domain-specific datasets for domain $\mathcal{X}$ and $\mathcal{Y}$ consist of $206,450$ and $42,665$ points, respectively. The glue dataset, shown as black circles in C and D, consists of 10 points to 'glue' the domains together.

The test set consists of $3,080$ corresponding points and is shown as E and F in Fig. 3. To demonstrate the benefit of using the subsampling process proposed in Sec. IV-D, we also collect imbalanced domain-specific datasets by manually replacing 50% of the points into the points that are uniformly sampled within each small region depicted by the black rectangles in C and D.

We first optimize mappings between two domains using the method described in Sec. IV with balanced and imbalanced datasets. All encoder, decoder, and discriminator networks have three layers with 256 hidden units and ReLU activations are used. As a baseline, we also train a feed-forward neural network with an $L2$ loss function using the glue dataset as it cannot leverage domain-specific datasets. The domain transfer results of the shared latent space embedding using the balanced datasets without subsampling are shown as graphs I and J in Fig. 3. We can see that they outperform those of the domain

[3]Theo consists of the upper-body of Robotis ThorMang with additional revolute joints on its both hands. It also has soft bubble padding on its exterior for safe human-robot interaction as well as alleviating the impact of self-collision.

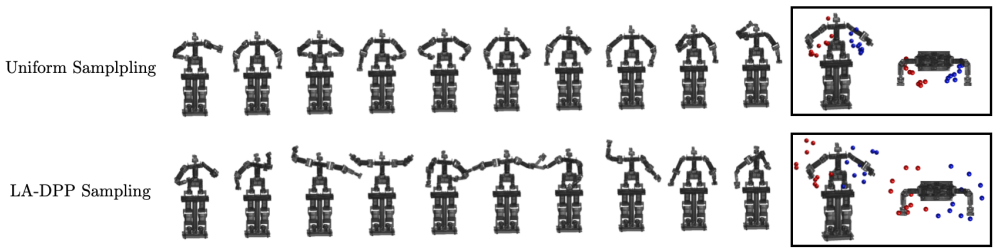

Fig. 4. Different poses that are randomly sampled from the dataset (upper row) and sampled from the proposed subsampling (lower row). The rightmost poses show the front and top views Theo with right and left hand positions shown with red and blue colors, respectively.

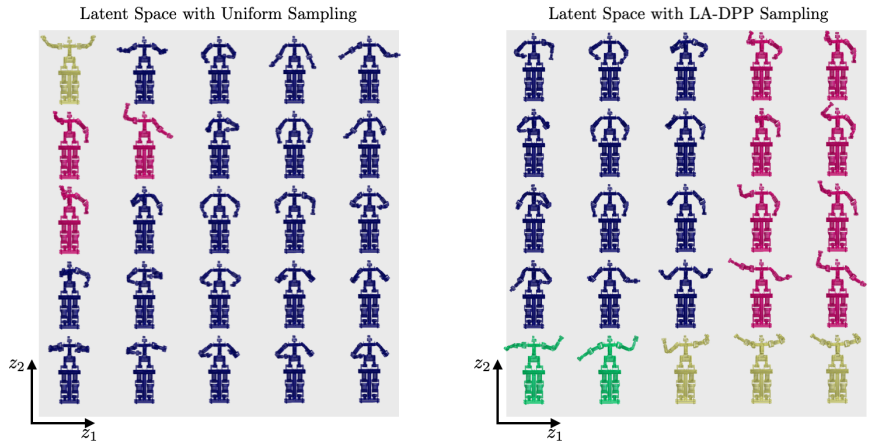

Fig. 5. Two dimensional latent space of Theo modeled without (left) and with (right) the proposed subsampling method.

transfer results of the baseline shown as graphs G and H representing a neural network that was trained on glue data only. On the other hand, When the imbalanced datasets are given, the shared latent space embedding shows suboptimal results (graphs K and L). However, when applied with the proposed subsampling process, the performances of domain transfer increase significantly (graphs M and N).

B. Latent Space of Theo

LA-DPP requires a distance measure between two robot poses. Since we are mostly interested in retargeting arm movements, we define the distance between robot poses as the sum of Euclidean distances between hand positions and rotational distances of 3D hand orientations in $SO(3)^2$. The rotational distance is defined as the norm of the difference of quaternions proposed in [24]:

$$d(\mathbf{q}_1 - \mathbf{q}_2) = \min\{\|\mathbf{q}_1 - \mathbf{q}_2\|, \|\mathbf{q}_1 + \mathbf{q}_2\|\} \qquad (14)$$

where $\mathbf{q}$ is the quaternion of a 3D rotation matrix.

