Spring 2024

An Exploration of Dimensionality Reduction of Dynamics on Lie Groups via Structure-Aware Canonical Correlation Analysis

Wooyoung Chung
San Jose State University

AN EXPLORATION OF DIMENSIONALITY REDUCTION OF DYNAMICS ON LIE GROUPS VIA STRUCTURE-AWARE CANONICAL CORRELATION ANALYSIS

A Thesis

Presented to

The Faculty of the Department of Computer Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Wooyoung Chung

May 2024

The Designated Thesis Committee Approves the Thesis Titled

AN EXPLORATION OF DIMENSIONALITY REDUCTION OF DYNAMICS ON LIE GROUPS VIA STRUCTURE-AWARE CANONICAL CORRELATION ANALYSIS

by

Wooyoung Chung

APPROVED FOR THE DEPARTMENT OF COMPUTER ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2024

Stas Tiomkin, Ph.D.	Department of Computer Engineering
Carlos Rojas, Ph.D.	Department of Computer Engineering
Jun Liu, Ph.D.	Department of Computer Engineering

ABSTRACT

AN EXPLORATION OF DIMENSIONALITY REDUCTION OF DYNAMICS ON LIE GROUPS VIA STRUCTURE-AWARE CANONICAL CORRELATION ANALYSIS

by Wooyoung Chung

Incorporating prior knowledge into a data-driven modeling problem can drastically improve performance, reliability, and generalization outside of the training sample. The stronger the structural properties, the more effective these improvements become.

Manifolds are a powerful nonlinear generalization of Euclidean space for modeling finite dimensions. When additionally assuming that the manifold carries (Lie) group structure, this imposes a drastically stricter global constraint. The range of their applications is very wide and includes the important case of robotic tasks. We apply this idea to Canonical Correlation Analysis (CCA).

In traditional CCA one constructs a hierarchical sequence of maximal correlations of up to two paired data sets in Euclidean spaces. We here generalize the CCA concept to respect the structure of Lie groups and demonstrate its efficacy through the substantial improvements it achieves in making structure-consistent predictions about changes in the state of a robotic hand.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Stas Tiomkin for all the support, patience, and dedication to make this possible. From all the middle-of-the-night responses to the discussions on the whiteboard, I grew more than I could have ever imagined in the field of AI research.

I would also like to thank Dr. Liu and Dr. Rojas for their time by being part of this committee.

TABLE OF CONTENTS

List of Figures vii

1 Introduction 1

2 Overview 4

3 Preliminaries 5
3.1 Intrinsic vs. Extrinsic Means 5
3.1.1 Intrinsic mean 5
3.1.2 Extrinsic mean 6
3.2 Banana Distribution 7
3.3 Projection to Subgroups 8

4 Literature Review 11
4.1 Information Compression 11
4.1.1 Information Bottleneck 11
4.1.2 Principal Component Analysis 12
4.1.3 Canonical Correlation Analysis 14
4.2 Structured Learning 14
4.2.1 Manifold Learning 15
4.2.2 Structured Prediction 15

5 Methodology 17
5.1 Intrinsic Canonical Correlation Analysis 18
5.1.1 First ICCA pair 18
5.1.2 Next ICCA Pairs 20
5.2 ICCA Decomposition 21
5.3 ICCA Reconstruction 22

6 Experiments 23
6.1 Experimental settings 23
6.2 Results 24

7 Future Work 29

8 Conclusions 30

Literature Cited 31

Appendix A 35
A.0.1 Groups of rotation and translation 35

LIST OF FIGURES

Fig. 1. An experiment of comparing intrinsic and extrinsic Gaussian distribution was conducted on a two-dimensional car with the task of moving straight with a noise added to its steering. The red circles represent extrinsic Gaussian distribution while the blue circles represent intrinsic Gaussian distribution (concentrated Gaussian distribution). From this, we see that concentrated gaussian was successfully formed its distribution that is non-linear 8

Fig. 2. Concentrated Gaussian Distribution 9

Fig. 3. MSE between the true and the reconstructed configurations of the anthropomorphic robotic arm for CCA and ICCA in training and testing. 25

Fig. 4. **Optimal Projection Time**. t^* vs s^* comparison that uses the canonical pair to map to the original data. t^* and s^* show a linear relationship between one another. Thus a simple linear regression model would be suitable to map t^* to s^*. 26

Fig. 5. Anthropomorphic Robotic Hand used in Sec.6. The images above show one of the simulations along with its ICCA reconstruction. Top-left represents the original state. Bottom-left represents the initial configuration (the neutral state with noise). Top-right represents the ground-truth configuration. Bottom-right represents the ICCA reconstruction of the final configuration given the initial configuration. 27

Fig. 6. Convergence of the ICCA loss for Grasping Hand in the iteration stage. The loss plot is created by taking the average on 10 sets of experiments on the grasping hand task. The curve is based on loss in the iteration stage of ICCA Decomposition. There is an additional loss decrease in the initialization stage. 28

Fig. 7. Here we can see a 3D rotation represented through the $SO(3)$ group. The 3D rotation can be represented through the surface of a sphere. The group movement contains two components, the arbitrary unit vector of the 3D movement t and the scale of the movement t. 36

1 INTRODUCTION

Simplicity that respects structure is a desirable property of effective control. Typically, it improves a method's robustness, feasibility, and flexibility and is achieved by reducing its complexity. A common form of complexity reduction is realized by dimensionality reduction, which represents a special case of the more general principle of information compression methods.

One popular information compression model is the Information Bottleneck (IB) [1], [2], a principled method to achieve such reductions with intimate links to statistical machine learning. Of particular interest to the control community is the fact that the IB is a direct generalization of the well-established CCA [3]. In the case of (locally) linear Gaussian models, CCA permits tuning the degree of structural preservation from one variable to another. The IB thus implements a "soft" CCA in the Gaussian case [4], [5].

