The following publication A. Duan, W. Liuchen, J. Wu, R. Camoriano, L. Rosasco and D. Navarro-Alarcon, "Learning Rhythmic Trajectories With Geometric Constraints for Laser-Based Skincare Procedures," in IEEE Transactions on Robotics, vol. 41, pp. 1956-1973, 2025 is available at https://doi.org/10.1109/TRO.2025.3543301.

Learning Rhythmic Trajectories with Geometric Constraints for Laser-Based Skincare Procedures

Anqing Duan, *Member, IEEE*, Wanli Liuchen, Jinsong Wu, Raffaello Camoriano, *Member, IEEE*, Lorenzo Rosasco, and David Navarro-Alarcon, *Senior Member, IEEE*

***Abstract*—The increasing deployment of robots has significantly enhanced the automation levels across a wide and diverse range of industries. This paper investigates the automation challenges of laser-based dermatology procedures in the beauty industry; This group of related manipulation tasks involves delivering energy from a cosmetic laser onto the skin with repetitive patterns. To automate this procedure, we propose to use a robotic manipulator and endow it with the dexterity of a skilled dermatology practitioner through a learning-from-demonstration framework. To ensure that the cosmetic laser can properly deliver the energy onto the skin surface of an individual, we develop a novel structured prediction-based imitation learning algorithm with the merit of handling geometric constraints. Notably, our proposed algorithm effectively tackles the imitation challenges associated with quasi-periodic motions, a common feature of many laser-based cosmetic tasks. The conducted real-world experiments illustrate the performance of our robotic beautician in mimicking realistic dermatological procedures; Our new method is shown to not only replicate the rhythmic movements from the provided demonstrations but also to adapt the acquired skills to previously unseen scenarios and subjects.**

***Index Terms*—Robotic manipulation, learning by demonstration, geometric modeling, trajectory planning, cosmetic dermatology robots.**

I. Introduction

DUE to its rapidly advancing nature, robotics has drastically changed many aspects of our lives. In just a couple of decades, we have witnessed how many labor-intensive industrial processes have been upgraded by the use of robots.

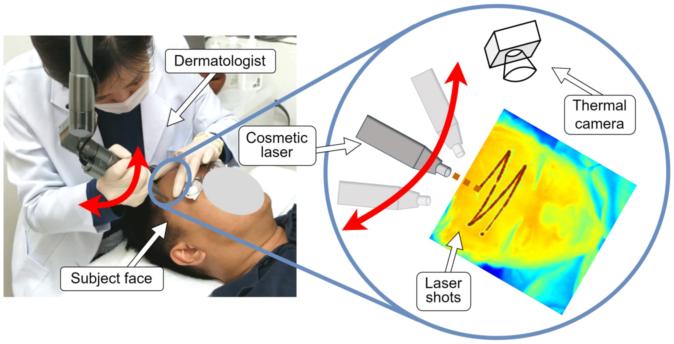


Fig. 1. Illustration of a skin photorejuvenation procedure where a dermatology practitioner rhythmically manipulates a cosmetic laser handpiece that fires multiple laser shots onto the subject's skin surface.

Following this spirit, in this work, we investigate the development of a new generation of robotic systems that automate laser-based cosmetic procedures in the beauty industry [1]. The manipulation tasks under consideration involve the controlled delivery of energy from a cosmetic laser onto the subject's skin with (quasi-)periodic patterns, a procedure that serves multiple purposes, e.g., to improve the aesthetic condition of the skin, decrease the visibility of scars and freckles, remove hair from legs and armpits, to name a few cases [2]. Fig. 1 depicts the so-called skin photorejuvenation task, where a human practitioner "rhythmically" moves the laser handpiece over the area of interest to uniformly deliver thermal energy via pulsating laser shots [3]. This photorejuvenation task (as well as other related laser-based procedures in cosmetic dermatology) can potentially be automated by robots capable of synthesizing skilled (quasi-)periodic motions. Our goal in this paper is precisely to develop this type of system.

Conventional methods for programming a desired robot behavior may not entirely capture the required skills to perform a dermatology procedure, as these approaches typically hardcode specific motions for the robot, which are difficult to adapt to new situations and subjects. To this end, we employ the framework of *learning by demonstration*, also known as robot imitation learning, which offers a convenient paradigm to seamlessly transfer the motion skills from non-expert robot users, such as dermatology practitioners, to a robot by providing the user's motion data [4]. The entire process of learning by demonstration typically comprises three phases [5]: a *demonstration* phase, where motion data is collected to teach robots the desired behavior, a *reproduction* phase, where robots reproduce the learned skills, and an *adap-*

This work is supported in part by the Research Grants Council (RGC) of Hong Kong under grant 15212721, and in part by the Jiangsu Industrial Technology Research Institute Collaborative Funding Scheme under grant ZG9V. R. C. acknowledges the following: This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (Piano nazionale di ripresa e resilienza (pnrr) - missione 4 componente 2, investimento 1.3 - D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. *Corresponding author: David Navarro-Alarcon.*

A. Duan is with Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), United Arab Emirates (anqing.duan@mbzuai.ac.ae).

W. Liuchen, J. Wu, and D. Navarro-Alarcon are with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hong Kong. (e-mail: aduan@polyu.edu.hk, chenwanli.liu@connect.polyu.hk, jinsong.wu@connect.polyu.hk, dnavar@polyu.edu.hk).

R. Camoriano is with the Visual and Multimodal Applied Learning Laboratory, Politecnico di Torino, Turin, Italy, and with Istituto Italiano di Tecnologia, Genoa, Italy (e-mail: raffaello.camoriano@polito.it).

L. Rosasco is with DIBRIS, Università degli Studi di Genova, Genoa, Italy, with Laboratory for Computational and Statistical Learning (IIT@MIT), Istituto Italiano di Tecnologia and Massachusetts Institute of Technology, Cambridge, MA, United States, and also with Machine Learning Genoa (MaLGa) Center, Università di Genova, Genoa, Italy (e-mail: lrosasco@mit.edu).

tation phase, where robots adapt the learned skills to operate in a different environment from the initial demonstration.

From an algorithmic perspective, studying representations for motor skills holds significant importance in facilitating robot imitation learning. The well-established *movement primitives* emerges as a powerful tool [6]. Once the demonstration is complete, movement primitives are commonly utilized for motion analysis to extract motion patterns from experts' demonstrations. Subsequently, during the reproduction and adaptation phases, the learned movement primitives will be employed to accomplish robot tasks.

Our Contribution. Several technical challenges arise when transferring laser manipulation skills from a human expert to a robot. The main complications are twofold: (i) extracting the laser's distinct motion patterns that entail the laser spots rhythmically evolving on a subject's skin surface, and (ii) adapting the extracted motor skills for subsequent treatment of a new subject. Accordingly, the original contributions of this paper are outlined as follows:

- Development of a novel imitation learning algorithm that can generate rhythmic motion with geometric constraints for laser-based cosmetic procedures.
- Derivation of a motor skill adaptation strategy for treating a new subject's face that is absent in the demonstration.

Specifically, compared with state-of-the-art methods, our proposed imitation learning algorithm is characterized by:

- Allowing for imitation of trajectories that involve geometric constraints.
- Supporting the acquisition of periodic or quasi-periodic motion patterns.
- Eliminating the need for manual configuration or fine-tuning of basis functions.

Since photorejuvenation-based skincare involves delivering multiple laser shots across the skin surface, it is crucial to adhere to the constraints imposed by the surface's geometry. Therefore, the ability to satisfy these geometric constraints is an essential requirement to learn an effective motion policy. This capability represents a key advantage of our algorithm.

Our proposed new algorithm can be readily utilized for a variety of imitation learning problems that occur in either the Euclidean space or in the presence of other types of manifold constraints. This versatility expands its potential applications beyond the specific context of cosmetic dermatology [7]. For example, our method can also be used for other cosmetic applications, such as laser-based hair removal. Additionally, our method can enable robotic artistic creation by learning to draw images or engrave patterns. Moreover, in manufacturing, our method can be leveraged to learn motion skills for welding or laser-based cutting processes.

Organization. The rest of the paper is organized as follows: Sec. II presents the related work; Sec. IV describes the developed learning from demonstration algorithm; Sec. V derives the skill adaptation approach; Sec. VI numerically and experimentally validates the proposed methodology; Sec VII concludes the paper. Notation is provided in Appendix A.

II. Related Work

Geometric trajectory generation. During photorejuvenation, the robotic arm must generate laser traces on the subject's face. Therefore, it is essential to ensure that these traces conform to the geometric constraints of the facial surface. To achieve geometric trajectory generation, multiple perspectives can be referred to, such as mapping 2D curves onto the 3D surface [8]–[10] and sampling [11], [12] or neural networks-based motion planning [13]. Compared to the aforementioned approaches, our approach tackles the problem from an imitation perspective where the geometric constraints are guaranteed by addressing a geometric optimization problem. It overcomes the distortion issue commonly associated with the mapping strategy. Moreover, in comparison to sampling-based methods, our approach exhibits lower computational complexity. Particularly, the imitation framework offers a user-friendly interface for non-expert robot users, making it well-suited for dermatologists to transfer their motion skills to a robot arm.

Constrained and latent-space imitation learning. Learning with constraints has long been a significant area of research. Constraints can be addressed through various techniques, such as bounded transformation functions [14], constrained optimization formulations [15], or null-space estimation [16], among others. However, unlike previous work on constrained imitation learning, this study specifically focuses on constraints arising from Riemannian geometry, which has seen widespread applications in robotics, including tasks involving orientation, manipulability, and joint stiffness. The extension of imitation learning to Riemannian manifolds has been gaining increasing attention [17]–[20]. In contrast to earlier approaches, we propose addressing geometric constraints through the use of *structured prediction*, which has proven to be an effective tool for enabling robot movement imitation [21]. A key element of our framework is the exploitation of kernel functions in the embedding space, which provide flexibility in capturing complex motion patterns. Notably, our framework's embedding space is implicit, distinguishing it from other methods that rely on the explicit design of sophisticated latent spaces, such as those in [22]–[24].

Geodesic distance calculation. Typically, the calculation of geodesic distances constitutes a crucial component in imitation learning on manifolds. The geodesic distance is commonly available as a by-product of dimensionality reduction methods such as Isometric Mapping (ISOMAP) [25] or Multi-Dimensional Scaling (MDS) [26]. Also, physics-inspired methods, such as the heat diffusion equation [27] and the Eikonal equation [28], offer an alternative approach to calculating geodesic distances. In comparison to the aforementioned approaches, we propose to approximate a surface using osculating spherelets, which have the merit of analytic expressions for geodesic distances. Consequently, it is efficient to calculate the geodesic distance between two manifold points, without iteratively searching for the shortest path or solving differential equations.

Rhythmic movement learning. Another related area of study is the acquisition of rhythmic movement skills. Rhythmic motion in robotic systems has been applied in various tasks, including wiping tables [29], stirring with a spoon [30], and playing the drums [31]. Imitation based on parametric approaches usually captures periodic movement by cyclic basis functions [32], while non-parametric imitation methods typically utilize periodic kernel functions [33]. In this work, we explicitly provide definitions for (quasi-)periodic trajectories under geometric constraints. Besides, signal processing techniques, such as the Fourier transform, can also be applied to capture quasi-periodic motion patterns [34]. However, these approaches disregard the geometry-structured rhythm that our method emphasizes.

III. Preliminaries

In this section, we outline the essential technical prerequisites for the developed imitation algorithm. Specifically, we review the structured prediction strategy, which effectively addresses geometric constraints in motion imitation. Our goal is to capture the underlying motion patterns based on the demonstrations provided.

To begin with, the dataset of laser trajectory points, representing the intersections between the laser ray and the skin surface as demonstrated by a dermatologist, is denoted as $\mathbb{T} : \{(t_n, \mathbf{p}_n = [x_n, y_n, z_n]^\intercal)\}_{n=1}^N$, where $t \in \mathcal{T}$ is the time stamp with $\mathcal{T}$ being the input space and $\mathbf{p}$ is the corresponding trajectory position. During the treatment, the trajectory of the laser generator cannot freely evolve. Rather, it is required to be constrained on the inflated surface at a certain height above the subject's face.

Remark 1. All the laser trajectory points from the demonstrated dataset $\mathbb{T}$ lie on the surface manifold $\mathcal{S}$, i.e., $\mathbf{p}_n \in \mathcal{S}$ with $\forall(t_n, \mathbf{p}_n) \in \mathbb{T}$.

The inherent constraint underlying the demonstrated trajectories poses a unique challenge in applying existing imitation learning algorithms due to the Riemannian metric associated with the surface manifold. It is thus imperative to devise geometry-aware movement primitives, which is exactly the objective of this paper. As a powerful tool to deal with supervised learning, structured prediction is especially competent at handling cases where outputs possess geometrically rich structures [35]. In this paper, we propose to tackle the issue of learning (quasi-)periodic trajectories for skin photorejuvenation by resorting to the structured prediction framework.

A major strategy employed in structured prediction is the surrogate approach [36], whose core notion can be outlined by: 1) Embedding the outputs into a linear surrogate space $\mathcal{H}$, then 2) Solving the learning problem in the surrogate space, and finally 3) Mapping the solution back to the structured space with a decoding rule. More formally, the surrogate approach to structured prediction is sketched as follows:

1) *Encoding.* Design an encoding rule $\mathbf{c} : \mathcal{S} \to \mathcal{H}$.
2) *Surrogate Learning.* Solve the surrogate learning problem $\mathbf{g} : \mathcal{T} \to \mathcal{H}$ which minimizes $\mathcal{L}(\mathbf{c}(\mathbf{p}_n), \mathbf{g}(t_n))$ with the surrogate loss $\mathcal{L} : \mathcal{H}\times\mathcal{H} \to \mathbb{R}$, given the surrogate dataset $\mathbb{D}_s : \{t_n, \mathbf{c}(\mathbf{p}_n)\}_{n=1}^N$.
3) *Decoding.* Obtain the structured output $\mathbf{s} = \mathbf{c}^{-1} \circ \mathbf{g} : \mathcal{T} \to \mathcal{S}$ with a suitable decoding $\mathbf{c}^{-1} : \mathcal{H} \to \mathcal{S}$.