Fig. 4 illustrates the poses of Theo as they are randomly sampled from the domain-specific pose dataset (first row) and via LA-DPP (second row). The diagrams to the right of the figure shows front and top views of sampled right and left-hand positions with red and blue balls. As most of the collected motions start and end with an idle pose, most of the randomly sampled poses are similar to a *stand at attention* pose, which would warp the latent space. However, the poses sampled from LA-DPP show more diverse postures.

We then train two WAEs with two-dimensional latent space with and without the subset sampling process using the same network topologies presented in Sec. V-A. The poses of Theo in two-dimensional latent spaces with and without LA-DPP sampling are shown in Fig. 5. For better visualization, We group the poses into four categories with different colors: blue for both hands are below shoulders, red (or green) when the right (or left) hand is above the shoulder, respectively, and yellow when both hands are above the shoulders. As most of the poses in the dataset are in idle poses, the latent space trained without LA-DPP sampling is mostly filled with postures that are similar to an idle pose (blue). However, when trained with LA-DPP sampling, more diverse poses are found in the latent space.

C. Motion Retargeting Experiments

Here, we conduct motion retargeting experiments using two different humanoid robots, COMAN and Theo. To collect a paired training dataset for the training phase, we leverage eight different basic motions: *ArmCross*, *BigPoint*, *BigWave*, *BowBeg*, *HandHeart*, *HeadScratch*, *HugHigh*, *NeckScratch*, collected from four different subjects. Additionally, we use seven expressive motions from the CMU mocap database [25]: *Genie*, *Monkey*, *Animal*, *Bear*, *Panda*, *SuperHero*, and *Devil*. The test dataset consists of the same eight basic motions recorded from a fifth subject and seven motions from the CMU database: *Monkey2*, *Bear*, *Penguin*, *Pterosaur*, *Dragon*, *SuperHero*, and *Robot*. All mocap poses are converted into the robot joint spaces of Theo and COMAN using an optimization-

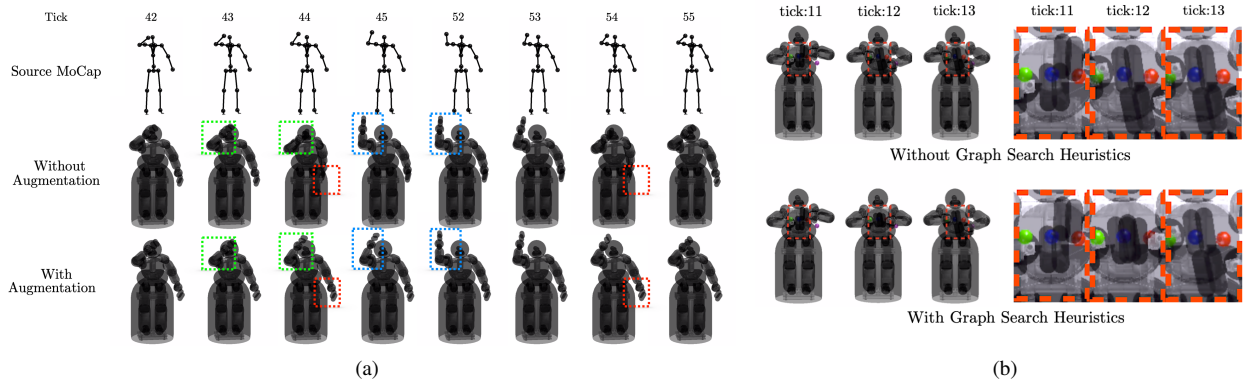

Fig. 6. (a) *BigWave* motion of Theo with and without post data augmentation. (b) The effect of graph search heuristics on Theo with *ArmCross* motion.