However, CCA and its informational generalization (IB) purely concentrate on preserving the dependency of the target variables. They are utterly indifferent to any particular additional structure of the problem, some paradigmatic consequences of which we now illustrate in a pertinent example.

In [5], an IB is applied to a linear Gaussian control channel which thus reduces to a soft CCA model. Reducing the information that a simpler model has access to, the process leads to a progressive reduction of the dimensionality in the accompanying soft CCA. However, this reduction is purely correlational and does not consider the special structure of the control loop, as discussed in [6]. Concretely, in the controlled system, the matrix transforms the process' past dynamics into the process' future dynamics. The original transformation has a particular recursive structure following that of the Hankel matrix [3]. After the system is naively subject to information/dimensionality reduction, the resulting reduced transformation matrix for the compressed system no longer has a Hankel structure [5].

The structural deficit is alleviated in [6] by modifying the approach of [5] to constrain the reduced transformation matrices to satisfy the properties of a proper Hankel matrix to represent actual control systems. However, this approach is unsuitable for general use due to the requirement of handcrafting the information reduction method to respect the Hankel matrix structure of the control problem.

The present work presents a method to support a generalizable approach to produce structure-respecting IB in the future. We modify the traditional CCA method to respect the structure of a manifold to which the variables of interest and their interrelation are confined. In other words, we constrain our problem space to manifolds instead of the Euclidean space.

To achieve this we note that the naive concept of various averaging operations used to compute the CCA in the Euclidean case needs to be modified, as manifolds do not offer mean and variance computation via vector addition operations. Therefore, instead, we resort to the variational description of the quantities of interest. The idea is analog to the fact that, in Euclidean space, the centroid of given data points can not only be computed by directly taking the average vector of the data points but, alternatively, by finding that point that minimizes the sum of its squared Euclidean distances to the given data.

The key components of the proposed method will be: 1. this variational principle, now with manifold-intrinsic distance, to replace the Euclidean averaging operation to compute the generalized mean, and 2. using projections to sub-manifolds replacing those to Euclidean vector subspaces.

The paper is organized as follows. Section 2 begins with a general overview of well-known concepts and the novelty of our approach. The overview is followed by the preliminaries relevant to the development of the formalism in Section 3 (note that Lie Theory and manifolds as relevant to the paper are covered in the Appendix). We highlight how to replace the computation of the mean and centroid with a variation of expected distance to transfer the concept from Euclidean space onto Lie groups. This step is crucial for the development of the

Intrinsic CCA. A Literature Review is presented in Section 4 where we go through important related works that have aided in development of ICCA. Section 5 presents the proposed method for ICCA, extending the concept of principal geodesic curves to canonical geodesic curve pairs, denoted by *intrinsic CCA*, and represents an efficient algorithm for the calculation of ICCA from data points on high-dimensional Lie groups. In Section 6, we demonstrate this algorithm on a high-dimensional articulated robotic system, an anthropomorphic robotic hand, whose configuration space is given, in general, by the corresponding multi-dimensional Lie group. Section 7 represents potential future directions. Finally, we reanalyzes the paper to summarize our work and understand the impact of the findings.

This work has been accepted to the American Control Conference 2024 and this book has been adapted from the paper ”Dimensionality reduction of dynamics on lie groups via structure-aware canonical correlation analysis” [7].

2 OVERVIEW

Existing methods for dimensionality reduction consider their source data living in ”flat” Euclidean spaces and are utterly agnostic to any potential additional structure or constraints [3], [4], [8]–[10]. Specifically, there are no methods for *joint* dimensionality reduction (compression) of two sets of points when these are restricted to a Lie group.

Joint compression could reveal predictive models between one set to another, with additionally choosing a desired level of complexity and details. In particular, such structure-preserving predictive models would permit the reconstruction of a point in one set on the Lie group from a point in another set on the same Lie group. A particular scenario of interest is the estimation of dynamical models on the manifold, where two sets of points on the manifold represent the current state and the state in T time steps to the future. This estimation is much more difficult or impossible if a model does not preserve the manifold structure. Concretely, here, we propose a method to generalize the CCA to the ICCA on general Lie groups.

Applying the Riemannian approach to CCA allows us to control the model's intrinsic complexity by choosing how many intrinsic geodesic pairs are being used (Section 5) while accounting for the additional intrinsic structure and.

Our main contributions include:

1) Two algorithms for ICCA decomposition and reconstruction;
2) Reduction in the prediction error of the future state of robotic hand from the current state compared to the existing baseline dimensionality reduction methods;
3) Structural guarantees that the predicted state is confined to the Lie group.

As one remarkable result, we found that the relationship between the ”intrinsic” times of the basic geodesic movements can be mapped linearly to each other (Section 6).

3 PRELIMINARIES

This work uses the standard definitions of the Lie group (including the groups for rotation and translation) frequently used in robotics. This background is provided in the Appendix for completeness.

The next section gives an overview over the key components of the proposed method (intrinsic distance, averaging operations, and sub-manifold projection).

3.1 Intrinsic vs. Extrinsic Means

Intrinsic and extrinsic calculations differ fundamentally from one another. Extrinsic (Euclidean) calculation limits itself to the linear constraint and does not account for the non-linear structure embedded in many datasets. To develop a structure-aware ICCA, we will need to calculate the intrinsic mean and projections of data points on the Lie group to its sub-manifolds (Section 3.3). We now introduce the definitions on which the intrinsic analogues of Euclidean concepts will be based on.