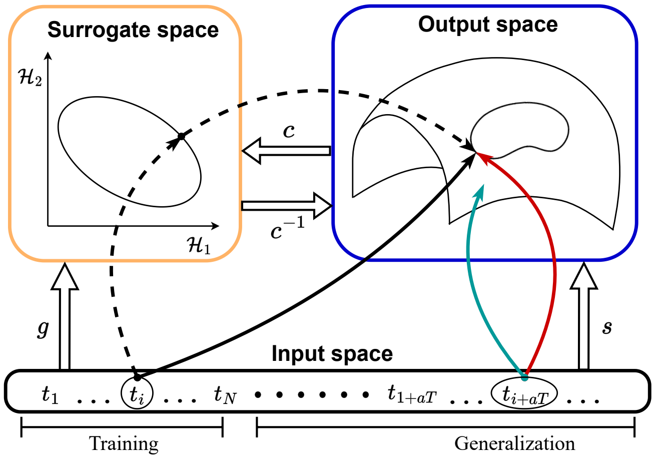


Fig. 2. Illustration of making structured prediction for periodic motions using the surrogate approach. Black curves denote the training procedure while red curves and cyan curves denote the generalization procedure using the periodic and squared exponential kernels, respectively.

There emerge recent insights on the loss function in the structured space that it implicitly carries a natural corresponding geometric structure. The exploitation of such a structure allows us to relax the aforementioned surrogate framework further, leading to an implicit formulation.

Definition 1 (Structure Encoding Loss Function [36])**.** A loss function $\Delta : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is called a Structure Encoding Loss Function if there exist a separable Hilbert space $\mathcal{H}$ associated with an inner product $\langle\cdot,\cdot\rangle_\mathcal{H}$, a continuous feature map $\boldsymbol{\Psi} : \mathcal{S} \to \mathcal{H}$ and a continuous linear operator $\mathbf{V} : \mathcal{H} \to \mathcal{H}$ such that for all $\mathbf{p}, \mathbf{p}' \in \mathcal{S}$,

$$\Delta(\mathbf{p}, \mathbf{p}') = \langle \boldsymbol{\Psi}(\mathbf{p}), \mathbf{V}\boldsymbol{\Psi}(\mathbf{p}')\rangle_\mathcal{H}. \tag{1}$$

By leveraging the ridge regression estimator that minimizes the regularized empirical risk [37], we can formulate the learning problem in the surrogate space as a regularization problem

$$\mathbf{g} = \operatorname*{argmin}_{\mathbf{g}\in\mathcal{G}} \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{\Psi}(\mathbf{p}_n) - \mathbf{g}(t_n)\|_\mathcal{H}^2 + \lambda\|\mathbf{g}\|_\mathcal{G}^2, \tag{2}$$

where $\mathcal{G}$ is a normed space of functions $\mathcal{T} \to \mathcal{H}$ and $\lambda > 0$ is the Tikhonov hyperparameter. Considering a reproducing kernel Hilbert space (RKHS) of vector-valued functions and the associated matrix-valued kernel $\boldsymbol{\mathcal{K}} : \mathcal{T} \times \mathcal{T} \to \mathbb{R}^{D\times D}$ where D is the dimension of $\mathcal{H}$, the solution to (2) is obtained by employing the *representer theorem* within the vector-valued setting [37]

$$\mathbf{g}(t) = \sum_{n=1}^{N} \boldsymbol{\mathcal{K}}(t_n, t)\boldsymbol{\beta}_n. \tag{3}$$

By choosing the matrix-valued kernel to be $\boldsymbol{\mathcal{K}}(t, t') = k(t, t')\mathbf{I}_\mathcal{H}$ with $k : \mathcal{T} \times \mathcal{T} \to \mathbb{R}$ being a reproducing kernel, the concatenation of the coefficient $\boldsymbol{\beta}_n$ gives

$$\widetilde{\boldsymbol{\beta}} = \left[\boldsymbol{\beta}_1^\intercal, \dots, \boldsymbol{\beta}_N^\intercal\right]^\intercal = \left((\mathbf{K} + \lambda N\mathbf{I}_N) \otimes \mathbf{I}_\mathcal{H}\right)^{-1}\widetilde{\boldsymbol{\Psi}} \tag{4}$$

where the concatenation of the output vectors is given as $\widetilde{\boldsymbol{\Psi}} = \left[\boldsymbol{\Psi}_1^\intercal, \dots, \boldsymbol{\Psi}_N^\intercal\right]^\intercal$ and $\mathbf{K} \in \mathbb{R}^{N \times N}$ is defined by $\mathbf{K}_{i,j} = k(t_i, t_j)$. For any $t \in \mathcal{T}$, we then have

$$\mathbf{g}(t) = (\mathbf{k}^\intercal \otimes \mathbf{I}_\mathcal{H})\big((\mathbf{K} + \lambda N \mathbf{I}_N) \otimes \mathbf{I}_\mathcal{H}\big)^{-1} \widetilde{\boldsymbol{\Psi}} \tag{5a}$$

$$= \sum_{n=1}^{N} \boldsymbol{\alpha}_n(t) \boldsymbol{\Psi}(\mathbf{p}_n), \tag{5b}$$

where $\boldsymbol{\alpha}(t) = [\boldsymbol{\alpha}_1(t), \dots, \boldsymbol{\alpha}_N(t)]^\intercal = (\mathbf{K}+\lambda N \mathbf{I}_N)^{-1}\mathbf{k} \in \mathbb{R}^N$ and $\mathbf{k} \in \mathbb{R}^N$ is constructed by $\mathbf{k}_i = k(t, t_i)$. Notably, the kernel function involved in (3) plays a crucial role in capturing motion data patterns, and its design is a central focus of this work.

Leveraging the structure of the SELF loss function in the expected risk, the following decoding rule can be introduced:

$$\mathbf{c}^{-1}(\boldsymbol{\zeta}) = \operatorname*{argmin}_{\mathbf{p} \in \mathcal{S}} \langle \boldsymbol{\Psi}(\mathbf{p}), \mathbf{V}\boldsymbol{\zeta} \rangle_\mathcal{H}, \tag{6}$$

where $\boldsymbol{\zeta} \in \mathcal{H}$.

Finally, by combining (5b) with (6), the estimated output can be exposed as [36]

$$\mathbf{s}(t) = \operatorname*{argmin}_{\mathbf{p} \in \mathcal{S}} \sum_{n=1}^{N} \boldsymbol{\alpha}_n(t) \Delta(\mathbf{p}, \mathbf{p}_n). \tag{7}$$

The usage of estimator (7) consists of two steps: 1) *Training*: the score function $\boldsymbol{\alpha}$ is computed given a test input t, and 2) *Prediction*: a linear α_n-weighted cost is minimized. Due to the presence of $\Delta(\mathbf{p}, \mathbf{p}_n)$ in the prediction step, it is advantageous to have an efficient method for this computation, which we will discuss in the following section.

Remarkably, the merits of structured prediction for motion imitation are twofold. First, it enables the learning from data with complex structures. Second, compared with other mapping space-based imitation strategies, there is no need for explicit knowledge of the embedding space $\mathcal{H}$, the feature map $\boldsymbol{\Psi}$, or the operator $\mathbf{V}$ thanks to the structure of the loss Δ. Hence an implicit embedding framework is attained, as shown in Fig. 2.

IV. Imitation of Geometric Trajectories with Rhythm

In this section, we illustrate the determination of the kernel functions for motion pattern capture (Sec. IV-A), followed by efficient calculation of loss functions (Sec. IV-B).

A. Kernel Design for Motion Pattern Capture

As in other laser-related applications, the trajectory of the cosmetic laser also exhibits the property of (quasi-)periodicity. To capture such a rhythmic motion pattern, it is not a trivial issue to consider the design of the reproducing kernels since the form of the kernel significantly determines the *generalization* capability of a learning model.

The impact of kernel functions on pattern analysis within nonparametric learning methods has been a crucial concern. The design of the kernels for pattern analysis has been well studied from the Bayesian perspective such as Gaussian processes (GP) where the kernels appear as covariance functions [38]. The design of kernel functions is application-dependent, and certain kernels may be more suitable than others for specific scenarios. Additionally, combining kernels, through operations like multiplication or addition (i.e., multiple kernel learning), remains a powerful technique for capturing complex patterns that a single kernel function may not be able to represent. Also, kernel properties are exploited in the frequentist learning field for supervised tasks. Despite Bayesian learning with GP and frequentist methods with RKHS representing two distinct approaches to nonparametric regression, the properties of the kernels can indeed be harnessed due to the connections between Bayesian and frequentist regularization approaches [39].

Remark 2. Prediction with RKHS as in (3) is equivalent to the mean prediction of a multi-task GP with the zero mean prior and noise variance being $\lambda N \mathbf{I}_N$ using the same kernel.

Given the equivalence between GP and RKHS, the constitution of kernels considered in the context of GP can also be directly leveraged into the regularization framework of RKHS. In other words, it is reasonable for one to design kernels for RKHS by referring to the properties of the kernels in GP as discussed by [40, Ch. 2].

As a *universal* kernel, the squared exponential kernel is commonly employed as a default kernel in kernel-based machine learning, which has the form:

$$k_{\mathtt{SE}}(t, t') = \sigma_s^2 \exp\left(-\frac{(t-t')^2}{2l_s^2}\right), \tag{8}$$

where length-scale l_s and variance σ_s^2 are the hyperparameters. Despite its prevalence, the squared exponential kernel could fail to capture certain patterns. We shall see shortly that it is not suitable for expressing the robot's periodic movement, whose definition is given as follows.

Definition 2 (Periodicity of a geometric trajectory)**.** A periodic geometric trajectory is defined by

$$\mathbf{P}(t) = \mathbf{P}(t+T), \tag{9}$$

where $T > 0$ is called the period of the geometric trajectory.

In the following, we will show the necessity of choosing a proper kernel for capturing our motion pattern.

Proposition 1 (Model misspecification)**.** Given a query point $t \to +\infty$, the output of the estimator (7) with the squared exponential kernel (8) remains constant.

Proof. For any two training points t_{n1} and t_{n2}, we have

$$\lim_{t \to +\infty} \frac{\mathbf{k}_{n1}(t)}{\mathbf{k}_{n2}(t)} = \exp\left(\frac{(t-t_{n2})^2 - (t-t_{n1})^2}{2l_s^2}\right) = 1, \tag{10}$$

which implies that $\mathbf{k}$ becomes a vector with all equal elements with the query point t being far from the training set. As a result, $\boldsymbol{\alpha}$ becomes independent of the query input, leading to the loss function of the n-th element scored by a constant. In particular, the constant weight is given by $\boldsymbol{\alpha}_n = \sum_{i=1}^{N}[(\mathbf{K} + \lambda N \mathbf{I}_N)^{-1}]_{n,i}$. ■

Proposition 1 claims that the estimator (7) with the squared exponential kernel cannot *extrapolate* well on the periodic

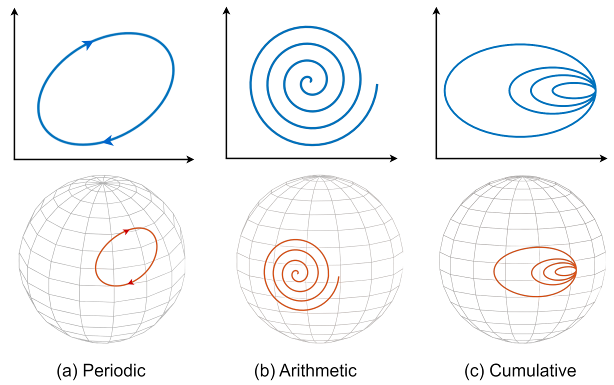

(a) Periodic (b) Arithmetic (c) Cumulative

Fig. 3. Illustration of (a) periodic, (b) arithmetic, and (c) cumulative patterns where Euclidean trajectories (*upper row*) are plotted in blue and trajectories on a manifold (*bottom row*) are plotted in red.

dataset since the output is simply a constant as the query point moves far away from the training dataset. A similar problem could also happen in the case of *interpolation*. When the interval between two consecutive training points is larger than one period of the periodic trajectory, the prediction along this interval is also non-periodic, which is caused by the non-periodicity of $\boldsymbol{\alpha}$.

In view of the *model misspecification* issue, we propose to employ the periodic kernel to capture the periodic motion pattern. The form of the periodic kernel is given by [41]

$$k_{\texttt{PER}}(t,t') = \sigma_p^2 \exp\left(-\frac{2\sin^2\left(\pi(t-t')/p\right)}{l_p^2}\right), \tag{11}$$

where period p, length-scale l_p, and variance σ_p^2 are the kernel hyper-parameters.

Proposition 2 (Blessing of Abstraction)**.** Given a query point t, the output of the estimator (7) with the periodic kernel (11) remains periodic.

Proof. Given a training point t_n, we have

$$\begin{aligned}\mathbf{k}_n(t) &= \sigma_p^2 \exp\left(-\frac{2\sin^2\left(\pi(t-t_n)/p\right)}{l_p^2}\right)\\ &= \sigma_p^2 \exp\left(-\frac{2\sin^2\left(\pi(t+p-t_n)/p\right)}{l_p^2}\right) \qquad (12)\\ &= k_{\texttt{PER}}(t+p,t_n) = \mathbf{k}_n(t+p),\end{aligned}$$

which implies that $\mathbf{k}$ is periodic with the period being p. Therefore, the output of (7) becomes also periodic due to the periodicity of $\boldsymbol{\alpha}$. ■

Proposition 2 shows that based on a properly designed kernel function, our structured predictor can perform well regarding out-of-distribution generalization, as depicted in Fig. 2, since the underlying motion pattern is accounted for by the structure of the kernel function, which gives rise to the *blessing of abstraction*.

It is worth noting that although the motion pattern of the structured output space (9) suggests the usage of the periodic kernel, it is the surrogate space, where the kernel ridge regression is performed as in (3), that directly prescribes the kernel design. Therefore, it is of interest to investigate the behavior of the vectors in the surrogate space. In fact, given (9) holds, we can conclude that the surrogate space indeed exhibits periodic, namely $\boldsymbol{\Psi}(\mathbf{p}(t)) = \boldsymbol{\Psi}(\mathbf{p}(t{+}T))$, which can be shown by the bijection property of the feature map.