		COMAN [mm]		Theo [mm]	
Name	Dur (s)	LWL^2	Baseline	LWL^2	Baseline
ArmCross	8.5	79.9	**60.8**	164.7	**162.0**
BigPoint	6.5	**40.1**	42.0	101.3	**98.6**
BigWave	7.3	49.1	**44.6**	93.1	**91.7**
BowBeg	7.1	**51.0**	53.3	112.7	**104.1**
HandHeart	6.5	**41.2**	45.7	**109.0**	110.1
HeadScratch	8.8	**39.4**	41.6	108.4	**106.6**
HugHigh	6.5	52.1	**50.7**	99.0	**93.6**
NeckScratch	8.8	**37.8**	38.4	126.0	**125.9**
Monkey	15.0	**152.8**	191.2	196.7	**183.4**
Bear	15.0	**150.6**	190.1	**220.9**	236.4
Penguin	15.0	**92.7**	105.8	137.0	**129.1**
Pterosaur	15.0	**95.5**	106.5	148.7	**142.8**
Dragon	15.0	138.9	**118.3**	182.2	**168.1**
SuperHero	15.0	**77.7**	86.9	139.8	**126.0**
Robot	15.0	**86.8**	89.8	**159.6**	162.4
Average	11.0	79.0	84.4	139.9	136.0

TABLE I
TRACKING PERFORMANCES OF THE PROPOSED LWL^2 AND THE BASELINE METHOD. BOLD FONTS INDICATE BETTER RESULTS.

		COMAN [%]		Theo [%]	
Name	Dur (s)	LWL^2	Baseline	LWL^2	Baseline
ArmCross	8.5	–	39.4	–	36.5
BigPoint	6.5	–	–	–	–
BigWave	7.3	–	–	–	25.2
BowBeg	7.1	–	3.5	–	16.9
HandHeart	6.5	–	–	–	30.8
HeadScratch	8.8	–	–	–	–
HugHigh	6.5	–	–	–	–
NeckScratch	8.8	–	–	–	2.9
Monkey	15.0	–	–	–	–
Bear	15.0	–	5.0	–	56.7
Penguin	15.0	–	10.0	–	15.0
Pterosaur	15.0	–	–	–	–
Dragon	15.0	–	15.0	–	24.0
SuperHero	15.0	–	28.3	–	17.3
Robot	15.0	–	10.3	–	13.0

TABLE II
SEF-COLLISION RATE OF THE PROPOSED LWL^2 AND THE BASELINE METHOD.

based motion retargeting method presented in [11]. They are then post-processed to be collision-free using the method presented in Sec. III. These paired datasets of both mocap and robot joint angles are used as glue datasets.

We also augment the training dataset with robot domain-specific datasets. While one method to obtain such datasets is to sample feasible poses randomly, this approach often generates peculiar poses. Instead, we randomly select poses within the dataset and then arbitrarily perturb randomly selected joints within their position limits, while ensuring the result is still feasible. Using this augmentation method, we collect additional $200,000$ poses for domain-specific datasets to be used in the execution phase after training. All encoder, decoder, and discriminator networks are three-layer ResNet structures with hyperbolic tangent activations. The LA-DPP sampling method is used on mini-batches during the training phase; the graph search heuristics presented in Sec. IV-C are used in the execution phase with $10,000$ nodes sampled with LA-DPP within the domain-specific pose dataset. For a baseline comparison, we deploy associate latent encoding (ALE) [3] which also learns the domain embedding via shared latent space using VAEs.

Fig. 6(a) demonstrates the benefits of using additional augmented domain-specific datasets: *BigWave* motions of Theo with and without the augmentation show that the augmentation increases the smoothness of behaviors (see dotted-line boxes). Comparisons between using and not using the graph search heuristics are shown in Fig. 6(b). Here, graph search heuristics exhibit smoother and feasible transitioning between robot poses during an arm over arm rotation (see supplementary video). In the transition without graph search, the robot exhibits considerable pose changes that is likely to cause self-collision as shown in red dotted-line boxes. Note that all independent poses are collision-free. Fig. 7 and 8 show mocap skeletons of *HandHeart* and *SuperHero* motions and retargeted robot poses of Theo and COMAN, respectively, where red capsules indicate self-collision occurrences. Furthermore, as depicted by red dotted-line boxes in Fig. 8, LWL^2 often shows better retargeting results that the baseline.

