

# Incremental Learning of Full-Pose Via-Point Movement Primitives on Riemannian Manifolds

Tilman Daab, Noémie Jaquier, Christian Dreher, André Meixner, Franziska Krebs, and Tamim Asfour

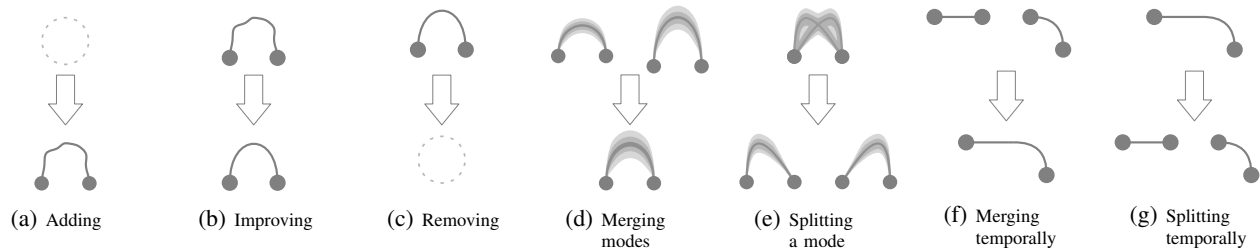


Fig. 1: Proposed fundamental operations to incrementally learn MP libraries. We provide formulations of the spatial operations (a)–(e) for VMPs.

**Abstract**—Movement primitives (MPs) are compact representations of robot skills that can be learned from demonstrations and combined into complex behaviors. However, merely equipping robots with a fixed set of innate MPs is insufficient to deploy them in dynamic and unpredictable environments. Instead, the full potential of MPs remains to be attained via adaptable, large-scale MP libraries. In this paper, we propose a set of seven fundamental operations to incrementally learn, improve, and re-organize MP libraries. To showcase their applicability, we provide explicit formulations of the spatial operations for libraries composed of Via-Point Movement Primitives (VMPs). By building on Riemannian manifold theory, our approach enables the incremental learning of all parameters of position and orientation VMPs within a library. Moreover, our approach stores a fixed number of parameters, thus complying with the essential principles of incremental learning. We evaluate our approach to incrementally learn a VMP library from motion capture data provided sequentially.

## I. INTRODUCTION

Robots providing help to humans at home, taking care of the elderly, or collaborating with workers must be able to learn new skills and to adapt them to unseen situations. In this context, learning from demonstrations [1] and imitation learning [2] are promising frameworks to learn generalizable skills from human demonstrations. Such skills are often represented in the form of movement primitives (MPs) [3], [2], and may be stored to form a MP library [2], [4]. Learning and adapting skills thus essentially entails incrementally updating the library. Different definitions of incremental learning exist in the literature, see, e.g., [5], [6] for an overview. Akin to Gepperth and Hammer [5], we consider *online learning* as learning from sequentially-arriving data, and *incremental learning* as online learning with limited amount of memory.

The research leading to these results has received funding from the European Union’s Horizon 2021 Research and Innovation programme under grant agreement No 101070292 (HARIA) and from the Carl Zeiss Foundation through the JuBot project.

The authors are with the Institute for Anthropomatics and Robotics, High Performance Humanoid Technologies Lab (H2T), at the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany. {tilman.daab, asfour}@kit.edu

Specifically, we require the memory for learning  $m$  models from  $n$  samples with  $m < n$  to be bound by  $\mathcal{O}(f(m))$ , i.e., independently of  $n$ . This implies that approaches requiring to permanently store all samples or individually-retrieved sample representations are not considered as incremental.

In this paper, we tackle the problem of incrementally learning a MP library. Specifically, we argue that this goes beyond adding and improving existing MPs and introduce seven spatial and temporal operations to extend, update, and re-organize MP libraries based on incoming new demonstration examples (see Figure 1 and Section IV). To showcase their applicability, we provide an explicit formulation of the spatial operations for a library of Via-Point Movement Primitives (VMPs) [7] as MPs. Complex robotic skills also require to handle full-pose (i.e., positions and orientations) end-effector trajectories. Therefore, we provide methods to incrementally adapt all parameters of full-pose VMPs using the fundamental spatial operations (Section V). By building on Riemannian manifolds theory (Section V-A), our approach soundly treats full-pose trajectories, while adhering to the aforementioned incremental learning definition.

The contributions of this paper are threefold: (i) We introduce a set of fundamental operations to incrementally learn MP libraries; (ii) We provide formulations to incrementally learn all parameters (i.e., weights, via-points, and task parameters) of the VMPs within a library according to the fundamental spatial operations; and (iii) in doing so, we provide a Riemannian formulation of VMPs to handle full-pose trajectories. We showcase our approach by incrementally learning a VMP library from human demonstrations recorded via motion capture.

## II. RELATED WORK

In the following, we review existing works in the field of incremental learning of MPs. The influential works [8], [9], [10] are examples of online learning of MPs. Kulić et al. [8] learn MPs from dynamically-clustered demonstrations. This allows adding, improving, and splitting modes of MPs. How-

ever, this approach is not incremental as it permanently stores a model of each demonstration. Niekum et al. [9], [10] learn dynamical movement primitives (DMPs) [11] and represent tasks as graphs. The model can be changed arbitrarily by re-creating it, but not via specific operations. The approach is not incremental as all demonstrations are stored. Gutierrez et al. [12], [13] proposed to add and improve MPs online based on HMMs.

Existing incremental approaches often only support a single operation to modify models in a MP library. Adding was formulated in [14] for DMPs. Improving was formulated for periodic DMPs [15], for MPs based on Gaussian Mixture Models (GMMs) [16], for Dynamical Systems (DS) [17], for HMMs [18], and for expected sensor data during the execution of DMPs [19]. Adding, improving and removing as three operations were formulated for DMPs specifically in 2D positional space [20]. In this paper, we instead propose seven fundamental operations of incrementally learning a MP library and formulate the five spatial ones within a single framework.

### III. BACKGROUND

In this section, we introduce the MP formulation on which we focus in this paper, as well as the mathematical tools that are required to handle full-pose demonstrations.