In practice, the periodic trajectory may not exactly recur itself. Most likely, the trajectory could be *quasi-periodic*, i.e. the recurring part may evolve over time [42]. It is therefore necessary to modify the periodic kernel such that the pattern of quasi-periodicity can be captured by the modified kernel. In this regard, we propose to employ the quasi-periodic kernel obtained by the product of a periodic kernel and a squared exponential kernel [43]:

$$k_{\texttt{QP}}(t,t') = k_{\texttt{SE}}(t,t') \times k_{\texttt{PER}}(t,t'). \tag{13}$$

As a sanity check, it is also worth investigating the corresponding behavior of the structured output space given the quasi-periodicity of the surrogate space to justify the usage of the quasi-periodic kernel in the surrogate space learning. To this end, we first define two representative quasi-periodic patterns for the case of vector-valued functions, provided $\mathbf{q}(t)$ is defined on $[0,T]$.

Definition 3 (Quasi-Periodicity for vector-valued function)**.** $\textbf{Arithmetic}:$

$$\mathbf{q}(\tau + aT) = \mathbf{q}(\tau + (a-1)T) + \mathbf{C}(\tau), \tag{14a}$$

$\textbf{Cumulative}:$

$$\mathbf{q}(\tau + aT) = \mathbf{q}(\tau + (a-1)T) + a\mathbf{C}(\tau), \tag{14b}$$

where $\tau \in [0,T]$, $a \in \mathbb{N}^+$ denotes the period index, and $\mathbf{C}(\tau)$ is a continuous function deciding the temporal change following the first period.

Akin to Definition 3, we extend the corresponding concepts of quasi-periodicity to their geometric counterparts.

Definition 4 (Quasi-Periodicity for geometric trajectory)**.** $\textbf{Arithmetic}:$

$$\mathbf{P}(\tau + aT) = \texttt{Exp}_{\mathbf{P}(\tau+(a-1)T)}\boldsymbol{\Gamma}_{\mathbf{P}(\tau)\to\mathbf{P}(\tau+(a-1)T)}\mathfrak{C}(\tau), \tag{15a}$$

$\textbf{Cumulative}:$

$$\mathbf{P}(\tau + aT) = \texttt{Exp}_{\mathbf{P}(\tau+(a-1)T)}\boldsymbol{\Gamma}_{\mathbf{P}(\tau)\to\mathbf{P}(\tau+(a-1)T)}a\mathfrak{C}(\tau), \tag{15b}$$

where $\mathbf{P}(t)$ is given on $[0,T]$, and $\mathfrak{C}(\tau)$ is defined on the tangent space at $\mathbf{P}(\tau)$.

Fig. 3 illustrates the periodic and quasi-periodic behaviors on a plane and a sphere manifold.

Theorem 1. If the trajectories in surrogate space satisfy quasi-periodic behavior as in (14a) or (14b), then the geometric trajectories conform to (15a) and (15b), respectively.

Proof. See Appendix B. ■

Remarkably, Theorem 1 states that the quasi-periodic pattern in the surrogate space denotes a sufficient condition for the geometric output space to be quasi-periodic as well.

B. Loss Function Calculation

While solving the geometric optimization problem in (7) ensures that the trajectory adheres to the geometric constraints, computing the geodesic distance involved can be computationally intensive. The geodesic path on 3D surfaces is usually difficult to find and the procedure may require many iterations [44]. Provided that we need to compute the geodesic distance of a manifold point to all demonstrated trajectory points, it is thus favorable to efficiently compute the geodesic distance of two points on the facial surface.

We propose to exploit the strategy of *spherelets*, which refers to manifold approximation using spheres [45]. Calculating geodesic distance using the spherelet technique is efficient because it allows for a closed-form solution. While other geometric shapes, such as cylinders, may yield more accurate approximations in certain areas of the face, they are less preferred in our work due to their higher computational cost.

The main steps of using spherelets for geodesic distance calculation are sketched as follows: 1) Partitioning the facial surface into composing regions, 2) Designating an *osculating sphere* for each region to approximate the manifold, and 3) Calculating the geodesic distance by summing the length of each composing geodesic path segment.

In this work, we propose partitioning the facial surface using a grid composed of an array of intersecting lines. For simplicity, we set the grid lines to be uniformly spaced.

As a result, the surface manifold is composed by $\mathcal{S} = \bigcup_{s=1}^{d} \mathbb{S}_s^2$, where d denotes the total number of regions. As mentioned earlier, we propose to use an osculating sphere to approximate each surface region. Formally, an osculating sphere is defined by $\mathbb{S}_s^2 := \{\boldsymbol{\zeta} : \|\boldsymbol{\zeta} - \mathbf{O}_s\| = r_s\}$. Given the 3D points collection of the facial surface $\mathbb{F} : \{\boldsymbol{\xi}_m = (x_m, y_m, z_m)\}_{m=1}^{M}$, the center $\mathbf{O}_s$ and the radius r_s are obtained by minimizing the following *algebraic* loss function:

$$C_s(\mathbf{O}_s, r_s) = \sum_{\boldsymbol{\xi}_i \in \mathbb{S}_s^2} \left((\boldsymbol{\xi}_i - \mathbf{O}_s)^\intercal (\boldsymbol{\xi}_i - \mathbf{O}_s) - r_s^2 \right)^2. \tag{16}$$

Lemma 1 (Osculating Sphere Estimation [45])**.** The minimizer to (16) is given by

$$\mathbf{O}_s = \frac{1}{2}\boldsymbol{\Lambda}_s^{-1}\boldsymbol{\theta}_s \quad \text{and} \quad r_s = \frac{1}{M_s}\sum_{\boldsymbol{\xi}_i \in \mathbb{S}_s^2} \|\boldsymbol{\xi}_i - \mathbf{O}_s\|, \tag{17}$$

where M_s denotes the number of data points in the corresponding region and

$$\boldsymbol{\Lambda}_s = \sum_{\boldsymbol{\xi}_i \in \mathbb{S}_s^2} (\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_s)(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_s)^\intercal, \tag{18a}$$

$$\boldsymbol{\theta}_s = \sum_{\boldsymbol{\xi}_i \in \mathbb{S}_s^2} (\boldsymbol{\xi}_i^\intercal \boldsymbol{\xi}_i - \frac{1}{M_s}\sum_{\boldsymbol{\xi}_i \in \mathbb{S}_s^2} \boldsymbol{\xi}_i^\intercal \boldsymbol{\xi}_i)(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}}_s). \tag{18b}$$

To determine the segmentation points on the geodesic path connecting two manifold points of interest $\boldsymbol{\xi}_a$ and $\boldsymbol{\xi}_b$, we identify a plane that is determined by the maximum variance of the facial surface, i.e. the projected area of the face surface onto the plane is maximized. To this end, we compute the data covariance matrix as

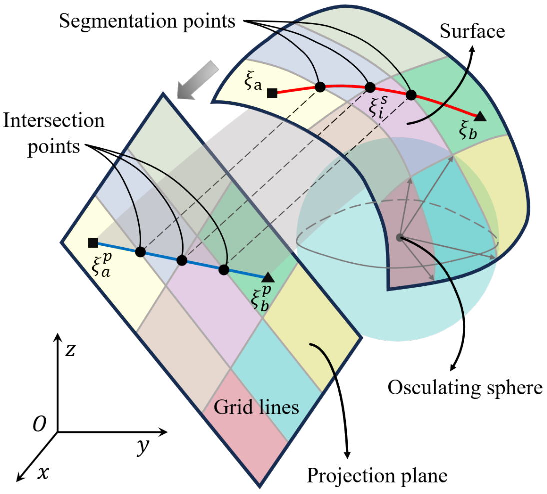


Fig. 4. Illustration of geodesic distance calculation by spherelets. The geodesic distance between two points as denoted by a square and a triangle on the surface is obtained by summing the lengths of the geodesic curves on the composing spheres. The segmentation points are determined using the intersection points between the mapped straight curve and the specified grid lines on the projection plane.

$$\boldsymbol{\Sigma}_f = \frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{\xi}_m - \bar{\boldsymbol{\xi}})(\boldsymbol{\xi}_m - \bar{\boldsymbol{\xi}})^\intercal, \text{ with } \bar{\boldsymbol{\xi}} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{\xi}_m. \tag{19}$$

The projection plane is then identified by aligning its normal with the least principal component of the data points, which is the eigenvector with the smallest eigenvalue of $\boldsymbol{\Sigma}_f$. By projecting $\boldsymbol{\xi}_a$ and $\boldsymbol{\xi}_b$ to the projection plane, we obtain the projected points $\boldsymbol{\xi}_a^p$ and $\boldsymbol{\xi}_b^p$, respectively. On the projection plane, we compute the intersection points by crossing the straight line $\overline{\boldsymbol{\xi}_a^p \boldsymbol{\xi}_b^p}$ with the grid lines $\{\{\boldsymbol{l}_{vi}^p\}_{i=1}^{nv}, \{\boldsymbol{l}_{hj}^p\}_{j=1}^{nh}\}$ where $\boldsymbol{l}_{vi}^p$ denote the vertical lines and $\boldsymbol{l}_{hj}^p$ denote the horizontal lines. Then, the segmentation points are obtained by projecting the intersection points on the projection plane to the approximated manifold, i.e., $\boldsymbol{\xi}_{vi,hi}^s = \boldsymbol{\Pi}_p^s(\boldsymbol{l}_{vi}^p, \boldsymbol{l}_{hi}^p \cap \overline{\boldsymbol{\xi}_a^p \boldsymbol{\xi}_b^p})$ where $\boldsymbol{\Pi}_p^s(\cdot)$ perpendicularly maps a point on the projection plane to the point lying on the corresponding approximated sphere manifold.

Finally, by sorting the segmentation points to be $\{\boldsymbol{\xi}_i^s\}_{i=1}^{L}$ such that they are ordered in accordance with the direction of $\boldsymbol{\xi}_a^p$ to $\boldsymbol{\xi}_b^p$, the geodesic distance can be approximated by

$$\mathcal{D}(\boldsymbol{\xi}_a, \boldsymbol{\xi}_b) \approx \sum_{l=0}^{L} r_l \arccos\left(\frac{\boldsymbol{\xi}_l^s - \mathbf{O}_l}{r_l} \cdot \frac{\boldsymbol{\xi}_{l+1}^s - \mathbf{O}_l}{r_l} \right), \tag{20}$$

where r_l and $\mathbf{O}_l$ represent the radius and center of the corresponding sphere, respectively. Besides, we have $\boldsymbol{\xi}_0 = \boldsymbol{\xi}_a$ and $\boldsymbol{\xi}_{L+1} = \boldsymbol{\xi}_b$. As an approximation method, the spherelet technique may result in non-smooth trajectories at the segmentation boundaries. While this is not a significant issue for the non-contact laser pulse application considered in this work, a dedicated trajectory smoothing procedure could be implemented for applications that require higher levels of smoothness. Fig. 4 provides a pictorial illustration of geodesic distance approximation using spherelets.

Given the solution to the geodesic distance calculation by (20), the geometric optimization problem in (7) can be readily addressed by *Riemannian gradient descent*, which represents the Riemannian counterpart of the usual gradient descent method [36]. The update rule iteratively proceeds as

$$\mathbf{p}_{i+1} = \mathtt{Exp}_{\mathbf{p}_i}\left(\eta_i \nabla_{\mathcal{M}} \mathtt{F}(\mathbf{p}_i)\right), \tag{21}$$

where $\eta_i \in \mathbb{R}$ is a step size, and we denote

$$\mathtt{F}(\mathbf{p}_i) = \sum_{n=1}^{N} \boldsymbol{\alpha}_n(t)\Delta(\mathbf{p}_i, \mathbf{p}_n), \tag{22a}$$

$$\Delta(\mathbf{p}_i, \mathbf{p}_n) = \mathcal{D}^2(\mathbf{p}_i, \mathbf{p}_n). \tag{22b}$$

Specifically, $\nabla_{\mathcal{M}}$ denotes the Riemannian gradient operator and the gradient defined with respect to the Riemannian metric is given by the projection of the usual gradient to the tangent space [46], which reads

$$\nabla_{\mathcal{M}} \mathtt{F}(\mathbf{p}_i) = \mathtt{Proj}_{\mathbf{p}_i} \nabla \mathtt{F}(\mathbf{p}_i). \tag{23}$$

Wherein we have

$$\textit{Orthogonal projector}: \quad \mathtt{Proj}_{\mathbf{p}_i} \triangleq r_i^2 \mathbf{I} - \mathbf{p}_i \mathbf{p}_i^{\mathsf{T}}, \tag{24a}$$

$$\textit{Euclidean gradient}: \quad \nabla \mathtt{F}(\mathbf{p}_i) \triangleq 2 \sum_{n=1}^{N} \boldsymbol{\alpha}_n \mathcal{D}_{i,n} \frac{\partial \mathcal{D}_{i,n}}{\partial \mathbf{p}_i}. \tag{24b}$$

We denote $\mathcal{D}_{i,n} := \mathcal{D}(\mathbf{p}_i, \mathbf{p}_n)$ for simplicity.

The algorithm for imitating rhythmic trajectory on a facial surface using structured prediction is summarized in Alg. 1.

V. Adaptation of Learned Skills to Treat New Faces

As a central topic in robot learning from demonstration, the adaptation of motor skills is crucial for robots to accomplish the task in an environment that is different from the one in the demonstration. In our application, the objective is to equip the robot with the capacity to execute treatment on subjects who were not present during the demonstration. To achieve adaptation of treatment skills, our strategy is to modulate trajectory in accordance with changes in the shape of human faces. Wherein, human facial shapes are captured by key facial features such as control points. Consequently, based on changes in the key facial features, we acquire rules for the laser generator's trajectory adaptation.

We propose to tackle the problem of discovering the rules for trajectory adaptation by *feature-based registration*, whose goal is to align a source surface (i.e., the face in the demonstration) with a target surface (i.e., the face during the adaptation) given landmark points. Specifically, we consider the *nonrigid* transformation, which permits soft deformations of the point sets. A common technique for non-rigid registration is to parametrize the transformation by leveraging *thin-plate spline* [47]. While nonrigid registration has gained attention in robot imitation learning for modulating learned trajectories towards new environments, the applications have been limited to deterministic scenarios [48].