Given a set of data points $\{x_i\}_{i=1}^N$ in a metric space, $\mathscr{X}$, their mean, μ_x, is defined by:

$$\mu_x = \text{argmin}_{x\in\mathscr{X}} \sum_{i=1}^{N} D^2(x, x_i), \tag{1}$$

where $D(\cdot,\cdot)$ denotes the distance between points in $\mathscr{X}$ and is assumed the minimum to be unique. I.e., the mean of a set of points in a general metric space is defined by the solution to the optimization problem in Eq. (1).

This variational formulation offers a generalization of the mean beyond spaces in which arithmetic means can be computed, i.e. which permit convex combinations or explicit addition operators, such as the standard Euclidean space. In the latter, Eq. (1) has as closed-form solution the traditional arithmetic average of the data points, $\mu_x = \frac{1}{N}\sum_i x_i$.

3.1.1 Intrinsic mean

We now apply this variational method to compute the mean intrinsically to a Lie group. Given a Lie group, $\mathscr{X}$, the distance between data points, $x_1, x_2 \in \mathscr{X}$ is given by the

Riemannian distance on the manifold:

$$D^2(x_1,x_2) \triangleq ||\log(x_1^{-1}x_2)||_2^2, \tag{2}$$

where the inverse is a Lie group inverse and 'log' denotes the logarithmic map[1]

. With this, we generalize the calculation of the intrinsic mean [11], [12] of $x_1,x_2,\ldots,x_N \in \mathscr{X}$ by solving Eq. (1). The problem in Eq. (1) can be solved iteratively [11], [12] or can be approximated for small x_1 and x_2 by the Baker–Campbell–Hausdorff formula [13], [14] given by Eq. (3):

$$||\log(x_1^{-1}x_2)||_2 \approx ||\log(x_2) - \log(x_1)||_2, \tag{3}$$

which omits the non-commutative terms between x_1^{-1} and x_2.

3.1.2 Extrinsic mean

If we would instead embed data points from the Lie manifold $\mathscr{X}$ into the ambient Euclidean space [15] we can calculate the mean in Eq. (1) directly using the Euclidean distance of the ambient space via the arithmetic computation of the mean, leading to the traditional *extrinsic mean*. However, in general, this mean will not be a point on a manifold. This creates a discrepancy between the internal structure of the data and its *extrinsic* statistics. This discrepancy results in imprecise modeling of data if these are actually constrained to a manifold and, hence, in deficient generalization across samples. By violating the constraints represented by the manifold structure, the extrinsic mean may not even represent a physically realizable configuration of the system at all.

1. Strictly spoken, the logarithmic map is defined locally. Without loss of generality, we resolve ambiguities by choosing the pre-image of its argument with respect to the exponential map with the minimal norm and breaking remaining ambiguities as per convenience of the respective computation.

3.2 Banana Distribution

Similar to the mean, the Gaussian distribution differs fundamentally from its intrinsic and extrinsic forms [16].

In extrinsic Gaussian distribution, the variation of the data is considered linearly, where the most likely position (calculated by the extrinsic mean) lies in its center. This creates an oval area of distribution that is utterly blind in accounting movement Fig. 1.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{4}$$

Contrarily, intrinsic Gaussian distribution, also known as concentrated Gaussian distribution [17], creates a distribution that accounts for the structure of the data. This can be done by mapping the standard multivariate Gaussian distribution into the group space ε and then adding it to the intrinsic movement μ seen in Fig. 2.

$$X = \mu exp_G(\varepsilon) \tag{5}$$

Let's consider the scenario of a two-dimensional car that moves straight with a little noise to the steering (seen in Fig. 1). We should expect the distribution to be in the shape of a banana as the variance in the rotation of the steering should not change the distance traveled from the initial point. This is what we get for concentrated Gaussian distribution labeled exp pdf in Fig. 1. And of course, the standard Gaussian distribution (labed XY pdf) was unable to correctly represent this movement as it just creates a little oval around its expected point.

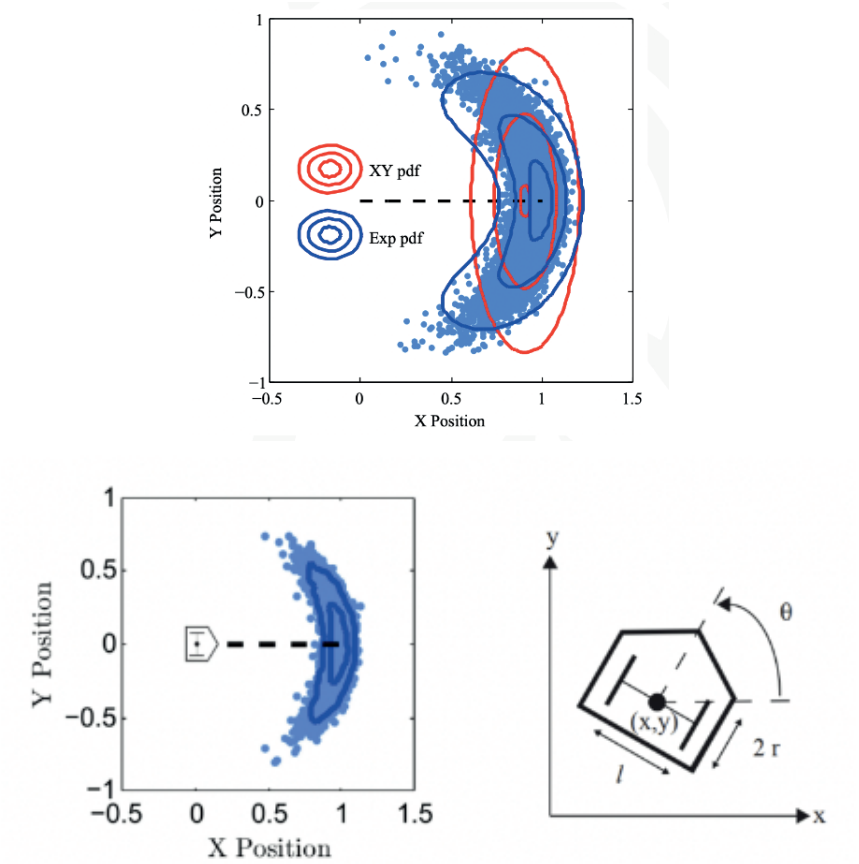

Fig. 1. An experiment of comparing intrinsic and extrinsic Gaussian distribution was conducted on a two-dimensional car with the task of moving straight with a noise added to its steering. The red circles represent extrinsic Gaussian distribution while the blue circles represent intrinsic Gaussian distribution (concentrated Gaussian distribution). From this, we see that concentrated gaussian was successfully formed its distribution that is non-linear