#### A. Via-Point Movement Primitives

Via-Point Movement Primitives (VMPs) [7] are highly flexible MPs that combine the extrapolation capabilities of DMPs [11] with the ability of probabilistic movement primitives (ProMPs) [21] to handle via-points. A VMP models a trajectory  $\mathbf{y}(\varphi)$  as the superposition of an elementary trajectory  $\mathbf{h}(\varphi)$  and a shape modulation  $\mathbf{f}(\varphi)$ , i.e.,

$$\mathbf{y}(\varphi) = \mathbf{h}(\varphi) + \mathbf{f}(\varphi), \quad (1)$$

where  $\varphi \in [0, 1]$  denotes the phase variable allowing for temporal modulation. The elementary trajectory forms the basic structure of the VMP by connecting two consecutive points such as a start  $\mathbf{h}_0$  and a goal  $\mathbf{h}_1$ . Without loss of generality, we consider linear elementary trajectories

$$\mathbf{h}(\varphi) = \mathbf{h}_0 + \varphi \cdot (\mathbf{h}_1 - \mathbf{h}_0). \quad (2)$$

The shape modulation is represented as a weighted sum of basis functions, so that

$$\mathbf{f}(\varphi) = \Psi(\varphi)\mathbf{w}, \quad (3)$$

where  $\Psi(\varphi) \in \mathbb{R}^{d \times dk}$  is an activation matrix computed as a block-diagonal matrix of radial basis functions (RBFs)  $\psi(\varphi)^T \in \mathbb{R}^{1 \times k}$ , and  $\mathbf{w} = [\mathbf{w}_1^T, \dots, \mathbf{w}_k^T]^T \in \mathbb{R}^{dk}$  is a concatenated weight vector. As in ProMPs,  $\mathbf{w}$  follows a Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ , whose parameters are computed from the weight vectors estimated for each demonstration. After the initial learning phase,  $\mathbf{f}(\varphi)$  can be adapted via the conditioning of  $\mathbf{w}$ . Alternatively, and as preferred by Zhou et al. [7] for orientations, adaptations can be achieved by modulating the elementary trajectory using a via-point.

Given a desired pose  $\mathbf{y}_v = \mathbf{y}(\varphi_v)$  associated with a phase value  $\varphi_v$ , the pose of an elementary via-point is

$$\mathbf{h}_v = \mathbf{y}_v - \mathbf{f}(\varphi_v). \quad (4)$$

With  $v^-$  and  $v^+$  denoting the surrounding via-points of a phase, the elementary trajectory (2) becomes

$$\mathbf{h}(\varphi) = \mathbf{h}_{v^-} + \frac{\varphi - \varphi_{v^-}}{\varphi_{v^+} - \varphi_{v^-}} \cdot (\mathbf{h}_{v^+} - \mathbf{h}_{v^-}).$$

Note that the start and end of the VMP are simply considered as via-points for  $\varphi = 0$  and  $\varphi = 1$ , respectively.

Since quaternions must satisfy a unit-norm constraint, additional care must be taken to formulate orientation VMPs. Zhou et al. [7] defined the elementary trajectory via spherical linear interpolation. Trajectories  $\mathbf{y}(\varphi)$  were then modeled via quaternion multiplication of  $\mathbf{h}(\varphi)$  with a shape modulation  $\mathbf{f}(\varphi)$ . The vector part of  $\mathbf{f}(\varphi)$  was learned using (3), and its scalar part deduced to respect the unit-norm constraint. However, such normalization operations are known to lead to inaccurate models [22], [23]. In this paper, we instead account for the intrinsic geometry of quaternions and formulate a Riemannian approach to incrementally learn full-pose VMPs.

#### B. Riemannian Manifolds

In this paper, we represent orientations via unit quaternions, which are widely-used nearly-minimal representations. As quaternions must satisfy a unit-norm constraint, they cannot be treated with traditional Euclidean methods. Instead, we leverage tools from Riemannian geometry to learn orientation trajectories. A smooth manifold  $\mathcal{M}$  is a curved topological space endowed with a smooth differential structure [24]. A tangent space  $\mathcal{T}_x\mathcal{M}$ , i.e., a Euclidean approximation of the manifold, is associated with each point  $\mathbf{x} \in \mathcal{M}$ . A Riemannian manifold is a smooth manifold equipped with a Riemannian metric, i.e., a smoothly-varying positive-definite inner product acting on each tangent space  $\mathcal{T}_x\mathcal{M}$  [25]. The Riemannian metric defines geodesics as shortest paths between two points on  $\mathcal{M}$ , thus generalizing the notion of straight lines to Riemannian manifolds. Euclidean tangent spaces can be leveraged via the use of the exponential map  $\text{Exp}_x(\mathbf{u}) : \mathcal{T}_x\mathcal{M} \rightarrow \mathcal{M}$  and logarithmic map  $\text{Log}_x(\mathbf{y}) : \mathcal{M} \rightarrow \mathcal{T}_x\mathcal{M}$ . Finally, the parallel transport operates  $\Gamma_{\mathbf{x} \rightarrow \mathbf{y}}(\mathbf{u}) : \mathcal{T}_x\mathcal{M} \rightarrow \mathcal{T}_y\mathcal{M}$  with tangent vectors lying on different tangent spaces. Full-pose trajectories are composed of points on the product of manifold  $\mathcal{M} = \mathbb{R}^3 \times \mathcal{S}^3$ , where positions and quaternions are elements of the Euclidean space  $\mathbb{R}^3$  and sphere manifold  $\mathcal{S}^3$ , respectively.

### IV. OPERATIONS TO INCREMENTALLY LEARN MOVEMENT PRIMITIVE LIBRARIES

As stated in Section I, learning a movement primitive library incrementally goes beyond adding or incrementally improving single MPs. Here, we propose to structure the incremental learning capabilities of MP libraries into seven fundamental operations (see Figure 1). These operations not only incorporate new knowledge but also provide the ability

to revert operations that were taken erroneously. This is crucial, as systems that decide which operation to perform hardly work perfectly. The three first operations are:

- (a) *Adding* a MP to the library to extend it. This is needed if a new observation does not match any existing MP in the library, e. g., as in [14].
- (b) *Improving* an existing MP. This allows generalization based on multiple demonstrations, e. g., as in [15].
- (c) *Removing* a MP that is not required anymore, e. g., as in [20], or was added erroneously, e. g., as a consequence of perception problems.