Algorithm 1: Imitation of Geometric Trajectory with Rhythm by Structured Prediction

```
Scan the subject's face for point cloud dataset $\mathbb{F}$;
Determine the projection plane perpendicular to the
  eigenvector with the minimum eigenvalue of (19);
Specify grid $\{\{\boldsymbol{l}_{vi}^{p}\}_{i=1}^{nv}, \{\boldsymbol{l}_{hj}^{p}\}_{j=1}^{nh}\}$ for facial partition;
Compute the centers and radius as per (17);
Collect the demonstrated trajectory for dataset $\mathbb{T}$;
Define the kernel function $k(\cdot,\cdot)$ and hyperparameters;
Choose the regularization term $\lambda$ and the step size $\eta$;
for $t = t_{\mathtt{init}}, \dots, t_{\mathtt{end}}$ do
    Input: Query point $t$;
    Calculate the scores $\boldsymbol{\alpha}(t)$;
    repeat
        for $n = 1, \dots, N$ do
            Determine the intersection points with the
              grid lines on the projection plane;
            Map the intersection points to the surface
              manifold for segmentation points;
            Calculate the geodesic distance to each
              training point $\mathcal{D}(\mathbf{P}(t), \mathbf{P}_n)$ as per (20);
        Compute the Riemannian gradient as per (23);
        Update the prediction as per (21);
    until convergence;
    Output: Prediction $\mathbf{P}(t)$;
```

In the following, we present a method for achieving *probabilistic* trajectory adaptation with nonrigid registration. By incorporating additional probabilistic information in the form of associated covariance matrices, our method offers greater flexibility in devising the transformation. Specifically, trajectory points with higher variances will be allowed to deviate more while those with lower variances will deflect less after the transformation. The proposed probabilistic trajectory adaptation technique is particularly relevant to our application of skincare, where covariance matrices could be utilized to encode local skin conditions. For example, when the adapted trajectories cover key facial regions, such as the eyes, the associated covariance matrices can be leveraged to provide additional flexibility in modulating the imitated trajectories, allowing them to avoid these critical areas.

Formally, assume that there are K landmark points for the source and target facial surface which are represented by $\{\hat{\boldsymbol{\xi}}_{l_i}\}_{i=1}^{K}$ and $\{\hat{\boldsymbol{\xi}}_i'\}_{i=1}^{K}$, respectively, with the points index stored as $\{l_1, \dots, l_K\}$. In addition, the associated covariance matrices, denoted by $\{\hat{\boldsymbol{\Sigma}}_{l_i}\}_{i=1}^{K}$, act as tunable parameters in our algorithm, allowing for fine-tuning of the adapted trajectory. Our goal is to find a transformation rule $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}^D$ such that each original source point can be mapped to its corresponding target point. The problem of determining the transformation function $\boldsymbol{f}$ can be cast as the following optimization problem:

$$E(\boldsymbol{f}) = \sum_{i=1}^{K} \left(\hat{\boldsymbol{\xi}}_i' - \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{l_i})\right)^{\mathsf{T}} \hat{\boldsymbol{\Sigma}}_{l_i}^{-1} \left(\hat{\boldsymbol{\xi}}_i' - \boldsymbol{f}(\hat{\boldsymbol{\xi}}_{l_i})\right) + \lambda \|\boldsymbol{f}\|_{\mathtt{tps}}^2 \tag{25}$$

where $E(\cdot)$ is the so-called bending energy. As a slight abuse of notation, $\lambda > 0$ is a smoothing parameter that balances the trade-off between smoothness and goodness-of-fit. Besides, $\|\boldsymbol{f}\|^2_{\texttt{tps}}$ denotes the thin plate spline regularization term, which imposes a soft constraint on smoothness in order to control the behavior of the mapping. It is defined by the space integral of the square of the second-order derivatives [47]:

$$\|\boldsymbol{f}\|^2_{\texttt{tps}} = \int_{\mathbb{R}^3} \left\| \frac{\partial^2 \boldsymbol{f}}{\partial^2 \hat{\boldsymbol{\xi}}} \right\|^2_{\texttt{Frob}} d\hat{\boldsymbol{\xi}}. \tag{26}$$

Remarkably, there exists a unique minimizer, and the analytical solution to $\boldsymbol{f}$ is composed of an affine or rigid part and a nonlinear or nonrigid part, as shown in [47]:

$$\boldsymbol{f}(\hat{\boldsymbol{\xi}}_{l_i}) = \underbrace{\mathbf{B}\hat{\boldsymbol{\xi}}_{l_i}}_{\text{Affine part}} + \underbrace{\boldsymbol{\omega}\boldsymbol{\rho}(\hat{\boldsymbol{\xi}}_{l_i})}_{\text{Nonlinear term}}, \tag{27}$$

where $\mathbf{B} \in \mathbb{R}^{(D+1)\times(D+1)}$ accounts for the affine transformation and $\boldsymbol{\omega} \in \mathbb{R}^{(D+1)\times K}$ contains the warping coefficients enabling non-affine deformation. It should be noted that we augment each point with an intercept term for homogeneous coordinates, i.e., $\hat{\boldsymbol{\xi}}_{l_i} \leftarrow [\hat{\boldsymbol{\xi}}^{\mathsf{T}}_{l_i}\ 1]^{\mathsf{T}}$ in order to include data offset conveniently. Each entry of the vector $\boldsymbol{\rho}(\hat{\boldsymbol{\xi}}_{l_i}) \in \mathbb{R}^K$ is a radial basis function for thin plate spline, and the center of the basis function in the j-th entry is $\hat{\boldsymbol{\xi}}_{l_j}$. More precisely, each entry of the vector is defined by

$$\boldsymbol{\rho}_j(\hat{\boldsymbol{\xi}}_{l_i}) = -\left\| \hat{\boldsymbol{\xi}}_{l_j} - \hat{\boldsymbol{\xi}}_{l_i} \right\|^2. \tag{28}$$

To solve for $\mathbf{B}$ and $\boldsymbol{\omega}$, we substitute (27) into (25). Then, the bending energy becomes

$$\begin{aligned} E &= \texttt{sum}\big((\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\boldsymbol{\omega}^{\mathsf{T}} - \widetilde{\boldsymbol{\xi}}_l\mathbf{B}^{\mathsf{T}}) \odot \widetilde{\mathbf{Q}} \odot (\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\boldsymbol{\omega}^{\mathsf{T}} - \widetilde{\boldsymbol{\xi}}_l\mathbf{B}^{\mathsf{T}})\big) \\ &\quad + \lambda\texttt{Tr}(\boldsymbol{\omega}\mathbf{G}\boldsymbol{\omega}^{\mathsf{T}}) \\ &= \|\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot (\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\boldsymbol{\omega}^{\mathsf{T}} - \widetilde{\boldsymbol{\xi}}_l\mathbf{B}^{\mathsf{T}})\|^2_{\texttt{Frob}} + \lambda\texttt{Tr}(\boldsymbol{\omega}\mathbf{G}\boldsymbol{\omega}^{\mathsf{T}}) \end{aligned} \tag{29}$$

where $\odot$ represents the element-wise product, $\mathbf{G} \in \mathbb{R}^{K\times K}$ is constructed by $\mathbf{G}_{i,j} = \boldsymbol{\rho}_j(\hat{\boldsymbol{\xi}}_{l_i})$, and we have the concatenated terms expressed as:

$$\begin{aligned} &\widetilde{\boldsymbol{\xi}}_l = \left[\hat{\boldsymbol{\xi}}_{l_1}, \ldots, \hat{\boldsymbol{\xi}}_{l_K}\right]^{\mathsf{T}}, \quad \widetilde{\boldsymbol{\xi}}' = \left[\hat{\boldsymbol{\xi}}'_1, \ldots, \hat{\boldsymbol{\xi}}'_K\right]^{\mathsf{T}}, \\ &\text{and} \quad \widetilde{\mathbf{Q}} = \texttt{blockdiag}(\hat{\boldsymbol{\Sigma}}^{-1}_{l_1}, \ldots, \hat{\boldsymbol{\Sigma}}^{-1}_{l_K}). \end{aligned} \tag{30}$$

In view of the optimization objective, it is not straightforward to calculate $\mathbf{B}$ and $\boldsymbol{\omega}$. Therefore, the QR decomposition is employed to separate the affine and non-affine warping spaces [47]. Specifically, we have

$$\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot \widetilde{\boldsymbol{\xi}}_l = [\mathbf{Q}_1\ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}. \tag{31}$$

By observing that $(\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot \widetilde{\boldsymbol{\xi}}_l)^{\mathsf{T}}\boldsymbol{\omega}^{\mathsf{T}} = \mathbf{0}$, we set

$$\boldsymbol{\omega}^{\mathsf{T}} = \mathbf{Q}_2\gamma. \tag{32}$$

Subsequently, the bending energy can be expressed as

$$\begin{aligned} E =& \|\mathbf{Q}^{\mathsf{T}}_2\big(\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot (\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\mathbf{Q}_2\gamma)\big)\|^2_{\texttt{Frob}} \\ &+ \|\mathbf{Q}^{\mathsf{T}}_1\big(\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot (\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\mathbf{Q}_2\gamma) - \mathbf{R}\mathbf{B}^{\mathsf{T}}\big)\|^2_{\texttt{Frob}} \\ &+ \lambda\texttt{Tr}(\gamma^{\mathsf{T}}\mathbf{Q}^{\mathsf{T}}_2\mathbf{G}\mathbf{Q}_2\gamma). \end{aligned} \tag{33}$$

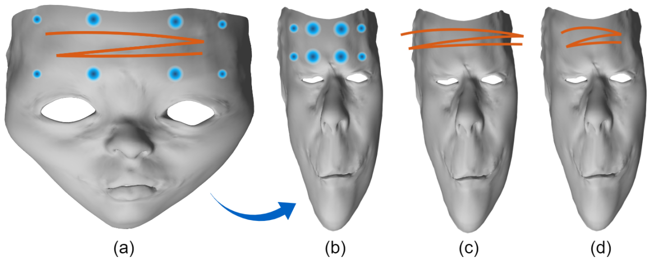


Fig. 5. Illustration of probabilistic nonrigid registration. (a) The original shape of the trajectory on the demonstration face. (b) The adaptation face where control points are depicted in blue with the radius indicating the covariance matrix contour. (c) Transformed trajectory shape without probabilistic registration, which exceeds the boundary of the forehead. (d) Transformed trajectory shape with probabilistic registration which ensures that the trajectory lies within the boundary of the forehead.

```
Algorithm 2: Skill Adaptation toward Treating Novel Faces by Probabilistic Nonrigid Registration
Choose source control points ${\hat{\boldsymbol{\xi}}_{l_i}}_{i=1}^K$;
Scan the new subject's face for the point cloud dataset;
Determine target control points ${\hat{\boldsymbol{\xi}}'_i}_{i=1}^K$;
Specify the associated covariance matrices ${\hat{\boldsymbol{\Sigma}}_{l_i}}_{i=1}^K$;
Concatenate terms for $\widetilde{\boldsymbol{\xi}}_l$, $\widetilde{\boldsymbol{\xi}}'$, and $\widetilde{\mathbf{Q}}$ as per (30);
Perform QR decomposition as per (31);
Solve for the affine transformation $\mathbf{B}$ and nonlinear
  warping coefficients $\boldsymbol{\omega}$ as per (34) and (32);
Input: Demonstrated trajectory point $\mathbf{P}(t)$;
Output: Adaptation point $\boldsymbol{f}(\mathbf{P}(t))$ as per (27);
```

Finally, the solution for $\mathbf{B}$ is given by

$$\mathbf{B}^{\mathsf{T}} = \mathbf{R}^{-1}\mathbf{Q}^{\mathsf{T}}_1\big(\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot (\widetilde{\boldsymbol{\xi}}' - \mathbf{G}\mathbf{Q}_2\gamma)\big), \tag{34}$$

where we have

$$\gamma = \big(\mathbf{Q}^{\mathsf{T}}_2\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot \mathbf{G}\mathbf{Q}_2 + \lambda\mathbf{I}\big)^{-1}\mathbf{Q}^{\mathsf{T}}_2\widetilde{\mathbf{Q}}^{\frac{1}{2}} \odot \widetilde{\boldsymbol{\xi}}'. \tag{35}$$

Moreover, the expression for $\boldsymbol{\omega}$ can be readily obtained by combining (32) and (35).

The effect of probabilistic non-rigid registration is shown in Fig. 5, where, as an illustration, we use two caricature faces [49]. The additional regulation can ensure that the adapted trajectory upon non-rigid registration can properly evolve on the new subject's face.

The steps for the adaptation of laser shot traces towards treating a new face are summarized in Alg. 2.

VI. Results

In this section, we present the results of numerical and experimental studies. Firstly, we showcase the effectiveness of kernel functions in capturing motion patterns (Section VI-A), the spherelet technique in calculating the geodesic distances (Section VI-B), and non-rigid registration in adapting treatment trajectories (Section VI-C). Then, We provide a comparison with state-of-the-art methods for learning periodic geometry-structured trajectories in Section VI-D. Finally, in Section VI-E, we conduct real-world experiments on robotic cosmetic dermatology using photorejuvenation.