3.3 Projection to Subgroups

Let G and $\mathfrak{g}$ be the Lie group and its corresponding algebra. For an arbitrary unit vector $v \in \mathfrak{g}$, we can define a one-parameter subgroup H_v of G [11]:

$$H_v \triangleq \{\exp(tv) \in G \,:\, t \in \mathbb{R}\}, \tag{6}$$

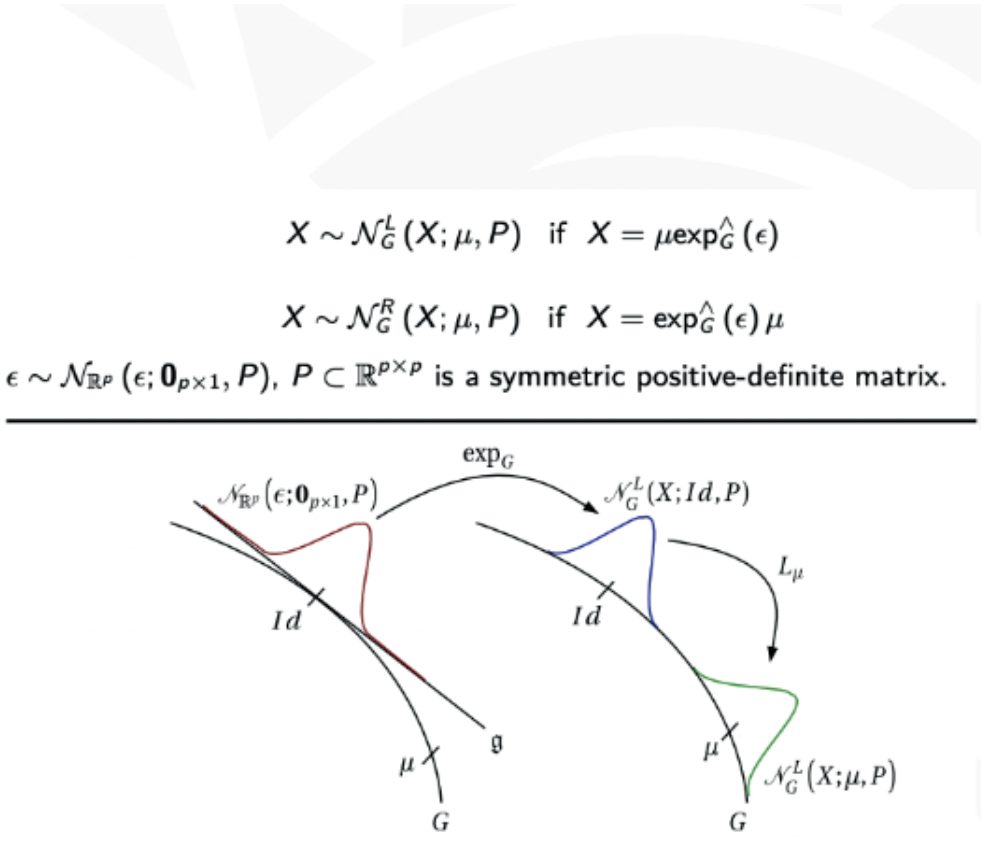

Fig. 2. Concentrated Gaussian Distribution

where 'exp' is the exponential map from $\mathfrak{g}$ to G, given in Eq. (31). The distance between any $x \in G$ and H_v is given by:

$$D(x,H_v) \triangleq \min_t D(x,\exp(tv)), \tag{7}$$

with the optimal value of t being given by:

$$t^* = \text{argmin}_t\, D(x,\exp(tv)), \tag{8}$$

determining the *projection* of x onto H_v:

$$\mathrm{Proj}_{H_v}(x) \triangleq \exp(t^* v). \tag{9}$$

This projection of a group element to a one-parameter subgroup is a core component in the Intrinsic PCA [11], [14] (explained in the next section), which we generalize to Intrinsic CCA in this work.

4 LITERATURE REVIEW

4.1 Information Compression

Information compression is the task of reducing the size of the data while trying to maintain as much information as possible. The foundation of information compression is based on *Shannon's Source Coding Theorem* [18] that states: A source with **entropy** rate H can be encoded with an arbitrarily small error probability at any rate $R > H$. Conversely, if $R < H$, the error probability will be bounded away from zero, independent of the encoder and decoder employed.

Machine learning has proven a promising direction for information compression. The effective data generalization task of machine learning mirrors the minimization of storage of data compression. Recent works include DeepMind's Chinchilla LLM [19] that used deep learning to efficiently compress images with performance being better than many recognizable image compression methods such as Portable Network Graphics (PNG).

Existing methods for dimensionality reduction consider their source data living in "flat" Euclidean spaces and are utterly agnostic to any potential additional structure or constraints [3], [4], [8]–[10]. Some common methodologies, such as Information Bottleneck (Section 4.1.1), Principal Component Analysis (Section 4.1.2), and Canonical Correlation Analysis (Section 4.1.3) are such methods.