In the following, we refer to multiple distinct ways to perform a movement as *modes*. The last four operations describe ways of re-organizing a MP library, namely:

- (d) *Merging modes* to merge two separately-stored MPs representing similar movements into a single one.
- (e) *Splitting a mode* into two distinct ones, each of them represented by a single MP, e. g., as (online) in [8]. This is useful when a MP consists of several modes that were erroneously learned as a single one.
- (f) *Merging temporally* several MPs that always occur together or correspond to a movement that was over-segmented into multiple MPs.
- (g) *Splitting temporally* a longer MP which, inversely, consists of several movements that could be used separately.

Given these fundamental operations, key challenges are how, when, and which of them to apply within MP libraries. We start by tackling the first of these challenges and formulate the five spatial operations (a)–(e) in a specific MP library.

## V. INCREMENTAL LEARNING OF FULL-POSE VIA-POINT MOVEMENT PRIMITIVES

In this section, we specifically address the problem of incrementally learning a library of Via-Point Movement Primitives (VMPs) [7]. As many robot skills are composed of position and orientation, we first introduce a Riemannian formulation for full-pose VMPs representing trajectories  $\mathbf{y} \in \mathbb{R}^3 \times \mathcal{S}^3$ . Then, we provide methods to incrementally learn full-pose VMPs within our library via the 5 spatial operations (a)–(e) introduced in the previous section. In this paper, we assume that demonstrations are provided with a perfect segmentation. Thus, temporal operations will be addressed as future work.

### A. Full-Pose VMPs

Defining  $\mathcal{M} = \mathbb{R}^3 \times \mathcal{S}^3$ , the VMP basic principle translates to the Riemannian case as follows. The trajectory  $\mathbf{y}(\varphi) \in \mathcal{M}$  is composed by an elementary trajectory  $\mathbf{h}(\varphi) \in \mathcal{M}$ , which is modified by a shape modulation  $\mathbf{f}_{\mathbf{h}(\varphi)}(\varphi) \in \mathcal{T}_{\mathbf{h}(\varphi)}\mathcal{M}$  as

$$\mathbf{y}(\varphi) = \text{Exp}_{\mathbf{h}(\varphi)}(\mathbf{f}_{\mathbf{h}(\varphi)}(\varphi)), \quad (5)$$

with phase variable  $\varphi \in [0, 1]$ . The elementary trajectory connects a start  $\mathbf{h}_0$  to a goal  $\mathbf{h}_1$  via a geodesic

$$\mathbf{h}(\varphi) = \text{Exp}_{\mathbf{h}_0}(\varphi \cdot \text{Log}_{\mathbf{h}_0}(\mathbf{h}_1)). \quad (6)$$

Note that Eqs. (5) and (6) simplify to (1) and (2), respectively, when  $\mathcal{M} = \mathbb{R}^d$ , i.e., we recover the Euclidean VMP.

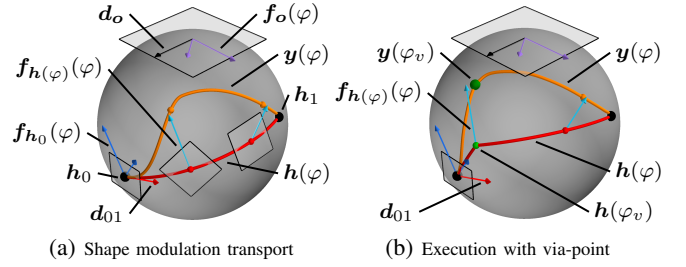


Fig. 2: Illustration of a Riemannian VMP on  $\mathcal{S}^2$ .

Importantly, the shape modulation is an element of the tangent space at the elementary trajectory. Thus, it follows a Euclidean structure and can be defined similarly as in (3) with special care of defining common weight vectors across the different tangent spaces  $\mathcal{T}_{\mathbf{h}(\varphi)}\mathcal{M}$ . To do so, we define the weights as  $\mathbf{w}_{\mathbf{h}_0} \in \mathcal{T}_{\mathbf{h}_0}\mathcal{M}$  and obtain  $\mathbf{f}_{\mathbf{h}(\varphi)}$  as

$$\mathbf{f}_{\mathbf{h}(\varphi)}(\varphi) = \Gamma_{\mathbf{h}_0 \rightarrow \mathbf{h}(\varphi)}(\mathbf{f}_{\mathbf{h}_0}(\varphi)), \quad (7)$$

with  $\mathbf{f}_{\mathbf{h}_0}(\varphi) = \Psi(\varphi)\mathbf{w}_{\mathbf{h}_0}$ . Note that parallel transport conserves the angle between shape modulation and the direction of the elementary trajectory.

Demonstrations characterized by different weight vectors  $\mathbf{w}_{\mathbf{h}_0, n}$  lead to a distribution  $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$ . The computation of this distribution is hindered by the fact that each  $\mathbf{w}_{\mathbf{h}_0, n}$  lies on a different tangent space  $\mathcal{T}_{\mathbf{h}_0, n}\mathcal{M}$ . This is resolved by transporting all  $\mathbf{f}_{\mathbf{h}_0}$  to a common tangent space  $\mathcal{T}_{\mathbf{o}}\mathcal{M}$  defined, e.g., at the origin  $\mathbf{o}$  of  $\mathcal{M}$ . However, in this case, parallel transport does not suffice to ensure consistency between shape modulation vectors in  $\mathcal{T}_{\mathbf{o}}\mathcal{M}$ . Specifically, two identical trajectories translated on the manifold would be encoded in  $\mathcal{T}_{\mathbf{o}}\mathcal{M}$  by shape modulation vectors differing by a rotation depending on their starting points. We compensate this by aligning the parallel-transported direction  $\mathbf{d}_{01} = \text{Log}_{\mathbf{h}_0}(\mathbf{h}_1) / \|\text{Log}_{\mathbf{h}_0}(\mathbf{h}_1)\|_{\mathbf{h}_0} \in \mathcal{T}_{\mathbf{h}_0}\mathcal{M}$  of each demonstration with a default unit direction  $\mathbf{d}_{\mathbf{o}} \in \mathcal{T}_{\mathbf{o}}\mathcal{M}$ . This is achieved via a rotation  $\mathbf{R}$  satisfying

$$\mathbf{d}_{\mathbf{o}} = \mathbf{R}\Gamma_{\mathbf{h}_0 \rightarrow \mathbf{o}}(\mathbf{d}_{01}). \quad (8)$$