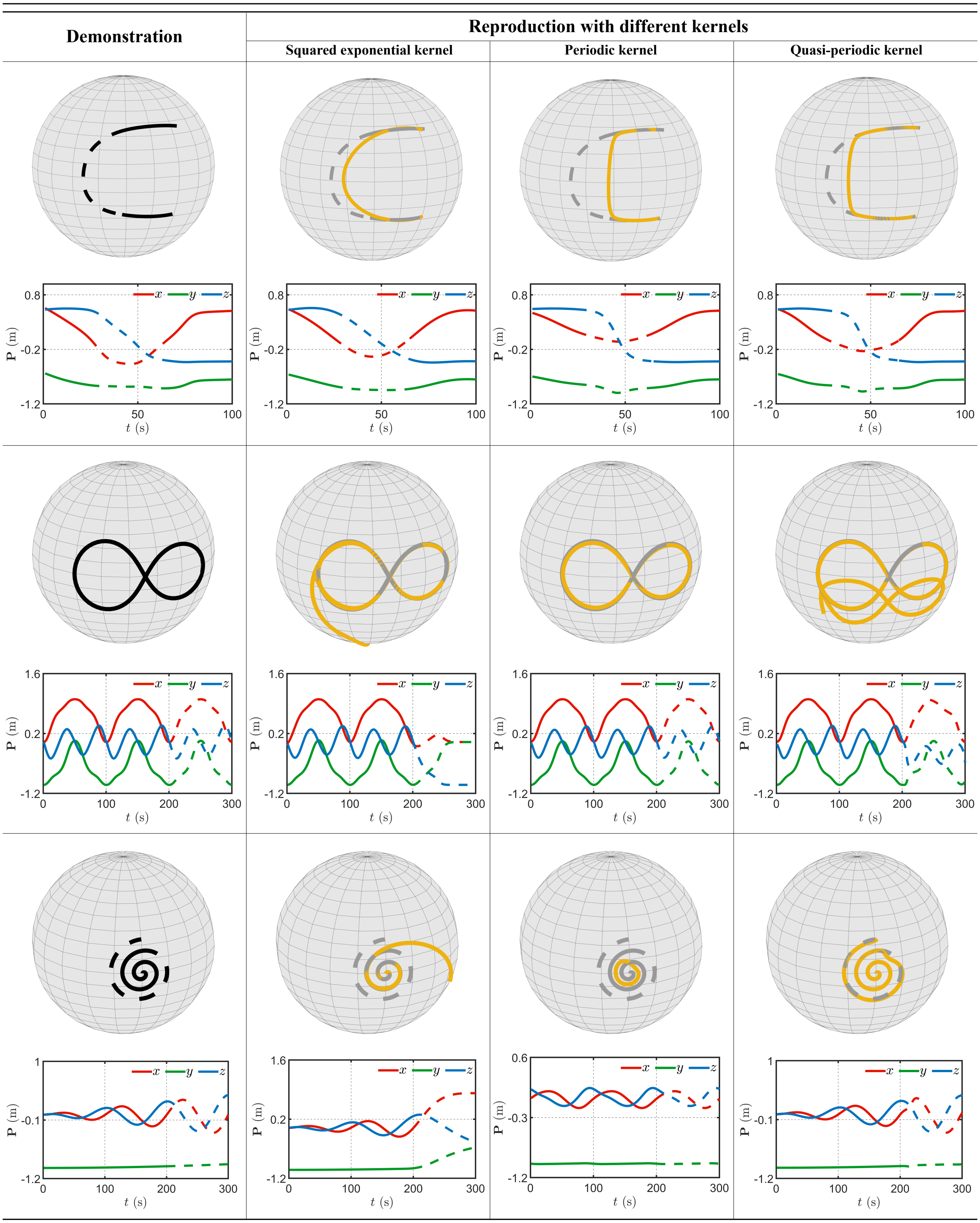


Fig. 6. Illustration of learning different motion patterns on a manifold with different kernel functions, where the demonstration trajectory is shown in the *first column*, the motion skills reproduction is achieved by the SE kernel (*second column*), the PER kernel (*third column*), and the QP kernel (*fourth column*). The representative motion patterns to capture are stroke-based trajectory (*top row*), periodic trajectory (*middle row*), and quasi-periodic trajectory (*bottom row*), where the solid lines denote the reproduction part and the dashed lines denote the generalization part. For each imitation instance, we first plot the trajectory evolution on the manifold, underneath which we show the temporal evolution of each trajectory dimension.

TABLE I
IMITATION ERRORS OF REPRODUCTION AND GENERALIZATION

	C-shape		**Infinity**		**Spiral**	
(1e−3)	Tra.	Gen.	Tra.	Gen.	Tra.	Gen.
SE	**25.0**	**76.3**	14.1	1258.5	12.3	918.5
PER	27.3	217.7	**3.4**	**3.6**	63.5	189.1
QP	25.4	162.4	5.7	293.9	**2.9**	**37.1**

A. *Effects of Kernel Functions*

To evaluate the learning effects when using different kernel functions for the estimator (7), we construct the $\boldsymbol{\alpha}$ scores based on several representative kernel functions to imitate different types of demonstration trajectories, including the Squared Exponential kernel (SE) $k_{\texttt{SE}}$ as in (8), the Periodic kernel (PER) $k_{\texttt{PER}}$ as in (11), and the Quasi-Periodic kernel (QP) $k_{\texttt{QP}}$ as in (13). Three typical patterns of demonstration trajectory, namely stroke-based trajectory, periodic trajectory, and quasi-periodic trajectory are chosen. These trajectories are designed to evolve on a unit sphere whose center coincides with the origin of the coordinate. In each learning scenario, we split the whole trajectory into two parts to perform both movement primitive training and skill generalization testing.

We specify a C-shaped, an infinity symbol ∞-shaped, and a spiral-shaped trajectory for stroke-based, periodic, and quasi-periodic trajectory imitation, respectively. In the case of learning with the SE kernel, we set $\sigma_s = 5$ and $l_s = 20$. When using the PER kernel, we set $\sigma_p = 1$, $l_p = 0.5$, and $p = 150$. As for the QP kernel, the parameters for the composing SE and PER kernels are the same as those used in the standalone cases. Besides, the regularization parameter is set to be $\lambda = 0.01$.

The learning results are shown in Fig. 6. It can be observed that the performance of each kernel function varies across different learning situations. Particularly, one kernel function outperforms the others for a specific learning task. This underscores the necessity of properly designing and selecting kernel functions to synthesize the $\boldsymbol{\alpha}$ scores for a given imitation task.

Certainly, for each task, there may exist a more optimal kernel function than the one currently used, which could potentially yield better learning performance. For instance, it has been observed that there is a relatively considerable deviation between the generalization trajectory and the demonstrated trajectory, even when using the squared exponential kernel. This suggests that an even more suitable kernel function may be required for this particular task.

For quantitative assessment, we define the following imitation metric based on the average of the accumulation of prediction errors. To evaluate the reproduction behavior, we have

$$\mathcal{C}_{\mathrm{T}} = \frac{1}{N}\sum_{n=1}^{N} \Delta(\mathbf{s}(t_n), \mathbf{p}(t_n)), \tag{36}$$

where $\Delta(\mathbf{s}(t_n), \mathbf{p}(t_n)) = \arccos(\mathbf{s}(t_n)^\intercal \mathbf{p}(t_n))$ in the case of spherical distance. Similarly, the evaluation metric for examining the generalization effects is given by

$$\mathcal{C}_{\mathrm{G}} = \frac{1}{M-N}\sum_{m=N+1}^{M} \Delta(\mathbf{s}(t_m), \mathbf{p}(t_m)), \tag{37}$$

where M represents the total number of prediction steps during generalization. The obtained numerical results are summarized in Table I. It can be numerically seen that the SE kernel achieves the best performance for the C-shaped trajectory, the PER kernel exhibits the optimal performance for the infinity symbol, and the QP kernel provides the most effective results for the spiral-shaped trajectory.

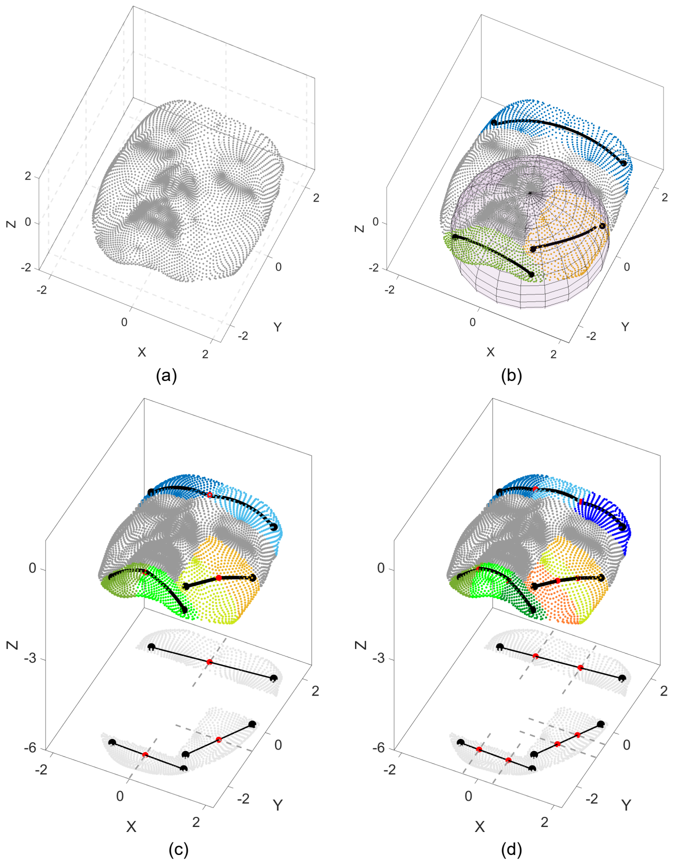


Fig. 7. Illustration of geodesic distance calculation using the spherelet approach on the forehead, cheek, and chin of Nefertiti's face. (a) The plot of point cloud data of Nefertiti's face on which we calculate the geodesic distances with (b) one spherelet, where we denote the terminal points with black dots and exemplify the osculating sphere for the cheek in purple, (c) two spherelets, where we denote the segmentation and intersection points with red dots and plot grid lines with dashed grey lines, and (d) three spherelets.

TABLE II
GEODESIC DISTANCE CALCULATION RESULTS.

	Forehead		**Cheek**		**Chin**	
Method	mm	s	mm	s	mm	s
IsoMap	3.386	0.820	2.108	0.389	2.562	0.373
MDS	3.3746	0.101	2.0489	0.077	2.4433	0.071
Heat	3.448	0.699	2.1831	0.626	2.564	0.599
Eikonal	3.3748	0.603	2.0583	0.501	2.4433	0.459
OneSph. (ours)	3.2992	0.018	1.9616	0.017	2.4876	0.018
TwoSph. (ours)	3.3128	0.026	2.0070	0.023	2.5196	0.023
ThreeSph. (ours)	3.3308	0.025	2.0223	0.023	2.5370	0.027

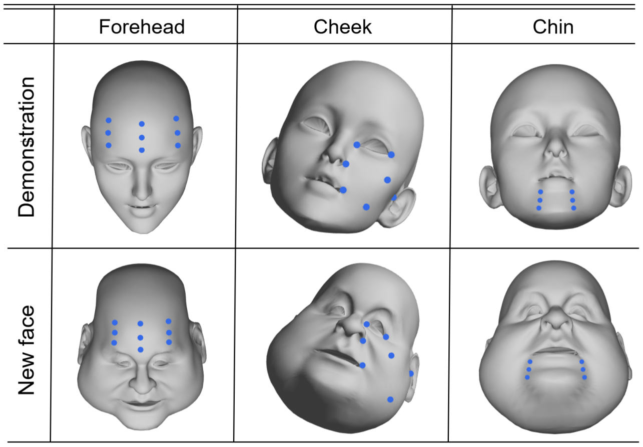


Fig. 8. Illustration of landmark point positions as denoted in blue on the forehead, cheek, and chin of demonstration and new faces, respectively.

B. Geodesic Distance Calculation

In this section, we showcase the efficacy of the spherelets-based approach for calculating geodesic distance, using the face of *Nefertiti* as an illustrative example (refer to Fig. 7(a)). While an explicit analysis of the approximation error may be challenging, our goal is to empirically calculate and compare the geodesic distance on Nefertiti's facial surface, as represented by the corresponding point cloud data, with other approaches. To begin with, we align the facial data points to ensure that the least principal component aligns parallel to the z-axis, and the face surface is symmetrically positioned with respect to the y-axis. For the projection plane, we select $z = -6$, which is parallel to the x-y plane. In each region of the forehead, cheek, and chin, we approximate the surface patch with one spherelet, two spherelets, and three spherelets, respectively. The geodesic distance calculated in each region using different numbers of spherelets is shown in Fig. 7(b)-(d).

For comparison, we also calculate the geodesic distances between the same endpoints using other methods, including ISOMAP (with the Floyd algorithm for pairwise geodesic distances), MDS, Heat Flow, and the Eikonal approach. Besides comparing the geodesic distances, we also record the total required running time. Table II summarizes the comparison results. It can be seen that our approach achieves on-par performance in the meanwhile possessing the merit of considerably reducing the computational time.

C. Skill Adaptation to New Faces

Here we evaluate the effectiveness of the probabilistic non-rigid registration technique. The goal is to justify the necessity of imposing additional regulation effects when performing an adaptation of a trajectory from the demonstration face to a new face. We investigate the adaptation behaviors on the regions of the forehead, cheek, and chin whose shapes are determined using nine, seven, and six landmarks, respectively. We arrange the landmarks in a grid formation on the forehead and chin, while positioning the cheek landmarks along the eye, nose, and ear. A visual illustration of the landmark placement is provided in Fig. 8.

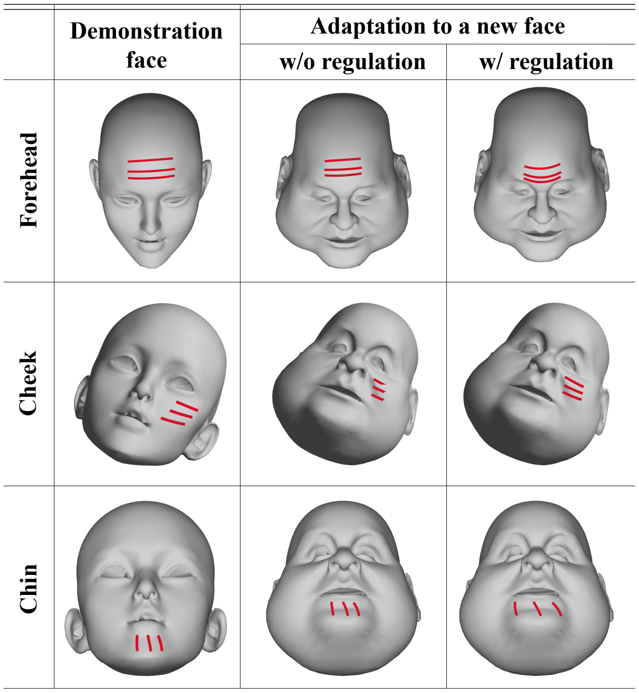


Fig. 9. Illustration of reference trajectory on the demonstration face (*left column*), its reproduction on a new face without regulation (*middle column*), and its reproduction on a new face with (*right column*).