4.1.1 Information Bottleneck

With the success of deep neural networks (DNN) and deep learning (DL) in many applications of supervised learning, information bottleneck (IB) was developed with the formulation of using deep learning as a trade-off between compression and prediction [1].

The information bottleneck (IB) method was developed for extracting relevant information from random variable X about random variable Y through the use of mutual information I(X;Y) from its joint distribution p(X,Y) [1]. Its basis lies in encoding the information of the

target random variable Y into a latent representation Z from the input target variable X, I.E. minimizing I(X;Z) and maximizing I(Z;Y) through the minimization of the Lagrangian:

$$\mathscr{L}[p(Z|X)] = I(X;Z) - \beta I(Z;Y) \tag{10}$$

The information bottleneck principle is further extended through variational approximation that utilizes the reparameterization trick [20] to present variational approximation to the IB [21]. From this, variational IB performs better generalization and sample efficiency by finding a better structure of the latent representation Z.

The IB and its extended variational IB method have been used in word cluster analysis [22], image sparse latent representation [23], and robotic prior exploration [24].

4.1.2 Principal Component Analysis

Principal component analysis (PCA) is one of the most popular dimensional reduction methods for data exploration, visualization, and preprocessing. The linear dimension reduction method compresses data into orthogonal linear combinations called principal components (PC) that capture the variance [25].

Traditional PCA is concerned with dimensionality reduction through the projection of data to linear subspaces, minimizing the reconstruction error. As one increases the dimensionality of the subspaces, new independent (orthogonal) additional features are accounted for.

Whether in the Euclidean or the manifold case, the calculation is based on the above principle of hierarchical projection of data on those subspaces or -manifolds which minimize the mean projection error between the data points and the corresponding subspace [14]. The difference between PCA in the Euclidean space and PCA on the Lie group is in the definition of the subspace and that of the distance, and the operators used for averaging.

PCA in Euclidean Space One calculates the hierarchical projections beginning with the first PCA component, $k = 1$, which is a one-dimensional linear (strictly spoken, affine) space.

Given the Euclidean space, X, we compute it by seeking a one-dimensional subspace onto which the data points $x_1, x_2, \ldots, x_N \in X$ project with the least total distance loss [14], more precisely, we seek $S_v = \{tv \,:\, t \in \mathbb{R}\}$ such that:

$$v^{(1)} = \text{argmin}_{||v||=1} \sum_{i=1}^{N} ||x_i - \text{Proj}_{S_v}(x_i)||_2^2, \tag{11}$$

where $\text{Proj}_{S_v}(x) = (x \cdot v)v$ is the optimal projection of x on S_v. We compute the subsequent PCA components recursively by proceeding to increasingly higher-dimensional subspaces, by removing the contribution of the already established subspaces and minimizing the distance loss of the data points with respect to the newly added one. Concretely, for $k > 1$, we calculate recursively [14]:

$$v^{(k>1)} = \text{argmin}_{||v||=1} \sum_{i=1}^{N} ||x_i - \text{Proj}_{S_v}(x_i) - \sum_{\ell=1}^{k-1} \text{Proj}_{S_{v^{(\ell)}}}(x_i)||_2^2. \tag{12}$$

In other words, the first projection minimizes the deviations to the first component, and all subsequent projections minimize the residual deviation to the new component after all previous components have been accounted for.

There are two essential differences between PCA in the Euclidean space and on the Lie group. First, the Euclidean distance function is inappropriate for estimating the projection error on the Lie group. Instead, the manifold-intrinsic distance, such as the Riemannian distance, should be used [26]. Second, the sub-spaces/principal components in the Euclidean space are given by vector (strictly spoken, affine, if the mean does not coincide with the origin) subspaces, while in the Lie group, they are given by *principal geodesic curves*.

PCA on Lie Groups The generalization of the first principal vector in Eq. (11) to the first principal geodesic curve is achieved by combining Eq. (1),Eq. (7), and Eqs. (11) and (12). One obtains the first principal geodesic curve, [14]:

$$v^{(1)} = \text{argmin}_{||v||=1} \sum_{i=1}^{N} \min_t || \log\big((\mu^{-1}x_i)^{-1} \exp(tv)\big)||^2, \tag{13}$$

where μ, $'\log'$, and $'\exp'$, are the mean, given in Eq. (1), the logarithmic and exponential maps (cf., Appendix), accordingly.

The principal geodesic curves for $k > 1$ are defined analogously to $v^{(k>1)}$ in Eq. (12), again , with the appropriate distance (Eq. (7)), projection (Eq. (9)), and mean (Eq. (1)), respectively.

4.1.3 Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a fundamental tool to estimate two subspaces and a linear transformation between them from data, such that each of the original data sets are represented by (projected to) the corresponding subspaces with minimal error; similarly, there is a minimal error between the projected data to these subspaces.

Its applications extend far beyond pure data analysis; they include computer vision [27], speech synthesis [28], and robotic control [29]. Notably, and of particular interest for control, CCA and its information-theoretic generalization has been proven useful for the dimensionality reduction of linear dynamical systems [3]–[6]. With its wide variety of applications, however, conventional CCA tends to handle non-linear data poorly due to its Euclidean assumptions [30]. One can therefore expect that adapting CCA to the intrinsic structure of the data it is to represent should improve the ability of this dimension reduction technique to respect and faithfully preserve the mapping between the set of data pairs.

4.2 Structured Learning

Machine learning has experienced significant growth in both use and relevance in society. It is used in many applications such as self-driving cars and object manipulation. With bigger and grander tasks, the complexity and time cost have grown tremendously. To learn these data, we must learn the structure and its pattern either implicitly or explicitly. In conventional (unstructured) learning methods, the structure is learned implicitly.