Table I and II summarize tracking performances and collision rates of the proposed method and the baseline where the tracking performance is measured by the average discrepancies between the current and target positions of hand, elbow, and shoulder in the Cartesian space. The proposed method outperforms the baseline in terms of safety and shows comparable tracking performance. We would like to emphasize that no collisions were made when using LWL^2 for test motions which

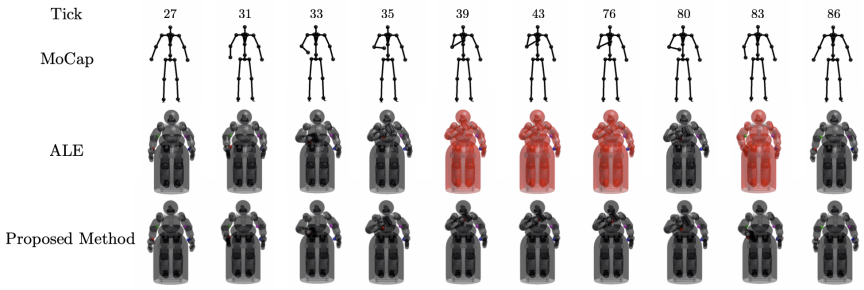

Fig. 7. *HandHeart* motion of Theo using the baseline (ALE) and the proposed method. The red colored meshes indicate collision has occurred.

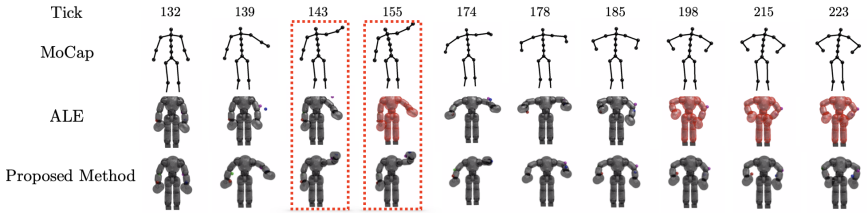

Fig. 8. *SuperHero* motion of COMAN using the baseline (ALE) and the proposed method. The red colored meshes indicate collision has occurred.

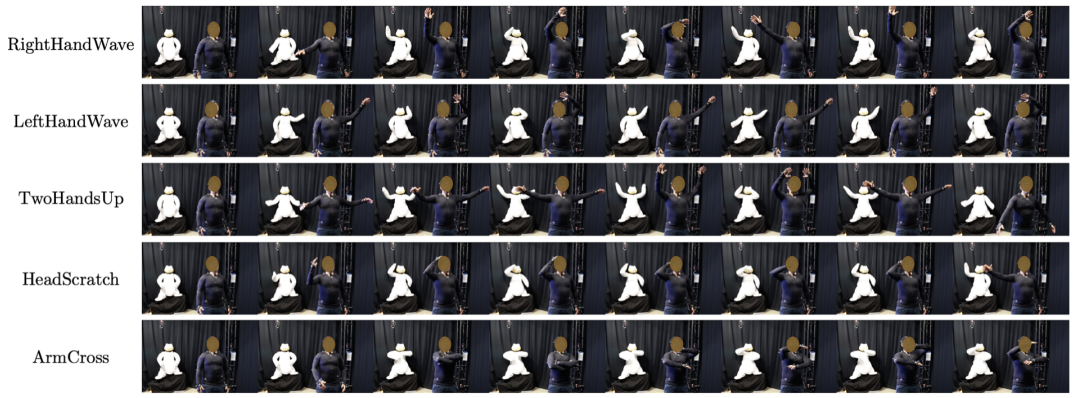

Fig. 9. Snapshots of online puppeteering of Theo on different motions.

allows us to deploy it to the online puppeteering of Theo.

D. Real-Time Puppeteering of Theo

We demonstrate online puppeteering of Theo from OptiTrack mocap systems where the same mocap input representations and networks trained in Sec. V-C are used. The resulting motions of Theo from the puppeteering are shown in Fig. 9. We conduct online puppeteering with five different motions, *RightHandWave*, *LeftHandWave*, *TwoHandsUp*, *HeadScratch*, and *ArmCross* where we can see that the resulting motions of Theo are collision-free despite having human motions involve self-contact (e.g., *HeadScratch*, and *ArmCross*).

While the poses of Theo generated from the *RightHandWave* motion show acceptable performance, we observe that the some poses from the *LeftHandWave* motion contain right arm movements (see *LeftHandWave*). We hypothesize that this is due to the lack of diverse motions in the training dataset implying that the current training dataset is not sufficient enough to cover all possible mocap and robot pose spaces. This is rectifiable via more training data or augmentation.