During the evaluations, we set the smoothing parameter to be $\lambda = 0.5$. For trajectory modulation on the forehead, the control point besides the hairline is assigned with a weight matrix of $20\mathbf{I}$, thus prohibiting the adapted trajectory from the potential interference with the subject's hair. For adaptation on the cheek region, we improve the occlusion between the adapted trajectories and the cheek surface by setting the cheek tip control point with a weight of $1e2\mathbf{I}$. Regarding the chin region, we choose the weight matrix as $10\mathbf{I}$ for the critical control points, stretching the demonstration trajectories to fit the wide chin of the new face. An illustration of the regulation effects is depicted in Fig. 9. It can be observed that with the added regulation of the covariance matrices, the adapted trajectory becomes more flexible, enabling it to better meet the requirements that arise when adapting to new faces. Additionally, we assess the adaptation performance across a wide range of face shapes. The adaptation is applied to the forehead, cheek, and chin regions, respectively. The selection of landmarks for adaptation is consistent with those in Fig. 8. A schematic illustration of the adaptation performance is provided in Fig. 10.

To quantitatively illustrate the effectiveness of adaptation performance, we propose to calculate the occlusion degree $\mathcal{C}_a$ to reflect the degree of matching between the adapted trajectories $\{\mathbf{p}_n\}_{n=1}^N$ and the new subject's face surface $\mathbb{F}_a$. Formally, the occlusion degree is given by averaging the minimum distance between the adapted trajectory and the subject's facial point cloud:

$$\mathcal{C}_a = \frac{1}{N}\sum_{n=1}^{N}\min_{\boldsymbol{\xi}\in\mathbb{F}_a}\|\boldsymbol{\xi} - \mathbf{p}_n\|_2.$$

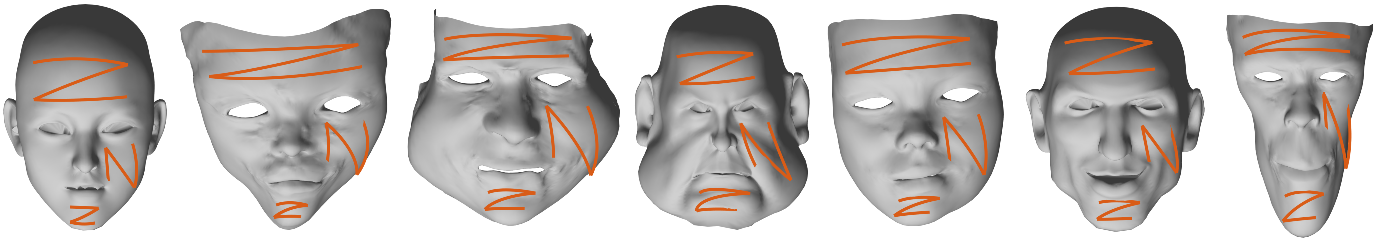


Fig. 10. Illustration of the adaptation results from the demonstration face (*left*) to more new faces in the regions of the forehead, cheek, and chin.

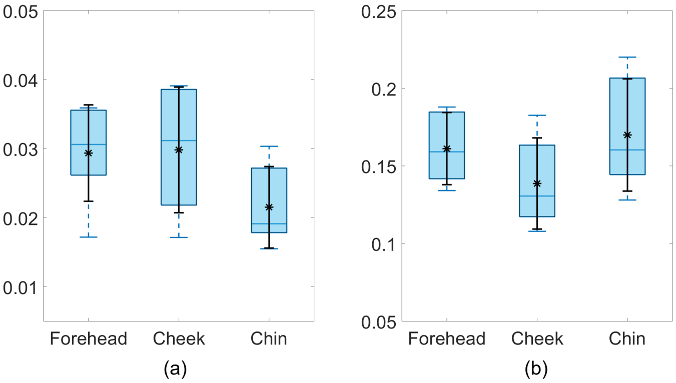


Fig. 11. Statistical analysis of adaptation performance using (a) the occlusion metric and (b) the Hausdorff metric, where the blue boxes denote the box plot and the black bars denote the standard deviation.

TABLE III
LEARNING ERRORS OF DIFFERENT MOTION IMITATION ALGORITHMS.

	′1′	′0′	′8′	′🦇′
GA-DMP [18]	0.037	0.041	0.057	0.023
R-LQT [19]	0.014	0.022	0.031	0.036
Ori-ProMP [20]	0.032	0.016	0.012	0.052
Our approach	**0.007**	**0.008**	**0.002**	**0.012**

In addition to our proposed occlusion metric, which quantifies the average degree of matching, we also utilize the well-known Hausdorff metric, which measures the greatest distance between a point in one set and its nearest point in another set. The results of the statistical analysis, including both the box plot and standard deviation for the two metrics, are presented in Fig. 11. It can be observed that the adaptation errors are relatively low, indicating the effectiveness of our developed adaptation strategy.

The mismatch could result in either under- or over-bending of the adapted trajectory, which may affect how well it conforms to the shape of the new face, leading to varying outcomes. When the trajectory is over-bent, it may cause the laser pulses to penetrate the skin. Besides, this could lead to an unintended concentration of laser energy. To ensure a safe treatment in such cases, it is essential to carefully regulate the laser energy dosage [3].

D. Comparison with Baselines

In this section, we compare our proposed approach with several representative state-of-the-art movement primitive algorithms in terms of imitating geometric motion with periodicity. Specifically, for benchmark baselines, we select Geometry-Aware Dynamics Movement Primitives (GA-DMP) [18], Riemannian Linear Quadratic Tracking (R-LQT) [19], and Orientation Probabilistic Movement Primitives (Ori-ProMP) [20] that represent the geometry-aware counterparts of the classical dynamics movement primitives, linear quadratic tracking, and probabilistic movement primitives. We specify four periodic demonstration trajectories that evolve on a unit sphere to imitate, namely number trajectories $'1'$, $'0'$, $'8'$, and a complex motion pattern of a bat contour.

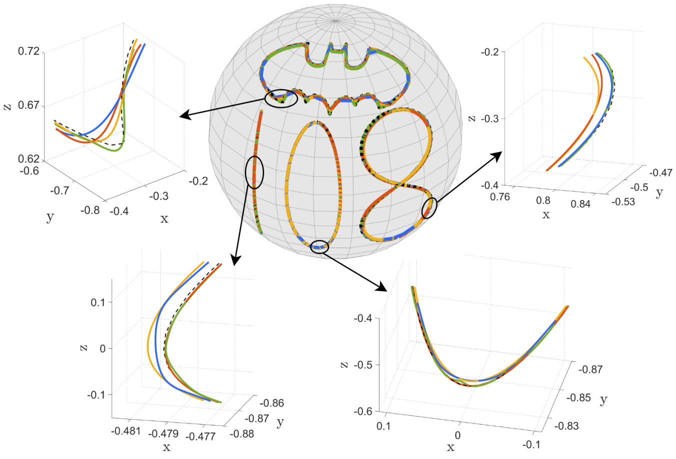


Fig. 12. Schematic illustration of the learning results with different imitation algorithms. The demonstrated $'1'$, $'0'$, and $'8'$-shaped periodic trajectories (dashed black line) are separately learned by our approach (green), GA-DMP(yellow), R-LQT (red), and Ori-ProMP (blue), respectively.

For our approach, we use the PER kernel, and the relevant parameters are selected to be $\sigma_p = 1$, $l_p = 2$, and $p = 100$. We use 30 Von-Mises basis functions per dimension for GA-DMP, which is defined as $\exp(\cos(2\pi(t - c_i))/h)$, where $h = 5$ defines the width of the basis function and c_i determines the uniformly distributed center of the i-th basis function within the period range. For imitating with R-LQT, we set the penalization matrix on the control inputs to be $0.2\mathbf{I}$ and the time window is 30 steps. Regarding Ori-ProMP, the number of basis functions is 30 and the learning rate is 0.05.

The learning performances by all the algorithms are depicted in Fig. 12, which implies that all the algorithms can truthfully reproduce the demonstrated trajectories. We then quantify the imitation performances based on the imitation metric of (36), and the obtained numerical results are summarized in Table III, revealing that our approach exhibits a

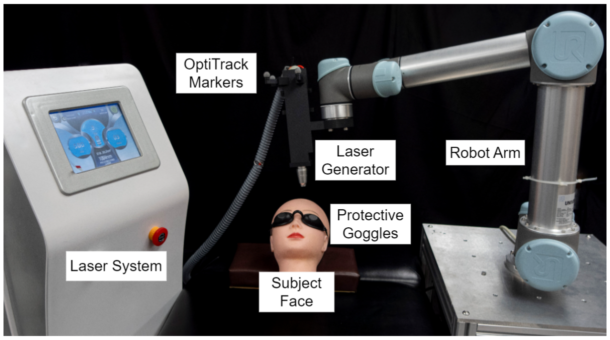


Fig. 13. Illustration of the experimental setup for performing robotic cosmetic dermatology using photorejuvenation.

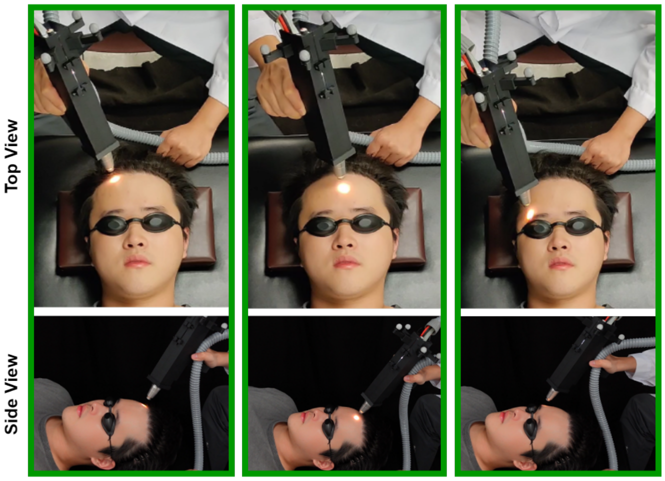


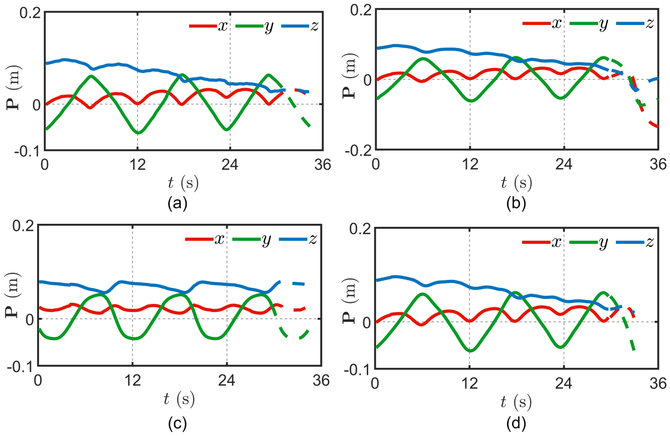
Fig. 14. Snapshots of the demonstration procedure where the treatment motion on a human forehead is transferred by passive observation.

superior level of imitation fidelity.

E. Skin Photorejuvenation Experiments

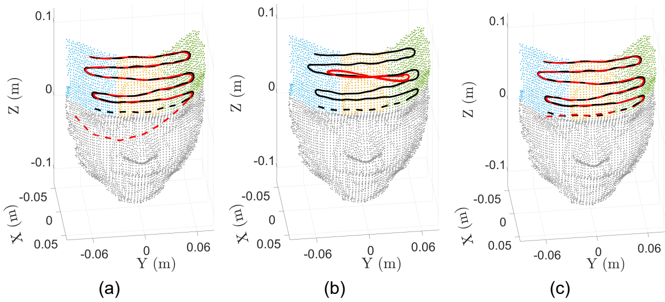
In this section, we conduct real experiments to learn the treatment trajectory of skin photo-rejuvenation from demonstrations. Fig. 13 presents the overall experimental setup. A UR5 robot arm from Universal Robots is employed, and a laser cosmetic instrument fixed with a 3D-printed fixation case is attached to its end-effector. For the sensing system, we use an Intel RealSense D405 RGBD camera to scan the subject's face for the point cloud data. The orientation of the robot end-effector is adjusted to be perpendicular to the face surface, based on calculations from the corresponding point cloud data.

Also, the OptiTrack motion capture system is used to record the demonstrated trajectory with the markers attached to the laser cosmetic instrument. We would like to align the instrument perpendicularly to the skin surface, which can be readily accomplished through point cloud data processing. Therefore, we will focus only on the position data. During the motion data collection process, the interaction points between the laser ray and the face surface are determined by tracking the movement of a virtual point fixed at a constant distance from the laser shooter's tip. For the laser system, the pulse frequency of the laser shots is controlled with an on-board relay. We set constant values for the pulse frequency with $10\,\mathrm{Hz}$. In addtion, we set the thermal power as $100\,\mathrm{mJ}$, and spot diameter as $10\,\mathrm{mm}$. Throughout the experiment, the involved human subjects[1] exposed to the laser pulses are asked to wear a pair of protective goggles.



(a) (b) (c) (d)

Fig. 15. Illustration of temporal evolutions of the trajectory of (a) demonstration, and reproduction with the (b) SE kernel (c) PER kernel, and (d) QP kernel where solid lines denote the reproduction trajectory and dashed lines denote the generalization trajectory.



(a) (b) (c)

Fig. 16. Illustration of the learned geometric trajectory (red) from the demonstrated motion (black) by (a) SE kernel (b) PER kernel, and (c) QP kernel where solid lines denote the reproduction trajectory and dashed lines denote the generalization trajectory, and the segmentation with spherelets is represented by different colors.

During the skill transfer phase, the point cloud data of the subject's face is first obtained by the depth camera. Subsequently, a dermatologist personally demonstrates the desired treatment trajectory by manipulating the laser cosmetic instrument, which is recorded by the motion capture cameras. During the treatment, the involved subject is required to remain in a still supine position. The demonstration procedure by passive observation can be seen in Fig. 14.

We first evaluate the reproduction performance in terms of treating the same subject as in the demonstration. Likewise, we compare the learning performance using the SE, PER, and QP kernels to learn the demonstrated motion pattern. The learning results are shown in Fig. 15. Besides, the trajectory evolution on the facial point cloud is shown in Fig. 16. It can be seen that the trajectory prediction by the QP kernel attains

[1] Ethics Approval Reference Number: HSEARS20201202001, Human Subjects Ethics Sub-committee, Departmental Research Committee, The Hong Kong Polytechnics University, Hong Kong.