Unstructured methods treat data linearly, where all points in the data can be placed in a global Euclidean space. Although Euclidean representation can be generalized to nearly all data-oriented tasks, it comes at the price of representing data in a high-dimensional space

where not all points are possible data configurations. Especially when trying to learn a highly complex task, it is essential to have sample efficiency and interpretability

On the contrary, structured learning explicitly defines the structure of the data. This finds a balance of expressive power and querying efficiency by incorporating prior knowledge [31]. By taking advantage of prior knowledge that is often easily obtainable from data, the task of learning provides structural guarantees that follow the implicit rules of the data. This leads to exponentially decreased learning complexity by limiting the learning space to its structured constraints.

4.2.1 Manifold Learning

Manifold learning is a non-linear dimension reduction method where data is represented through lower-dimensional latent manifolds. Manifold learning is based on the Manifold Hypothesis, which states that high-dimensional data often lies within low-dimensional manifolds [32].

Manifold learning provides mapping that facilitates a smooth transition between two data points, aligning with the inherent structure of the dataset. This provides a significantly improved compression of data with minimal loss of information. The unsupervised technique has found application across various tasks, including its utilization within the discriminator of GANs to augment generative learning [33]. Additionally, manifold learning has gained popular use in improving cluster analysis for better visualization and analysis [34].

4.2.2 Structured Prediction

Structured prediction employs a supervised methodology within structured learning, aiming to narrow down the predictive range of the output. When implemented with neural networks, this is typically achieved through two primary methods: either by structuring the output directly [35] or by integrating structural constraints within the hidden layers [36].

NLP has been popular in incorporating structured prediction by incorporating structures such as Part-Of-Speech Tagging [35], non-linear feature parsing [37], and sequence labeling [38].

5 METHODOLOGY

In this work, we propose a particularly useful specialization of the nonlinear case, namely a CCA that "lives" specifically on Lie groups. Such an operation is of particular interest in the context of robotics [39]. Environments that contain complex manipulation, shape modifications, or other structural changes can be suitably described in the language of Lie group operations.

Much like in PCA, CCA implements a hierarchy of projections between data and sub-spaces. However, CCA uses two projections per component, called the *canonical pairs*. In the Euclidean space these pairs are derived using the standard distance, averaging and projection operators.

In order to define the intrinsic CCA, we thus translate its functionality, given in the first paragraph of this section, into the language of intrinsic distance, averaging and projection operators on the Lie group, as follows.

Projection on the Subgroups We define one-parameter subgroups H_v and H_u of G, a distance between $x, y \in G$ and H_v, H_u, respectively, where $v, u \in \mathfrak{g}$:

$$H_v \triangleq \left\{\exp(tv) \in G \,:\, t \in R\right\}, \tag{14}$$

$$H_u \triangleq \left\{\exp(su) \in G \,:\, s \in R\right\}, \tag{15}$$

$$D(x, H_v) \triangleq \min_t D(x, \exp(tv)), \tag{16}$$

$$D(y, H_u) \triangleq \min_s D(y, \exp(su)), \tag{17}$$

By using the base movement of (v, u), we can represent each data point pair through the *optimal projection times* (t^*,s^*):

$$t^* = \text{argmin}_t \, D(x, \exp(tv)), \tag{18}$$

$$s^* = \text{argmin}_s \, D(y, \exp(su)). \tag{19}$$

The distance, $D(\cdot,\cdot)$, in Eq. (16) and Eq. (17) is the intrinsic Riemannian distance Eq. (2).

5.1 Intrinsic Canonical Correlation Analysis

Our methodology relies on optimizing projections to minimize the distance between the two subgroups of the set of points. We compress the original data into the pair (v,u) with their corresponding projection times (t^*,s^*). (v,u) and (t^*,s^*) are selected to minimize the prediction error between the past and future trajectory on the manifold, corresponding to the first vs. second entry of the pair, respectively, as explained below.

5.1.1 First ICCA pair

Given N point pairs, $\{x_i \in G, y_i \in G\}_{i=1}^N$, on the Lie manifold, G, we defined the first canonical geodesic pair as a pair of vectors in the corresponding Lie algebra, $\mathfrak{g}$, $\big(v^{(1)} \in \mathfrak{g}, u^{(1)} \in \mathfrak{g}\big)$, representing two one-parameter subgroups $\big(H_v, H_u\big)$, with which the data are maximally associated through their projections $\text{Proj}_{H_v}(x_i) \in G, \text{Proj}_{H_u}(y_i) \in G, i = 1 \ldots N$ with corresponding parametrizations t_i^*, s_i^*.

We propose to calculate the first ICCA pair from N point pairs $\{x_i, y_i\}_{i=1}^N$ on the manifold by:

$$\begin{aligned} v^{(1)}, u^{(1)} = \underset{||v||=||u||=1}{\text{argmin}} \sum_{i=1}^{N} \Big(& D^2(\mu_x^{-1} x_i, H_v) \\ & + D^2(\mu_y^{-1} y_i, H_u) \\ & + D^2\big(\text{Proj}_{H_v}(\mu_x^{-1} x_i), \text{Proj}_{H_u}(\mu_y^{-1} y_i)\big)\Big). \end{aligned} \tag{20}$$

or, explicitly,

$$\begin{aligned} v^{(1)}, u^{(1)} = \underset{||v||=||u||=1}{\text{argmin}} \sum_{i=1}^{N} \Big(& \min_t D^2(\mu_x^{-1} x_i, \exp(tv)) \\ & + \min_s D^2\big(\mu_y^{-1} y_i, \exp(su)\big) \\ & + D^2\big(\exp(t_i^* v), \exp(s_i^* u)\big)\Big), \end{aligned} \tag{21}$$

where the first two terms define for each i the time pair (t_i^*, s_i^*) via Eqs. (18) and (19), while the third term is the distance between the projected x_i and y_i, with the corresponding t_i^* and s_i^*, from each other. The joint projections given by the first two terms and the last term in Eq. (21) extend the intrinsic PCA towards the ICCA.