The shape modulation vectors are then obtained as

$$\mathbf{f}_{\mathbf{o}} = \mathbf{R}\Gamma_{\mathbf{h}_0 \rightarrow \mathbf{o}}(\mathbf{f}_{\mathbf{h}_0}).$$

Figure 2a illustrates the transport for two shape modulations. If  $\mathbf{h}(0) = \mathbf{h}(1)$ ,  $\mathbf{d}_{01}$  is ill-defined, and the rotation is omitted. Notice that, in general,  $\mathbf{y}_0 \approx \mathbf{h}_0$  and  $\mathbf{y}_1 \approx \mathbf{h}_1$  hold aside from approximation errors of  $\mathbf{f}(0)$  and  $\mathbf{f}(1)$ .

Similarly as Euclidean VMPs, adaptation of Riemannian VMPs can be achieved by conditioning  $\mathbf{w}$  or by modulating the elementary trajectory with via-points. Analogously to Eq. (4) in Euclidean spaces, elementary via-points  $\mathbf{h}_v$  at  $\varphi_v$  are derived from trajectory via-points  $\mathbf{y}_v = \mathbf{y}(\varphi_v)$  so that

$$\mathbf{y}_v = \text{Exp}_{\mathbf{h}_v}(\mathbf{f}_{\mathbf{h}_v}(\varphi_v)) = \text{Exp}_{\mathbf{h}_v}(\Gamma_{\mathbf{h}_0 \rightarrow \mathbf{h}_v}(\mathbf{f}_{\mathbf{h}_0}(\varphi_v))). \quad (9)$$

In contrast to the Euclidean case, Eq. (9) does not yield an analytical solution. Instead, it resembles a geodesic regression problem [26] and a solution  $\mathbf{h}_v$  can be calculated by minimizing the distance  $d_{\mathcal{M}}^2(\mathbf{y}_v, \tilde{\mathbf{y}}_v)$  between the desired

pose  $\tilde{\mathbf{y}}_v$  and the pose (9). This is achieved via the Riemannian gradient descent update

$$\mathbf{h}_{v,i+1} \leftarrow \text{Exp}_{\mathbf{h}_{v,i}} \left( 2\alpha \Gamma_{\mathbf{y}_{v,i} \rightarrow \mathbf{h}_{v,i}} \left( \text{Log}_{\mathbf{y}_{v,i}} (\tilde{\mathbf{y}}_v) \right) \right),$$

where we use a gradient approximation akin to and the adaptive stepsize  $\alpha$  of [27]. Finally, Eq. (6) becomes

$$\mathbf{h}(\varphi) = \text{Exp}_{\mathbf{h}_{v-}} \left( \frac{\varphi - \varphi_{v-}}{\varphi_{v+} - \varphi_{v-}} \cdot \text{Log}_{\mathbf{h}_{v-}} (\mathbf{h}_{v+}) \right).$$

Note that  $\mathbf{R}$  is still defined as a function of the direction  $\mathbf{d}_{01}$  to avoid discontinuities in  $\mathbf{y}(\varphi)$  in the presence of via-points. Figure 2b illustrates the execution of a VMP with a via-point. Next, we show how to apply the fundamental spatial operations of Section IV to incrementally learn the weights, via-points, and task parameters of full-pose VMPs.

### B. Incremental Learning of VMP Weights

Following the definition of Section I, incremental learning of full-pose VMPs is achieved by storing only a fixed amount of parameters while demonstrations are provided sequentially. In this section, we present incremental approaches to learn the weights  $\mathbf{w}$  characterizing incrementally-learned VMPs. The building blocks  $\mathbf{w}_i$  of the concatenated weight vector  $\mathbf{w}$  lie in tangent spaces of  $\mathbb{R}^3 \times \mathcal{S}^3$  and thus display a Euclidean structure. Therefore,  $\mathbf{w}$  is learned incrementally as the Euclidean distribution  $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \subset \mathbb{R}^{7k}$ .

Given  $N$  samples  $\{\mathbf{x}_n\}_{n=1}^N$ , a Gaussian distribution  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathbb{R}^d$  is estimated batch-wise following

$$\hat{\boldsymbol{\mu}}_n = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad \text{and} \quad \hat{\boldsymbol{\Sigma}}_n = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_n)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_n)^\top, \quad (10)$$

with  $\hat{\boldsymbol{\Sigma}}_n$  being defined if  $n \geq 2$ . Internally, we represent the estimated Gaussian distribution with three parameters: The number of samples  $n$  and the two estimated non-central moments  $\hat{\boldsymbol{\mu}}_n = \mathbb{E}(X)$  and  $\hat{\mathbf{S}}_n = \mathbb{E}(XX^\top)$ . Thus, we have

$$\hat{\boldsymbol{\Sigma}}_n = \frac{n}{n-1} \left( \hat{\mathbf{S}}_n - \hat{\boldsymbol{\mu}}_n \hat{\boldsymbol{\mu}}_n^\top \right). \quad (11)$$

The weight distribution parameters are incrementally learned to achieve the five spatial operations of Section IV as follows.