VI. Conclusion

In this paper, we have presented a data-driven motion retargeting method that helps ensure the feasibility of the generated motions. To this end, we use a nonparametric method (locally weighted regression) combined with recently developed deep latent space modeling to incorporate both paired and domain-specific datasets. We show that domain-specific datasets not used during training can be utilized to augment retargeting performance. Also, we propose graph-search heuristics to facilitate the transition between poses.

While the proposed method aims to resolve some of the issues in existing motion retargeting methods, there still exists an explicit limitation in that we have to collect a sufficient number of poses to guarantee the smoothness of the resulting motion. One can adopt online learning of re-using the feasible robot positions enjoying the flexibility of nonparametric methods in adapting new samples. Self-supervised learning of motion retargeting to alleviate the data collection process can also be an exciting future research direction.

References

[1] Nancy S Pollard, Jessica K Hodgins, Marcia J Riley, and Christopher G Atkeson. Adapting human motion for the control of a humanoid robot. In *IEEE Proc. of International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1390–1397. IEEE, 2002.

[2] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

[3] Hang Yin, Francisco S Melo, Aude Billard, and Ana Paiva. Associate latent encodings in learning from demonstrations. In *AAAI Conference on Artificial Intelligence*, 2017.

[4] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning. In *Lazy learning*, pages 11–73. Springer, 1997.

[5] Mike Phillips, Benjamin J Cohen, Sachin Chitta, and Maxim Likhachev. E-graphs: Bootstrapping planning with experience graphs. In *Robotics: Science and Systems*, volume 5, page 110, 2012.

[6] Alex Kulesza and Ben Taskar. k-DPPs: Fixed-size determinantal point processes. In *Proc. of the International Conference on Machine Learning*, pages 1193–1200, 2011.

[7] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42. ACM, 1998.

[8] Katsu Yamane and Jessica Hodgins. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2510–2517. IEEE, 2009.

[9] Katsu Yamane, Stuart O Anderson, and Jessica K Hodgins. Controlling humanoid robots with human motion data: Experimental validation. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 504–510. IEEE, 2010.

[10] L Peneo, Brice Clément, V Moduano, E Mingo Hoffman, Gabriele Nava, Daniele Pucci, Nikos G Tsagarakis, J-B Mourert, and Serena Ivaldi. Robust real-time whole-body motion retargeting from human to humanoid. In *Proc. of the IEEE International Conference on Humanoid Robots*, pages 425–432. IEEE, 2018.

[11] Sungjoon Choi and Joohyung Kim. Towards a natural motion generator: a pipeline to control a humanoid based on motion data. In *Proc. ot IEEE/RSJ International Conference on Intelligent Robots and Systems Intelligent Robots and Systems*, 2019.

[12] Neil D Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. In *Advances in neural information processing systems*, pages 329–336, 2004.

[13] Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh P Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1233–1240, 2006.

[14] Sergey Levine, Jack M Wang, Alexis Haraux, Zoran Popović, and Vladlen Koltun. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (TOG)*, 31(4):28, 2012.

[15] Zhongyue Huang, Jingwei Xu, and Bingbing Ni. Human motion generation via cross-space constrained sampling. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 757–763, 2018.

[16] Katsu Yamane, Yuka Ariki, and Jessica Hodgins. Animating non-humanoid characters with human motion data. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 169–178. Eurographics Association, 2010.

[17] Stefan Schaal, Christopher G Atkeson, and Sethu Vijayakumar. Scalable techniques from nonparametric statistics for real time robot learning. *Applied Intelligence*, 17(1):49–60, 2002.

[18] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Ryan Tibshirani and Larry Wasserman. Nonparametric regression. *Statistical Machine Learning, Spring*, 2013.

[20] Chiara Fulgenzi, Christopher Tay, Anne Spalanzani, and Christian Laugier. Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes. In *Proc. of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[21] Markus Schneider and Wolfgang Ertel. Robot learning by demonstration with local gaussian process regression. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 255–260. IEEE, 2010.

[22] Sungjoon Choi, Kyungjae Lee, and Songhwai Oh. Robust learning from demonstration using leveraged Gaussian processes and sparse constrained opimization. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2016.

[23] Francisco Charte, Antonio J Rivera, María J del Jesus, and Francisco Herrera. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing*, 163:3–16, 2015.

[24] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[25] CMU graphics lab motion capture database. URL `http://mocap.cs.cmu.edu/`.