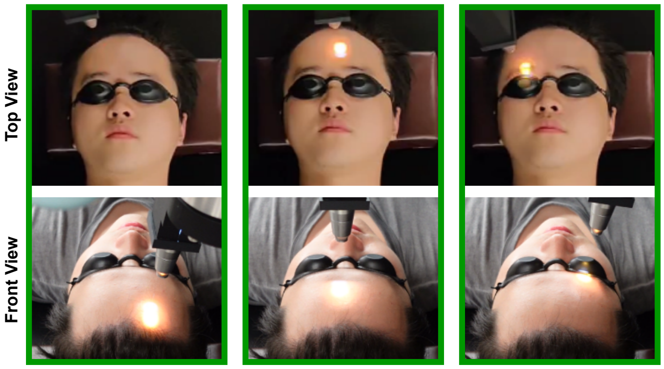


Fig. 17. Snapshots of reproduction of the treatment skill on the demonstration face with a robot arm.

TABLE IV
LEARNING ERRORS WITH DIFFERENT KERNELS.

	SE	PER	QP
Reproduction	0.001	0.03	**0.0008**
Generalization	0.125	0.06	**0.01**

the best performance as the demonstration trajectory exhibits a quasi-periodic motion pattern. The quantitative evidence of the learning performances by all three kernels is summarized in Table IV, where the evaluations for motion reproduction and generalization are conducted regarding (36) and (37), respectively. Once the trajectory is learned with the QP kernel, we then reproduce the treatment trajectory on the subject, as shown in Fig. 17.

Finally, we study the issue of adapting the learned trajectory from the face in the demonstration to the faces that were unseen before. When a new subject comes to receive the treatment, we first scan the face for the corresponding point cloud data. Afterward, we select the landmarks on the point cloud that are used for transferring the learned demonstration trajectory. Fig. 18 shows the point cloud for the demonstration face and two new subjects' faces. The red dots mark the source control points on the demonstration face as well as the target control points on the new faces. The snapshots illustrating the adaptation treatment procedure for two new subjects are shown in Fig. 19. The upper two rows depict one subject, while the bottom two rows represent the other subject.

The adaptation performance for the new subjects is then shown using the occlusion degree, which is calculated to be $0.78\,\mathrm{mm}$ and $1.01\,\mathrm{mm}$, respectively. The low values of the occlusion degree indicate that the proposed adaptation method performs satisfactorily when dealing with new subjects' faces.

VII. CONCLUSION

In this paper, we have addressed the issue of robotic cosmetic dermatology using the learning-by-demonstration paradigm. Specifically, we have tackled the challenge of motion imitation through the lens of structured prediction, which is powerful in handling geometry-structured data arising from facial surfaces. Additionally, we have developed an adaptation strategy based on the non-registration technique to treat the facial surface of a new subject, which was not observed during the demonstration. The real-world experiments have shown our proposed method's effectiveness in performing photorejuvenation for cosmetic dermatology.

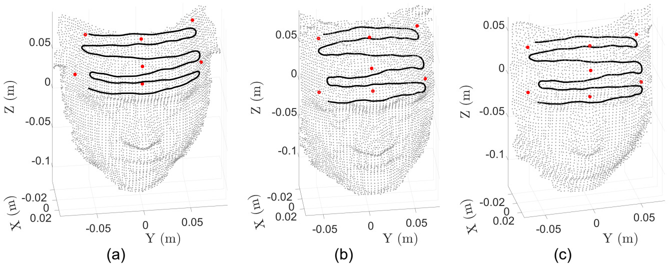


Fig. 18. Illustration of adapting the laser trajectory on the demonstration face (a) towards other unseen faces (b) and (c) where red dots denote the landmarks used for performing trajectory transfer.

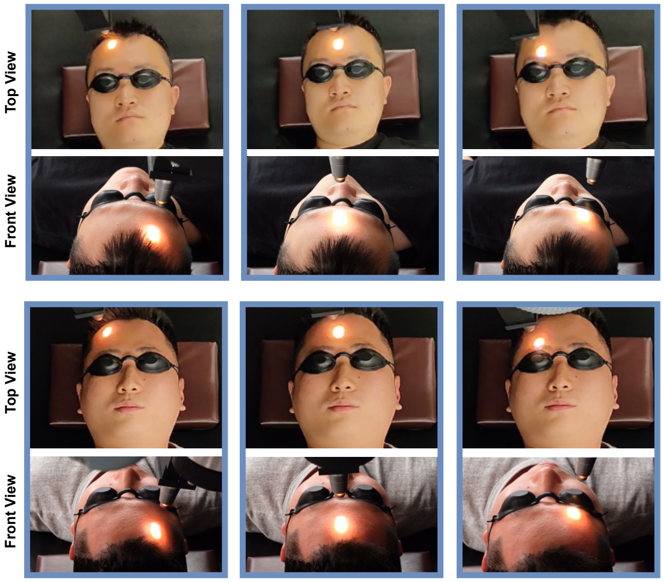


Fig. 19. Snapshots of adaptation of the demonstrated treatment skill to other new faces that are not involved in the demonstration.

There are still some limitations associated with our proposed approach. For instance, in the current setup, we require the human subject to maintain a static position throughout the treatment. Ideally, it would be favorable to incorporate reactive behaviors in the robot to respond to potential movements from the region of interest [50]. Additionally, in this work, we have set the grid lines for facial partitioning to be evenly spaced. It would be interesting to explore a more sophisticated spacing strategy for better approximating the facial manifold. Undoubtedly, the integration of robotic technologies into the beauty industry holds promise, and we plan to continue our efforts in this direction. For future work, we would like to investigate transferring the manipulation skills among different robotic platforms by exploiting the novel PRobabilistically-Informed Motion Primitives (PRIMP) [51]. Also, we intend to conduct further tests to obtain quantitative evidence regarding the administration of thermal doses for enhancing skin conditions; We are currently working on the development of new thermal servoing controls (e.g., as in [52]) for these types

TABLE V
SUMMARY OF KEY NOTATIONS.

	Notation	Description
Math	$(\cdot)^\mathsf{T}$	Transpose operator
	$\otimes$	Kronecker product
	$\odot$	Hadamard product
	$\mathcal{H}$	Hilbert space
	$\circ$	Function composition
	$\mathbf{I}$	The identity matrix
	$\langle \cdot, \cdot \rangle_\mathcal{H}$	Inner product of $\mathcal{H}$
	$\mathbf{\Gamma}$	Parallel transport
	$\mathtt{Exp}$	Exponential mapping
	$\nabla_\mathcal{M}$, ∇	Riemannian, Euclidean gradient
	$\| \cdot \|_\mathtt{Frob}$	Frobenius norm
	$\mathtt{Tr}(\cdot)$	Trace of a matrix
	$\mathtt{blockdiag}(\cdot)$	Block-diagonal concatenation
	$\mathtt{Proj}$	Orthogonal projector
	$\mathbf{Q}, \mathbf{R}$	QR composition matrices
Learning	$\mathbb{T}$	Dataset of demonstration
	$\mathbf{c}$, $\mathbf{c}^{-1}$	Decoding, encoding rule
	$\mathbf{g}$	Surrogate mapping function
	$\mathbf{s}$	Predicted structured output
	$\mathcal{L}$	Surrogate loss
	Δ	SELF loss function
	$\mathbf{V}$	Continuous linear operator
	$\mathbf{\Psi}$	Feature map
	$k(\cdot, \cdot)$	Kernel function
	σ, l	Hyperparameters of kernel
	$\boldsymbol{\mathcal{K}}$	Matrix-valued kernel
	$\mathbf{K}$	Gram matrix
	$\boldsymbol{\alpha}$	Score function
Robot	$\mathcal{S}$	Surface manifold
	$\mathbb{S}^2$	Sphere manifold
	t	Trajectory time stamp
	$\mathbf{p}$	Cartesian position
	T	Trajectory period
	$\mathbf{O}$, r	Center, radius of osculating sphere
	$C(\mathbf{O}, r)$	Algebraic fitting loss
	$\mathbb{F}$	Point cloud of facial surface
	$\boldsymbol{l}_v^p$, $\boldsymbol{l}_h^p$	Vertical, horizontal grid lines
	$\boldsymbol{\xi}^s$, $\boldsymbol{\xi}^p$	Segmentation, intersection points
	$\mathbf{\Pi}_p^s(\cdot)$	Mapping from projection plane to sphere
	$\boldsymbol{f}(\cdot)$	Adaptation rule
	$E(\cdot)$	Bending energy
	$\hat{\boldsymbol{\xi}}$, $\hat{\boldsymbol{\xi}}_i'$	Source, target control points
	$\boldsymbol{\rho}$	Thin-plate spline basis

of laser-based procedures.

APPENDIX A
NOTATION

The notation used throughout the paper is in Table V.

APPENDIX B
PROOF OF PROPOSITION 1

In this section, we show that the behavior of the trajectories in the surrogate space can also lead to the corresponding behavior in the structured space that has a loss function of the geodesic distance. We first illustrate the arithmetic case. For convenience, we denote

$$a+1 := a^+,\ a-1 := a^-, \text{ and } \mathbf{P}(\tau + aT) := \mathbf{P}_a. \tag{38}$$

From (15a), we equivalently have the following

$$\Delta(\mathbf{P}_{a^-}, \mathbf{P}_a) = \Delta(\mathbf{P}_a, \mathbf{P}_{a^+}) = \|\mathfrak{C}(\tau)\|, \tag{39a}$$

$$\Delta(\mathbf{P}_{a^-}, \mathbf{P}_{a^+}) = \Delta(\mathbf{P}_{a^-}, \mathbf{P}_a) + \Delta(\mathbf{P}_a, \mathbf{P}_{a^+}). \tag{39b}$$

By using (1), we can express (39a) as

$$\langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} = \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_{a^+}) \rangle_\mathcal{H}. \tag{40}$$

Denoting $\delta\mathbf{\Psi}_a := \mathbf{\Psi}(\mathbf{P}_a) - \mathbf{\Psi}(\mathbf{P}_{a^-})$, the LHS of (40) can be shown to be

$$\langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} \tag{41a}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}(\mathbf{\Psi}(\mathbf{P}_{a^-}) + \delta\mathbf{\Psi}_a) \rangle_\mathcal{H} \tag{41b}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}\delta\mathbf{\Psi}_a \rangle_\mathcal{H} \tag{41c}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a) - \delta\mathbf{\Psi}_a, \mathbf{V}\delta\mathbf{\Psi}_a \rangle_\mathcal{H} \tag{41d}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_a \rangle_\mathcal{H}, \tag{41e}$$

where we have used the linearity of the inner product and the fact that $\langle \mathbf{\Psi}(\mathbf{p}), \mathbf{V}\mathbf{\Psi}(\mathbf{p}) \rangle_\mathcal{H} = \Delta(\mathbf{p}, \mathbf{p}) = 0$. Also, the RHS of (40) can be shown to be

$$\langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_{a^+}) \rangle_\mathcal{H} \tag{42a}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}(\mathbf{\Psi}(\mathbf{P}_a) + \delta\mathbf{\Psi}_{a^+}) \rangle_\mathcal{H} \tag{42b}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H}. \tag{42c}$$

From (40), we then equate (41e) and (42c), leading to

$$\langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_a \rangle_\mathcal{H} = \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H}. \tag{43}$$

For (39b), its LHS can be written as

$$\langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_{a^+}) \rangle_\mathcal{H} \tag{44a}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a) - \delta\mathbf{\Psi}_a, \mathbf{V}(\mathbf{\Psi}(\mathbf{P}_a) + \delta\mathbf{\Psi}_{a^+}) \rangle_\mathcal{H} \tag{44b}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H} - \langle \delta\mathbf{\Psi}_a, \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} - \langle \delta\mathbf{\Psi}_a, \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H}. \tag{44c}$$

Also, the RHS of (39b) can be written as

$$\langle \mathbf{\Psi}(\mathbf{P}_{a^-}), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} + \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\mathbf{\Psi}(\mathbf{P}_{a^+}) \rangle_\mathcal{H} \tag{45a}$$

$$= \langle \mathbf{\Psi}(\mathbf{P}_a) - \delta\mathbf{\Psi}_a, \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} + \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H} \tag{45b}$$

$$= -\langle \delta\mathbf{\Psi}_a, \mathbf{V}\mathbf{\Psi}(\mathbf{P}_a) \rangle_\mathcal{H} + \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H}. \tag{45c}$$

By equating (44c) and (45c) due to (39b), we then have

$$\langle \delta\mathbf{\Psi}_a, \mathbf{V}\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H} = 0. \tag{46}$$

It can be seen that the condition $\delta\mathbf{\Psi}_a = \delta\mathbf{\Psi}_{a^+} := \delta\mathbf{\Psi}_\tau$ is sufficient to make both (43) and (46) hold, implying that the following arithmetic pattern in the surrogate space leads to (15a).

$$\mathbf{\Psi}(\mathbf{P}_a) = \mathbf{\Psi}(\mathbf{P}_{a^-}) + \delta\mathbf{\Psi}_\tau. \tag{47}$$

For the cumulative case (15b), we equivalently have

$$\Delta(\mathbf{P}_{a^-}, \mathbf{P}_a)/a = \Delta(\mathbf{P}_a, \mathbf{P}_{a^+})/a^+ = \|\mathfrak{C}(\tau)\|, \tag{48a}$$

$$\Delta(\mathbf{P}_{a^-}, \mathbf{P}_{a^+}) = \Delta(\mathbf{P}_{a^-}, \mathbf{P}_a) + \Delta(\mathbf{P}_a, \mathbf{P}_{a^+}). \tag{48b}$$

For (48a), the following can be shown

$$\langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}a^+\delta\mathbf{\Psi}_a \rangle_\mathcal{H} = \langle \mathbf{\Psi}(\mathbf{P}_a), \mathbf{V}a\delta\mathbf{\Psi}_{a^+} \rangle_\mathcal{H}. \tag{49}$$

Likewise, we can also obtain (46) for (48b).