Optimal Projection Time. The solution to Eq. (20) includes $\left(v^{(1)}, u^{(1)}\right)$ and $\left(t_i^*, s_i^*\right)_{i=1}^N$. Given this solution we train a linear regression model, $f_\psi^{(1)}$, parameterized by ψ, to predict s from t:

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left(s_i^* - f_\psi^{(1)}(t_i^*)\right)^2 \tag{22}$$

which we concisely denote by $\hat{s}(t)$:

$$\hat{s}(t) = f_{\psi^*}^{(1)}(t). \tag{23}$$

The model in Eq. (23) allows us to reconstruct $\hat{y}$ from x using the first ICCA pair by:

$$t^* = \operatorname{argmin}_t D(x, \exp(t v^{(1)})), \tag{24}$$

$$\hat{y} = \exp\left(\hat{s}(t^*) u^{(1)}\right), \tag{25}$$

where $D(\cdot,\cdot)$ and 'exp' are the (intrinsic) Riemannian distance Eq. (2) and the exponential map Eq. (31), respectively.

Given that the two sets of data are assumed to be associated, the (t^*, s^*) derived from these data sets is expected to preserve some level of association with one another. A fortiori, we empirically found in our experiments (cf., Section 6) that the dependency between the optimal (t_i^*, s_i^*) is linear, and even close to the identity, as shown at Figure 4. We have not yet established a stringent theoretical justification for this phenomenon, whether this is a general property of the algorithm, a consequence of the construction of t and s from normalized vectors, stemming from constraining ourselves to the Lie group property, or a peculiarity of the particular experimental scenario.

We proceed by recursively defining the next ICCA pairs.

5.1.2 Next ICCA Pairs

Denote the projection of $\mu_x^{-1}x_i$ on $H_{v^{(1)}}$ by $\text{Proj}^{(1)}(x_i)$, and the projection of $\mu_y^{-1}y_i$ on $H_{u^{(1)}}$ by $\text{Proj}^{(1)}(y_i)$.

Firstly, we remove $\text{Proj}^{(1)}(x_i)$ from $\mu_x^{-1}x_i$ and $\text{Proj}^{(1)}(y_i)$ from $\mu_y^{-1}y_i$, which results in $x_i^{(2)}$ and $y_i^{(2)}$, respectively. Then, given $\forall i : (x_i^{(1)}, y_i^{(1)}) = (\exp(t_i^* v^{(1)}), \exp(s_i^* u^{(1)}))$, we define the second ICCA pair by:

$$\begin{aligned} v^{(2)}, u^{(2)} = \underset{||v||=||u||=1}{\text{argmin}} \sum_{i=1}^{N} \Big(& D^2(x_i^{(2)}, H_v) \\ & + D^2(y_i^{(2)}, H_u) \\ & + D^2\big(\text{Proj}_{H_v}(x_i^{(2)}), \text{Proj}_{H_u}(y_i^{(2)})\big)\Big). \end{aligned} \tag{26}$$

In general, the $(k+1)$-th ICCA pair is recursively defined by:

$$\begin{aligned} v^{(k+1)}, u^{(k+1)} = \underset{||v||=||u||=1}{\text{argmin}} \sum_{i=1}^{N} \Big(& D^2(x_i^{(k)}, H_v) \\ & + D^2(y_i^{(k)}, H_u) \\ & + D^2\big(\text{Proj}_{H_v}(x_i^{(k)}), \text{Proj}_{H_u}(y_i^{(k)})\big)\Big). \end{aligned} \tag{27}$$

where $x^{(k)}$ and $y^{(k)}$ are the residual data pair from the last iteration. The solution to the ICCA problem consists of all k ICCA pairs $\left(v^{(k)}, u^{(k)}\right)$ and their corresponding mappings $f^{(k)}(t)$ between the optimal projection times.

The equations Eq. (26) and Eq. (27) represent the decomposition of data into the k canonical pairs. We present two algorithms to decompose and reconstruct two data sets on the Lie group using the first canonical pair Eq. (21), which can be extended to the k-th canonical pair by Eq. (26) and Eq. (27). We show in Section 6 that already the first ICCA pair results in a significantly lower prediction error in comparison to the standard Euclidean CCA.

Algorithm 1 ICCA Decomposition

1: **Input:** $\{x_i, y_i\}_{i=1}^N \in G$ - data pairs on the manifold, G.
2: **Initialize**
3: $v, t^* \leftarrow \underset{||v||=1}{\textbf{argmin}} \sum_{i=1}^N \min_t ||\log\big(\mu_x^{-1} x_i \exp(tv)\big)||^2$,
4: $u, s^* \leftarrow \underset{||u||=1}{\textbf{argmin}} \sum_{i=1}^N \min_s ||\log\big(\mu_y^{-1} y_i \exp(su)\big)||^2$,
5: **repeat**
6: $\quad v, u \leftarrow$ **Eq.**(21) ▷ with fixed $\{t_i^*, s_i^*\}$
7: $\quad \{t_i^*\}_{i=1}^N \leftarrow \big\{\underset{t}{\textbf{argmin}}\, D(\mu_x^{-1} x_i, \exp(sv))\big\}_{i=1}^N$
8: $\quad \{s_i^*\}_{i=1}^N \leftarrow \big\{\underset{s}{\textbf{argmin}}\, D(\mu_y^{-1} y_i, \exp(tu))\big\}_{i=1}^N$
9: **until convergence**
10: $\hat{s}(\cdot) \leftarrow$ **Eq.**(23)
11: **return** $\{v^{(1)}, u^{(1)}\}$ **and** $\hat{s}(\cdot)$**.**

1) ICCA Decomposition: extracts the first canonical pair, and calculates the model, $\hat{s}$, mapping one optimal projection time, t^*, to the other, s^*. The subsequent pairs can be calculated by applying Eq. (27).
2) ICCA Reconstruction: Predicts $y \in G$ from $x \in G$ using the first canonical pair.