- (a) *Adding* a VMP to the library is realized by creating a weight estimator. Based on a new demonstration  $\mathbf{x}_1$  it is initialized with  $n = 1$ ,  $\hat{\boldsymbol{\mu}}_1 = \mathbf{x}_1$  and  $\hat{\mathbf{S}}_1 = \mathbf{x}_1 \mathbf{x}_1^\top$ .
- (b) *Improving* a weight estimation based on an additional demonstration  $\mathbf{x}_{n+1}$  is achieved as

$$\hat{\boldsymbol{\mu}}_{n+1} = \frac{n\hat{\boldsymbol{\mu}}_n + \mathbf{x}_{n+1}}{n+1} \quad \text{and} \quad \hat{\mathbf{S}}_{n+1} = \frac{n\hat{\mathbf{S}}_n + \mathbf{x}_{n+1}\mathbf{x}_{n+1}^\top}{n+1}.$$

- (c) *Removing* a VMP simply corresponds to deleting its weight estimator and potential references in task models.
- (d) *Merging two modes* A and B results in a joint mode C, whose weight parameters are, with  $n_C = n_A + n_B$ ,

$$\hat{\boldsymbol{\mu}}_C = \frac{n_A}{n_C} \hat{\boldsymbol{\mu}}_A + \frac{n_B}{n_C} \hat{\boldsymbol{\mu}}_B \quad \text{and} \quad \hat{\mathbf{S}}_C = \frac{n_A}{n_C} \hat{\mathbf{S}}_A + \frac{n_B}{n_C} \hat{\mathbf{S}}_B. \quad (12)$$

- (e) *Splitting a mode* C into two modes A and B using a demonstration  $\mathbf{x}$  requires additional assumptions, as

there are more variables than constraints. We assume (i) the demonstration to be similar to one of the modes,  $\hat{\boldsymbol{\mu}}_A = \mathbf{x}$ , and (ii) both modes to have contributed equally to the existing estimation, i.e.,  $n_A = n_B = \frac{n_C}{2}$ . Then, Eq. (12) leads to  $\hat{\boldsymbol{\mu}}_B = 2\hat{\boldsymbol{\mu}}_C - \hat{\boldsymbol{\mu}}_A$ . While it might be tempting to fix  $\hat{\mathbf{S}}_A = \mathbf{x}\mathbf{x}^\top$  and derive  $\hat{\mathbf{S}}_B$  using Eq. (12), this always leads to  $\hat{\boldsymbol{\Sigma}}_A$  being degenerated, and can even lead to  $\hat{\boldsymbol{\Sigma}}_B$  not being positive semi-definite. To avoid this, we further assume  $\hat{\boldsymbol{\Sigma}}_A = \hat{\boldsymbol{\Sigma}}_B = \sigma^2 \mathbf{I}$  with  $\sigma = \frac{1}{3} \|\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B\|$ .  $\hat{\mathbf{S}}_A$  and  $\hat{\mathbf{S}}_B$  follow using Eq. (11). To speed up convergence during subsequent improvements, we finally reduce  $n$  for each mode by a factor of  $\frac{1}{2}$ .

Importantly, mathematical equivalence with batch-wise estimations holds for all operations except *splitting a mode*. As such, they offer the benefits of incremental learning without introducing drawbacks compared to batch-wise learning.

### C. Detection of Via-Points

Demonstrations with undetected via-points lead to distorted weight estimations as via-points distort the trajectory. As a counteraction, we tackle the detection of via-points in this section. Their incremental generalization is addressed as a part of the task parameter estimation in Section V-D. An experimental comparison of the approaches follows in Section VI-B. Here, we denote the reproduction of a demonstration  $\{\tilde{\mathbf{y}}_n\}_{n=1}^N$  with a VMP as  $\{\mathbf{y}(\varphi_n) = \mathbf{y}_n\}_{n=1}^N$ , with  $\mathbf{y}(0) = \tilde{\mathbf{y}}_1$  and  $\mathbf{y}(1) = \tilde{\mathbf{y}}_N$ . Their poses differ by

$$d(\tilde{\mathbf{y}}_n, \mathbf{y}_n) = d_{\mathbb{R}^3}(\tilde{\mathbf{y}}_n^p, \mathbf{y}_n^p) + \alpha d_{\mathcal{S}^3}(\tilde{\mathbf{y}}_n^q, \mathbf{y}_n^q),$$

with positions  $\mathbf{y}^p$ , orientations  $\mathbf{y}^q$ , and a weighting factor  $\alpha$ . This difference can be caused by a need to improve the VMP's weights and by undetected via-points. To reduce it, via-points are added until a termination criterion is fulfilled, e.g.,  $d(\tilde{\mathbf{y}}_n, \mathbf{y}_n) < \theta$  with a given threshold  $\theta$ . Each via-point is created based on its phase variable  $\varphi_v$  and the associated pose  $\tilde{\mathbf{y}}_n$ . We propose three ways to select  $\varphi_v$ . First,  $\varphi_v$  can be chosen as the phase of the *maximum distance*, i.e.,

$$\varphi_v = \underset{\varphi_n}{\operatorname{argmax}} d(\tilde{\mathbf{y}}(\varphi_n), \mathbf{y}(\varphi_n)).$$

Second, a *brute force* search can be conducted among all phase values of the demonstration. Evaluating the VMP with a via-point at  $\varphi_c$  yields a trajectory  $\{\mathbf{y}_n^c\}_{n=1}^N$ . The value  $\varphi_c$  leading to the lowest average distance is picked as

$$\varphi_v = \underset{\varphi_c}{\operatorname{argmin}} \frac{1}{N} \sum_n d(\tilde{\mathbf{y}}_n, \mathbf{y}_n^c).$$

Third,  $\varphi_v$  can be selected via a *weighted phase average* as

$$\varphi_v = \frac{\sum_{n=1}^N d(\tilde{\mathbf{y}}(\varphi_n), \mathbf{y}(\varphi_n)) \varphi_n}{\sum_{n=1}^N d(\tilde{\mathbf{y}}(\varphi_n), \mathbf{y}(\varphi_n))}.$$

For all approaches, redundant and low-influence via-points can be removed at the end of the greedy selection phase.

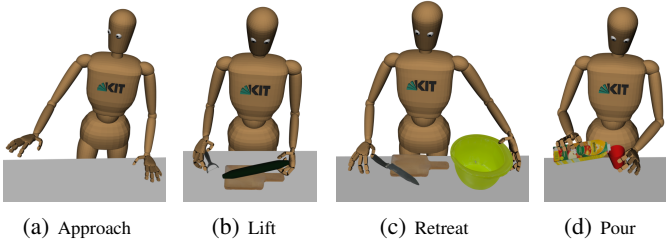


Fig. 3: Snapshots of the tasks from motion capture recordings from [30] that are incrementally learned in Section VI-A.