Consequently, it can be seen that the condition $a^+\delta\mathbf{\Psi}_a = a\delta\mathbf{\Psi}_{a^+}$ is sufficient to make both (48a) and (48b) hold. By choosing $\delta\mathbf{\Psi}_a := a\delta\mathbf{\Psi}_\tau$, the following cumulative pattern in the surrogate space then gives rise to (15b).

$$\mathbf{\Psi}(\mathbf{P}_a) = \mathbf{\Psi}(\mathbf{P}_{a^-}) + a\delta\mathbf{\Psi}_\tau. \tag{50}$$

References

[1] M. Muddassir, D. Gómez Domínguez, L. Hu, S. Chen, and D. Navarro-Alarcon, “Robotics meets cosmetic dermatology: Development of a novel vision-guided system for skin photo-rejuvenation,” *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 2, pp. 666–677, 2022.

[2] L. S. Baumann and L. Baumann, *Cosmetic dermatology*. McGraw-Hill Professional Publishing, 2009.

[3] M. Muddassir, G. Limbert, B. Zhang, A. Duan, J.-J. Tan, and D. Navarro-Alarcon, “Model predictive thermal dose control of a robotic laser system to automate skin photorejuvenation,” *IEEE/ASME Transactions on Mechatronics*, vol. 28, no. 2, pp. 737–747, 2023.

[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent advances in robot learning from demonstration,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.

[5] Z. Jin, W. Si, A. Liu, W.-A. Zhang, L. Yu, and C. Yang, “Learning a flexible neural energy function with a unique minimum for globally stable and accurate demonstration learning,” *IEEE Transactions on Robotics*, pp. 1–19, 2023.

[6] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Robot programming by demonstration,” in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Secaucus, NJ, USA: Springer, 2008, pp. 1371–1394.

[7] G. Zhao, C. Zeng, W. Si, and C. Yang, “A human-robot collaboration method for uncertain surface scanning,” *CAAI Transactions on Intelligence Technology*, 2023.

[8] D. Song, J. Park, and Y. J. Kim, “SSK: Robotic pen-art system for large, nonplanar canvas,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 3106–3119, 2023.

[9] R. Liu, W. Wan, K. Koyama, and K. Harada, “Robust robotic 3-D drawing using closed-loop planning and online picked pens,” *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1773–1792, 2021.

[10] B. H. Jafari and N. Gans, “Surface parameterization and trajectory generation on regular surfaces with application in robot-guided deposition printing,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6113–6120, 2020.

[11] A. H. Qureshi, J. Dong, A. Baig, and M. C. Yip, “Constrained motion planning networks X,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 868–886, 2021.

[12] S. Ruan, K. L. Poblete, H. Wu, Q. Ma, and G. S. Chirikjian, “Efficient path planning in narrow passages for robots with ellipsoidal components,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 110–127, 2022.

[13] G. Tiboni, R. Camoriano, T. Tommasi *et al.*, “PaintNet: Unstructured multi-path learning from 3D point clouds for robotic spray painting,” in *IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2023.

[14] A. Duan, R. Camoriano, D. Ferigo, D. Calandriello, L. Rosasco, and D. Pucci, “Constrained DMPs for feasible skill learning on humanoid robots,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–6.

[15] F. Frank, A. Paraschos, P. van der Smagt, and B. Cseke, “Constrained probabilistic movement primitives for robot trajectory adaptation,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2276–2294, 2021.

[16] L. Armesto, J. Moura, V. Ivan, M. S. Erden, A. Sala, and S. Vijayakumar, “Constraint-aware learning of policies by demonstration,” *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1673–1689, 2018.

[17] P. Liu, D. Tateo, H. B. Ammar, and J. Peters, “Robot reinforcement learning on the constraint manifold,” in *Conference on Robot Learning*. PMLR, 2022, pp. 1357–1366.

[18] F. J. Abu-Dakka and V. Kyrki, “Geometry-aware dynamic movement primitives,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4421–4426.

[19] S. Calinon, “Gaussians on Riemannian manifolds: Applications for robot learning and adaptive control,” *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 33–45, 2020.

[20] L. Rozo and V. Dave, “Orientation probabilistic movement primitives on riemannian manifolds,” in *Conference on Robot Learning*. PMLR, 2022, pp. 373–383.

[21] A. Duan, I. Batzianoulis, R. Camoriano, L. Rosasco, D. Pucci, and A. Billard, “A structured prediction approach for robot imitation learning,” *The International Journal of Robotics Research*, vol. 43, no. 2, pp. 113–133, 2024.

[22] M. Koskinopoulou, M. Maniadakis, and P. Trahanias, “Speed adaptation in learning from demonstration through latent space formulation,” *Robotica*, vol. 38, no. 10, pp. 1867–1879, 2020.

[23] P. Zhou, J. Zhu, S. Huo, and D. Navarro-Alarcon, “Lasesom: A latent and semantic representation framework for soft object manipulation,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5381–5388, 2021.

[24] Z. Lončarević, A. Gams, A. Ude *et al.*, “Robot skill learning in latent space of a deep autoencoder neural network,” *Robotics and Autonomous Systems*, vol. 135, p. 103690, 2021.

[25] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[26] G. Shamai, M. Zibulevsky, and R. Kimmel, “Efficient inter-geodesic distance computation and fast classical scaling,” *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[27] K. Crane, C. Weischedel, and M. Wardetzky, “Geodesics in heat: A new approach to computing distance based on heat flow,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 5, pp. 1–11, 2013.

[28] W.-K. Jeong and R. T. Whitaker, “A fast iterative method for Eikonal equations,” *SIAM Journal on Scientific Computing*, vol. 30, no. 5, pp. 2512–2534, 2008.

[29] J. Ernesti, L. Righetti, M. Do, T. Asfour, and S. Schaal, “Encoding of periodic and their transient motions by a single dynamic movement primitive,” in *2012 12th IEEE-RAS international conference on humanoid robots (humanoids 2012)*. IEEE, 2012, pp. 57–64.

[30] J. Yang, J. Zhang, C. Settle, A. Rai, R. Antonova, and J. Bohg, “Learning periodic tasks from human demonstrations,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8658–8665.

[31] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, “On-line learning and modulation of periodic movements with nonlinear dynamical systems,” *Autonomous robots*, vol. 27, pp. 3–23, 2009.

[32] A. Ude, A. Gams, T. Asfour, and J. Morimoto, “Task-specific generalization of discrete and periodic dynamic movement primitives,” *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[33] Y. Huang, F. J. Abu-Dakka, J. Silvério, and D. G. Caldwell, “Toward orientation learning and adaptation in cartesian space,” *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 82–98, 2020.

[34] X. Li, H. Cheng, H. Chen, and J. Chen, “Learning quasi-periodic robot motions from demonstration,” *Autonomous Robots*, vol. 44, pp. 251–266, 2020.

[35] G. BakIr, T. Hofmann, A. J. Smola, B. Schölkopf, and B. Taskar, *Predicting structured data*. MIT press, 2007.

[36] C. Ciliberto, L. Rosasco, and A. Rudi, “A general framework for consistent structured prediction with implicit loss embeddings.” *J. Mach. Learn. Res.*, vol. 21, no. 98, pp. 1–67, 2020.

[37] M. Alvarez, L. Rosasco, and N. Lawrence, “Kernels for vector-valued functions: A review,” *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[38] Z. Jin, A. Liu, W.-a. Zhang, L. Yu, and C. Yang, “Gaussian process movement primitive,” *Automatica*, vol. 155, p. 111120, 2023.

[39] G. S. Kimeldorf and G. Wahba, “A correspondence between Bayesian estimation on stochastic processes and smoothing by splines,” *The Annals of Mathematical Statistics*, vol. 41, no. 2, pp. 495–502, 1970.

[40] D. Duvenaud, “Automatic model construction with Gaussian processes,” Ph.D. dissertation, University of Cambridge, 2014.

[41] D. J. MacKay, “Introduction to Gaussian processes,” *NATO ASI series F: Computer and systems sciences*, vol. 168, pp. 133–166, 1998.

[42] S. Noorzadeh, M. Niknazar, B. Rivet, J. Fontecave-Jallon, P.-Y. Guméry, and C. Jutten, “Modeling quasi-periodic signals by a non-parametric model: Application on fetal ECG extraction,” in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 1889–1892.

[43] R. Angus, T. Morton, S. Aigrain, D. Foreman-Mackey, and V. Rajpaul, “Inferring probabilistic stellar rotation periods using Gaussian processes,” *Monthly Notices of the Royal Astronomical Society*, vol. 474, no. 2, pp. 2094–2108, 2018.

[44] P. Bose, A. Maheshwari, C. Shu, and S. Wuhrer, “A survey of geodesic paths on 3D surfaces,” *Computational Geometry*, vol. 44, no. 9, pp. 486–498, 2011.

[45] D. Li, M. Mukhopadhyay, and D. B. Dunson, “Efficient manifold approximation with spherelets,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 84, no. 4, pp. 1129–1149, 04 2022. [Online]. Available: https://doi.org/10.1111/rssb.12508

[46] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[47] G. Wahba, *Spline models for observational data*. Siam, 1990, vol. 59.

[48] J. Schulman, J. Ho, C. Lee, and P. Abbeel, “Learning from demonstrations through the use of non-rigid registration,” in *Robotics Research*. Springer, 2016, pp. 339–354.

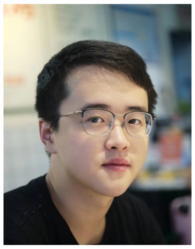
[49] H. Cai, Y. Guo, Z. Peng, and J. Zhang, “Landmark detection and 3D face reconstruction for caricature using a nonlinear parametric model,” *Graphical Models*, vol. 115, p. 101103, 2021.

[50] P. Zhou, P. Zheng, J. Qi, C. Li, A. Duan, M. Xu, V. Wu, and D. Navarro-Alarcon, “Neural reactive path planning with Riemannian motion policies for robotic silicone sealing,” *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102518, 2023.

[51] S. Ruan, W. Liu, X. Wang, X. Meng, and G. S. Chirikjian, “PRIMP: PRobabilistically-informed motion primitives for efficient affordance learning from demonstration,” *IEEE Transactions on Robotics*, vol. 40, pp. 2868–2887, 2024.

[52] L. Hu, D. Navarro-Alarcon, A. Cherubini, M. Li, and L. Li, “On radiation-based thermal servoing: New models, controls, and experiments,” *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1945–1958, 2022.

Anqing Duan received the BEng in mechanical engineering from Harbin Institute of Technology, China, in 2015, the Msc in mechatronics from KTH, Sweden, in 2017, and the Ph.D. degree in robotics from the Istituto Italiano di Tecnologia and the Università degli Studi di Genova, Italy, in 2021. He was a visiting researcher with the Learning Algorithms and Systems Laboratory at Swiss Federal Institute of Technology Lausanne, Switzerland, and a Research Associate with the Robotics and Machine Intelligence Laboratory at The Hong Kong Polytechnic University, Hong Kong. He is currently a Visiting Assistant Professor at Mohamed Bin Zayed University of Artificial Intelligence. His research interest lies in the intersection of machine learning and robotics.



Wanli Liuchen received the B.Eng. in Mechanical Engineering and Automation from the China University of Petroleum (Beijing) in 2022 and completed the M.Sc. in Mechanical Engineering at The Hong Kong Polytechnic University in 2024. He is currently a Research Assistant at the ROMI Lab, The Hong Kong Polytechnic University, where his work focuses on robotic systems, particularly in the field of medical robotics. His research interests include robot navigation, machine learning, and healthcare robotics applications.

Jinsong Wu received the BEng degree in mechatronics and robotic systems from the University of Liverpool, UK, in 2021, and the MRes degree in medical robotics and image-guided intervention from Imperial College London, UK, in 2022. From 2022 to 2023, he was a Research Assistant the The Chinese University of Hong Kong, Hong Kong. Since 2023, he has been pursuing a Ph.D. degree in mechanical engineering at The Hong Kong Polytechnic University, Hong Kong. His current research interests include medical robotics and control theory.

Raffaello Camoriano received the B.Sc. and M.Sc. degrees in computer and robotics engineering and the Ph.D. degree from the University of Genoa, Italy, in 2011, 2013, and 2017, respectively. He has been a Research Fellow and a Postdoctoral Researcher at the Istituto Italiano di Tecnologia, Genoa, Italy, from 2014 to 2022. He is currently an Assistant Professor with the Department of Control and Computer Engineering at Politecnico di Torino, Turin, Italy, and an Affiliated Researcher at Istituto Italiano di Tecnologia, Genoa, Italy. He is author of 20 international peer-reviewed articles in the fields of efficient learning from structured data, continual learning, reinforcement learning, and robotics. Prof. Camoriano received the IEEE Computational Intelligence Society Italy Section Chapter’s 2017 Best Ph.D. Thesis Award and is a Member of the European Laboratory for Learning and Intelligent Systems. He serves as Associate Editor for IEEE ROBOTICS AND AUTOMATION LETTERS, SPRINGER MACHINE LEARNING, and IEEE/RSJ IROS.

Lorenzo Rosasco received the Ph.D. degree in computer science from the University of Genova, Genoa, Italy, in 2006. He is currently a Full Professor with the University of Genova, Genoa, Italy. He is also affiliated with the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, where he is currently a Visiting Professor, and with the Istituto Italiano di Tecnologia (IIT), Genoa, where he is an External Collaborator. He is leading the efforts to establish the Laboratory for Computational and Statistical Learning, from a collaborative agreement between IIT and MIT. Between 2006 and 2009, he was a Postdoctoral Fellow with Center for Biological and Computational Learning, MIT, working with T. Poggio. His research interests include theory and algorithms for machine learning. He has developed and analyzed methods to learn from small as well as large samples of high-dimensional data, using analytical and probabilistic tools, within a multidisciplinary approach drawing concepts and techniques primarily from computer science but also from statistics, engineering, and applied mathematics.

David Navarro-Alarcon (Senior Member, IEEE) received the master’s degree in robotics from the Centre for Research and Advanced Studies, National Polytechnic Institute of Mexico, in 2009, and the Ph.D. degree in mechanical and automation engineering from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2014.

From 2015 to 2017, he was a Research Assistant Professor at the CUHK T Stone Robotics Institute, Hong Kong. Since 2017, he has been with The Hong Kong Polytechnic University (PolyU), Hong Kong, where he is currently an Associate Professor with the Department of Mechanical Engineering, and the Principal Investigator of the Robotics and Machine Intelligence Laboratory. His research interests include perceptual robotics and control theory.

Dr. Navarro-Alarcon currently serves as an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS and Associate Editor of the IEEE ROBOTICS AND AUTOMATION MAGAZINE.