#### D. Incremental Learning of Task Parameters

While weights are sufficient to characterize a VMP, its execution requires additional task parameters, i.e., a start pose  $\mathbf{y}_0$ , an end pose  $\mathbf{y}_1$ , optional via-points  $\{(\varphi_{v,n}, \mathbf{y}_{v,n})\}_{n=1}^N$ , and a duration  $t$ . Estimating  $\mathbf{y}_0$ ,  $\mathbf{y}_1$  and  $\mathbf{y}_{v,n} \in \mathbb{R}^3 \times \mathcal{S}^3$  amounts to estimating a position  $\mathbf{p} \in \mathbb{R}^3$  and orientation  $\mathbf{q} \in \mathcal{S}^3$ . Moreover,  $\varphi_{v,n}$  and  $t$  are estimated as scalars.

Incremental estimation of scalars and positions is achieved as detailed in Section V-B. Although we require  $\varphi_{v,n} \in [0; 1]$  and  $t > 0$ , we deem the Gaussian estimation to nevertheless be appropriate as we assume the variances to be low. The batchwise mean on Riemannian manifolds is generalized via the so-called Fréchet mean [28]. When samples are provided sequentially, the orientation part of task parameters is obtained using the incremental Fréchet mean estimator [29] as

$$\hat{\mathbf{q}}_{n+1} = \text{Exp}_{\hat{\mathbf{q}}_n} \left( \frac{1}{n+1} \text{Log}_{\hat{\mathbf{q}}_n} (\mathbf{x}_{n+1}) \right). \quad (13)$$

Although strict equivalence between batch-wise and incremental estimates does not apply in the Riemannian case, the estimator (13) provably converges to the batch-wise Fréchet mean [29]. This convergence is sufficient for our use case. Covariances are defined on tangent spaces  $\mathcal{T}_{\hat{\mathbf{q}}_n} \mathcal{M}$  as

$$\hat{\Sigma}_n = \frac{1}{n-1} \sum_{i=1}^n \text{Log}_{\hat{\mathbf{q}}_n} (\mathbf{x}_i) \text{Log}_{\hat{\mathbf{q}}_n} (\mathbf{x}_i)^\top,$$

and incrementally learned as in Section V-B with special care of transporting the covariances to appropriate tangent spaces. Finally, an executable task is represented by a task model containing a sequence of VMPs and their task parameters.

## VI. EXPERIMENTS

We evaluate our approach by incrementally learning a VMP library from human motions recordings. Specifically, we use 6D hand pose trajectories from selected motions of the KIT Bimanual Actions Dataset [30], as shown in Figure 3. We focus on evaluating the execution of the spatial operations formulated in Section V. Therefore, we assume to be given which fundamental operation should be performed when.

#### A. Incremental Learning of a VMP Library

First, we evaluate *adding*, *improving*, *merging two modes* and *splitting a mode*. *Removing* is considered as trivial.

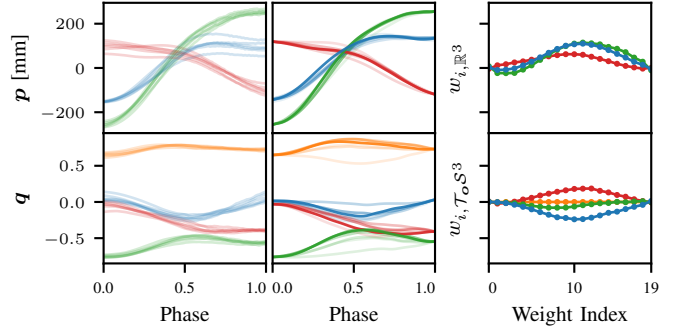


Fig. 4: *Added* and incrementally *improved* VMP for an approach task. *Left*: Incrementally provided demonstrations. *Middle*: Incrementally learned full-pose VMP, with positions  $\mathbf{p}$  ( $\color{red}\blacksquare$   $x$ ,  $\color{green}\blacksquare$   $y$ ,  $\color{blue}\blacksquare$   $z$ ), and orientations  $\mathbf{q}$  ( $\color{red}\blacksquare$   $q_x$ ,  $\color{green}\blacksquare$   $q_y$ ,  $\color{blue}\blacksquare$   $q_z$ ,  $\color{orange}\blacksquare$   $q_w$ ). All executions are performed w.r.t. the start and end of the first demonstration. Colors go from transparent to opaque to show the incremental updates. *Right*: Weight means after 6 demonstrations.

Operation	Distance (translation)	Distance (rotation)
Add and improve	$4.6 \times 10^{-15}$ mm	below detection limit
Merge two modes	$4.0 \times 10^{-15}$ mm	$5.4 \times 10^{-7}^\circ$
Split a mode	22.5 mm and 39.0 mm	$3.2^\circ$ and $6.7^\circ$
(Without splitting)	34.1 mm and 68.2 mm	$4.9^\circ$ and $9.9^\circ$

TABLE I: Average root mean square distances between batch-wise and incremental estimations of the VMP weight means. *Without splitting* is displayed for comparison with *split a mode*.

1) *Adding and Improving*: We use one demonstration of approaching an object with the right hand to perform a sweeping task to create a VMP and *add* it to the library. The VMP is then *improved* incrementally with additional demonstrations. Figure 4 shows the incrementally-learned VMPs, and the resulting weights. As intended, the addition and improvement operations learn a generalized representation of the demonstrations. As expected, quantitative differences between the incremental and batch-wise estimation are negligible (see Table I) since the calculations are mathematically equivalent.

2) *Merging Modes*: Next, we consider two VMPs already encoded in the library, namely lifting to cut and lifting to peel, previously trained from 6 and 3 demonstrations, respectively. As shown in Figure 5-*left*, *middle*, the two VMPs represent similar motions and are incrementally merged into a single lifting VMP (see Figure 5-*right*). As shown in Table I, mathematical equivalence results in negligible differences w.r.t. batch-wise estimations. The imbalanced number of demonstrations per mode is not detrimental.

3) *Splitting a Mode*: We now study the retreat segments of transfer motions, for which the single VMP encoded in the library is not sufficient. As shown in Figure 6-*left*, the learned retreat VMP is not able to represent the elements  $y$  and  $q_z$  of the 4 demonstrated full-pose trajectories. Therefore, the 5th demonstration is used to split the learned VMP into two modes. Figure 6-*middle* shows the resulting two modes, which result in better representations of the demonstrations. 4 additional demonstrations, each automatically assigned to the more probable mode, are further leveraged to refine the VMPs (see Figure 6-*right*). As shown in Table I, the incremental estimation of splitting a mode differs from batch-wise estimations computed from the demonstrations



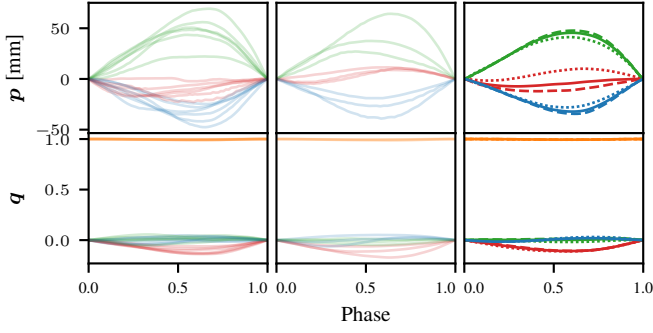


Fig. 5: Merge estimations of lift VMPs. The trajectories are normalized to the same start and end. *Left*: Demonstrations in the context of cutting. *Middle*: Demonstrations in the context of peeling. *Right*: Execution of the individual VMPs from the context of cutting (—) and peeling (···), and of the merged VMP (—).

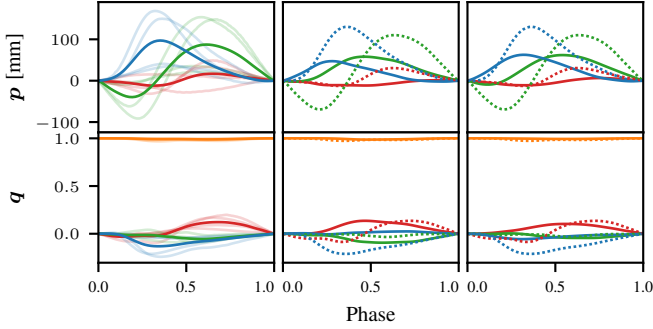


Fig. 6: Splitting a retreat VMP into two modes. The trajectories are normalized to the same start and end. *Left*: Joint estimation (—) after 4 demonstrations. Demonstrations and estimated VMP are depicted by semi-transparent and opaque lines, respectively. *Middle*: Mode 1 (—) and Mode 2 (···) after splitting with a 5th demonstration. *Right*: Mode 1 (—) and Mode 2 (···) after further improvement from 4 additional demonstrations.

of the respective modes. However, the differences are still reasonable, and lower than for a single VMP learned from all demonstrations. Therefore, the incremental splitting was successful, despite the assumption of both modes having been observed equally often at the time of splitting being violated (3 and 2 observations) and the overall number of demonstrations per mode (6 and 3) being imbalanced.

### B. Via-Point Estimation

Next, we compare the three approaches for via-point detection presented in Section V-C on demonstrations of a pouring task, whose amplitudes differ strongly. At first, we detect one via-point per demonstration. As shown in Figure 7-left, the *maximum weight* approach leads to strongly-varying via-points across the demonstrations. Instead, the *weighted phase average* (middle) yields more consistent results, that are better suited to generalize the estimation. As shown in Table II, the *maximum weight* and *brute force* approaches result in similar distances between demonstration and reconstruction with a high estimation time for the latter. The *weighted phase average* is competitively fast, while resulting in similar distances. Figure 7-right shows examples of reconstructions obtained with up to 1 or 3 via-points detected via *weighted phase average*. Here, the 3 points are reduced to 2 in the removal step. For this motion, a single via-point is not sufficient as only the midpart of the reproduction matches

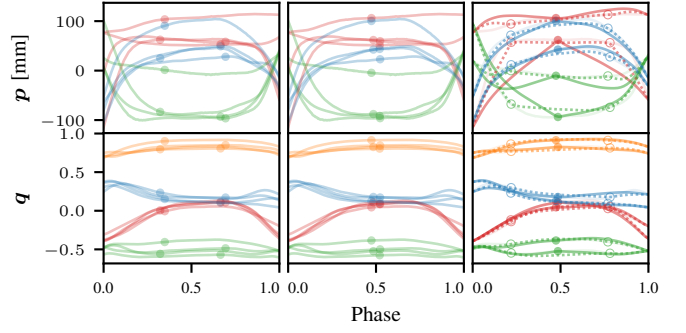


Fig. 7: Estimation of via-points for 9 demonstrations of pouring actions. For clarity, only 4 (left, middle) and 2 of the demonstrations (right) are visualized. *Left*: Detection of a via-point using *maximum distance*. *Middle*: Detection of a via-point using *weighted phase average*. *Right*: Execution using 1 (—/●) or 2 (···/○) via-points detected by *weighted phase average*.

#	Approach	Average Distance	Duration [s]
0	None	(53.8 ± 22.8)	—
1	<i>Maximum Distance</i>	(20.5 ± 10.8)	(0.34 ± 0.03)
	<i>Brute Force</i>	(20.9 ± 10.4)	(81.85 ± 12.50)
	<i>Weighted Phase Average</i>	(22.6 ± 9.3)	(0.33 ± 0.03)
3	<i>Maximum Distance</i>	(10.2 ± 2.9)	(3.11 ± 0.40)
	<i>Weighted Phase Average</i>	(10.1 ± 3.4)	(2.79 ± 0.87)

TABLE II: Evaluation of via-point detection approaches. Average weighted distance between demonstrations and VMP reconstructions and detection duration. *Brute force* is omitted for 3 via-points for runtime reasons. # denotes the maximum number of via-points to be detected.

the demonstration. Trajectories obtained with pairs of via-points reproduces the demonstrations considerably better.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we identified seven fundamental operations to incrementally learn MP libraries. We formulated all spatial operations for a specific type of MP library, thus providing a framework to incrementally learn VMPs, detect their via-points, and estimate their task parameters. We achieve mathematical equivalence to batch-wise estimation of VMP weights for all but one of them and convergence for via-points and task parameters. Our approach soundly treats full-pose trajectories by building on Riemannian manifolds theory. Importantly, it follows a strict definition of incremental learning that limits the amount of stored information.

In general, incremental learning of MP libraries opened the three main challenges of *how* and *when* to apply *which* of the presented fundamental operations. We see the first of these challenges, tackled in this paper, as a requirement to address the latter ones. Future work will consist in deciding *when* to incrementally apply *which* operation to learn generalizable MP libraries from a large set of demonstrations. Once solutions to all three challenges are sufficiently mature, the full potential of incremental learning might be leveraged for long-term usage of MP libraries, thus raising novel challenges regarding the stability and convergence of such a library. Moreover, we envision that the presented solutions on *how* to apply the fundamental operations can be transferred to other MPs building on basis functions, such as DMPs and ProMPs. In addition, we will treat the temporal operations, which requires reliable segmentation approaches.

## REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot Programming by Demonstration,” in *Handbook of Robotics*, pp. 1371–1394, Springer, 2008.
- [2] S. Schaal, “Is Imitation Learning the Route to Humanoid Robots?,” *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [3] S. F. Giszter, F. A. Mussa-Ivaldi, and E. Bizzi, “Convergent Force Fields Organized in the Frog’s Spinal Cord,” *Journal of Neuroscience*, vol. 13, pp. 467–491, 1993.
- [4] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, “Learning and Generalization of Motor Skills by Learning From Demonstration,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 763–768, 2009.
- [5] A. Geppert and B. Hammer, “Incremental Learning Algorithms and Applications,” in *European Symposium on Artificial Neural Networks (ESANN)*, 2016.
- [6] V. Losing, B. Hammer, and H. Wersing, “Incremental On-Line Learning: A Review and Comparison of State of the Art Algorithms,” *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.
- [7] Y. Zhou, J. Gao, and T. Asfour, “Learning Via-Point Movement Primitives With Inter- and Extrapolation Capabilities,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 4301–4308, 2019.
- [8] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, “Incremental Learning of Full Body Motion Primitives and Their Sequencing Through Human Motion Observation,” *Intl. Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.
- [9] S. Niekum, S. Chitta, A. Barto, B. Marthi, and S. Osentoski, “Incremental Semantically Grounded Learning From Demonstration,” in *Robotics: Science and Systems (R:SS)*, 2013.
- [10] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto, “Learning Grounded Finite-State Representations From Unstructured Demonstrations,” *Intl. Journal of Robotics Research*, vol. 34, no. 2, pp. 131–157, 2015.
- [11] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical Movement Primitives: Learning Attractor Models for Motor Behaviors,” *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [12] R. A. Gutierrez, V. Chu, A. L. Thomaz, and S. Niekum, “Incremental Task Modification via Corrective Demonstrations,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 1126–1133, 2018.
- [13] R. A. Gutierrez, E. S. Short, S. Niekum, and A. L. Thomaz, “Learning From Corrective Demonstrations,” in *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, pp. 712–714, 2019.
- [14] F. Meier, E. Theodorou, and S. Schaal, “Movement Segmentation and Recognition for Imitation Learning,” in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 761–769, 2012.
- [15] A. Gams, T. Petrič, M. Do, B. Nemec, J. Morimoto, T. Asfour, and A. Ude, “Adaptation and Coaching of Periodic Motion Primitives through Physical and Visual Interaction,” *Robotics and Autonomous Systems*, vol. 75, pp. 340–351, 2016.
- [16] S. Calinon and A. Billard, “Incremental Learning of Gestures by Imitation in a Humanoid Robot,” in *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, p. 255, 2007.
- [17] K. Kronander, M. Khansari, and A. Billard, “Incremental Motion Learning with Locally Modulated Dynamical Systems,” *Robotics and Autonomous Systems*, vol. 70, pp. 52–62, 2015.
- [18] W. Takano and Y. Nakamura, “Real-Time Unsupervised Segmentation of Human Whole-Body Motion and Its Application to Humanoid Robot Acquisition of Motion Symbols,” *Robotics and Autonomous Systems*, vol. 75, pp. 260–272, 2016.
- [19] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, “Towards Associative Skill Memories,” in *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pp. 309–315, 2012.
- [20] A. Lemme, R. F. Reinhart, and J. J. Steil, “Self-Supervised Bootstrapping of a Movement Primitive Library From Complex Trajectories,” in *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pp. 726–732, 2014.
- [21] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, “Probabilistic Movement Primitives,” in *Neural Information Processing Systems (NeurIPS)*, vol. 26, pp. 2616–2624, 2013.
- [22] L. Rozo and V. Dave, “Orientation Probabilistic Movement Primitives on Riemannian Manifolds,” in *Conference on Robot Learning (CoRL)*, 2021.
- [23] J. Zhang, H. B. Mohammadi, and L. Rozo, “Learning Riemannian stable dynamical systems via diffeomorphisms,” in *Conference on Robot Learning (CoRL)*, pp. 1211–1221, 2022.
- [24] J. M. Lee, *Introduction to smooth manifolds*. Springer, 2013.
- [25] J. M. Lee, *Introduction to Riemannian Manifolds*. Springer, 2018.
- [26] P. Thomas Fletcher, “Geodesic Regression and the Theory of Least Squares on Riemannian Manifolds,” *Intl. Journal on Computer Vision*, vol. 105, no. 2, pp. 171–185, 2013.
- [27] H. J. Kim, N. Adluru, M. D. Collins, M. K. Chung, B. B. Bendin, S. C. Johnson, R. J. Davidson, and V. Singh, “Multivariate General Linear Models (MGLM) on Riemannian Manifolds with Applications to Statistical Analysis of Diffusion Weighted Images,” in *Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2705–2712, 2014.
- [28] X. Pennec, “Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements,” *Journal of Mathematical Imaging and Vision*, vol. 25, no. 1, pp. 127–154, 2006.
- [29] H. Salehian, R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri, “An Efficient Recursive Estimator of the Fréchet Mean on a Hypersphere with Applications to Medical Image Analysis,” *Mathematical Foundations of Computational Anatomy*, vol. 3, pp. 143–154, 2015.
- [30] F. Krebs, A. Meixner, I. Patzer, and T. Asfour, “The KIT Bimanual Manipulation Dataset,” in *IEEE/RAS Intl. Conf. on Humanoid Robots (Humanoids)*, pp. 499–506, 2021.