5.2 ICCA Decomposition

The ICCA decomposition uses iteration to solve for the minimization for the first canonical pair. The algorithm contains two stages: initialization and iteration.

The distance, $D(\cdot,\cdot)$ between a point on the manifold and a one-dimensional sub-manifold has multiple local minima. We initialize the algorithm by firstly finding the optimal projection of the data set to the first canonical pair $(v^{(1)}, u^{(1)})$, lines 2-4 in Alg. 1. Then, we optimize the full objective in Eq. (21) until convergence, lines 5-9 in Alg.1. Note that, for this, the iteration stage alternates between finding the optimal canonical pair $(v^{(1)}, u^{(1)})$ and its respective optimal projection times for the data, $\{t_i^*, s_i^*\}_{i=1}^N$.

At convergence, we estimate the predictor for s^* from t^* using linear regression, Eq. (23). The algorithm returns the model $\hat{s}$, and the optimal first canonical pair $(v^{(1)}, u^{(1)})$.

5.3 ICCA Reconstruction

Given the k ICCA pairs $(v^{(k)}, u^{(k)})$ and the model for the optimal prediction time, $\hat{s}(t)$, we can use these to reconstruct a secondary point $\hat{y}(x) \in G$ on the Lie group from a primary point $x \in G$ on the Lie group, as follows:

$$
\begin{aligned}
& t^{(1)} = \text{argmin}_t \, D(x, \exp(tv^{(1)})) \\
& \forall k \geq 1 \, : \, t^{(k+1)} = \text{argmin}_t \, D(x^{(k)}, \exp(tv^{(k)})) \\
& \hat{y} = \exp\Big(\sum_{k=1}^{K} \hat{s}^{(k)}(t^{(k)}) u^{(k)} \Big) \in G,
\end{aligned} \tag{28}
$$

where $x^{(k)}$ is as explained after Eq. (27) in Section 5.1.2.

6 EXPERIMENTS

The state space in articulated robotic systems [39] lies within a manifold consisting of group elements such as rotations and translations. The novelty of ICCA is the ability to account for these intrinsic properties of the robotic states. A more consistent representation of such states than provided by standard CCA would be one that remains inside the manifold structure, leading to improved CCA output accuracy. The goal of the experiments is to answer the following questions:

1) Can ICCA be effectively calculated from a finite sample of point pairs on the Lie group?
2) Is the ICCA reconstruction more accurate than that achieved by the standard Euclidean CCA?

6.1 Experimental settings

The MuJoCo simulator [40] was used to generate the data for the evaluation of the ICCA method on the anthropomorphic hand. The hand consisted of 14 finger joints; the inner finger joint was represented as a 3D Special Orthogonal SO(3) group, and the other finger joints as a 2D Special Orthogonal SO(2) group.

The two data sets for the ICCA analysis consist of the original configurations of the hand and the final configurations after 20 simulation steps with a stochastic action sequence 'Action Noise', as explained below. An example of the original (current) configuration, $x \in G$, is shown in the top left image at Figure 5, and an example for the final configuration, $y \in G$, is shown at the top right image.

The set of the current configurations, $\{x_i\}_{i=1}^N \in G$, is generated by adding a stochastic perturbation (noise) to $x \in G$, as explained below. Each of the final configurations, $\{y_i\}_{i=1}^N \in G$, corresponds to a particular $\{x_i\} \in G$ after applying the predefined action sequence with small perturbation in each action. An example of a perturbed current configuration and the corresponding final configuration of the hand are shown at the bottom

left and right images of Figure 5, respectively. To create the original configuration set $\{x_i\}_{i=1}^{N} \in G$, we applied noise with the following features:

1) **Configuration Noise:** Gaussian noise with zero mean vector with dimensionality 14 and diagonal covariance matrix, $\Sigma_{14} = 0.02\,\mathbf{I}_{14\times 14}$. This noise is added to the original configuration.
2) **Action Noise:** Gaussian noise with zero mean, $\mu = 0.0$ and variance, $\sigma = 0.01$. This noise is independently added to each action in the predefined action sequence with length 20.

The total number of data points, $\{x_i, y_i\}_{i=1}^{N}$, is $N = 10 \times 1500$, split into 10 experiments with 1500 data points in each experiment. $2:1$ 'train-to-test' split was performed on the data. We used the training data set for the calculations of the first ICCA pair, $(u^{(1)}, v^{(1)})$, and of the regression model $\hat{s}(t)$. The test set is used for measuring the reconstruction error between the true final configuration, y, in (x, y), and the reconstructed configuration, $\hat{y}(x)$, with $(u^{(1)}, v^{(1)})$ and $\hat{s}(t)$, as explained in Section 5.3. The reconstruction error was defined as the Mean Squared Error (MSE) between $\hat{y} = \hat{y}(x)$ and y. We compared the reconstruction error between the conventional CCA (based on the Euclidean distance) and the proposed ICCA (based on intrinsic distance).

6.2 Results

Accuracy and generalization are improved significantly for ICCA compared to CCA. The MSE comparison shown at Figure 3. The training improvement achieved 16.41%, while the testing improvement was 23.08%. ICCA achieved better generalization as the train-test accuracy difference was 3.77% for ICCA and 16.05% for CCA. The significant improvement can be attributed to the successful mapping of the non-linear structure of the robotic configuration data, where a locally linear relationship is found (Seen in Figure 4). his outcome highlights the significance of explicitly acknowledging the non-linear structure inherent in

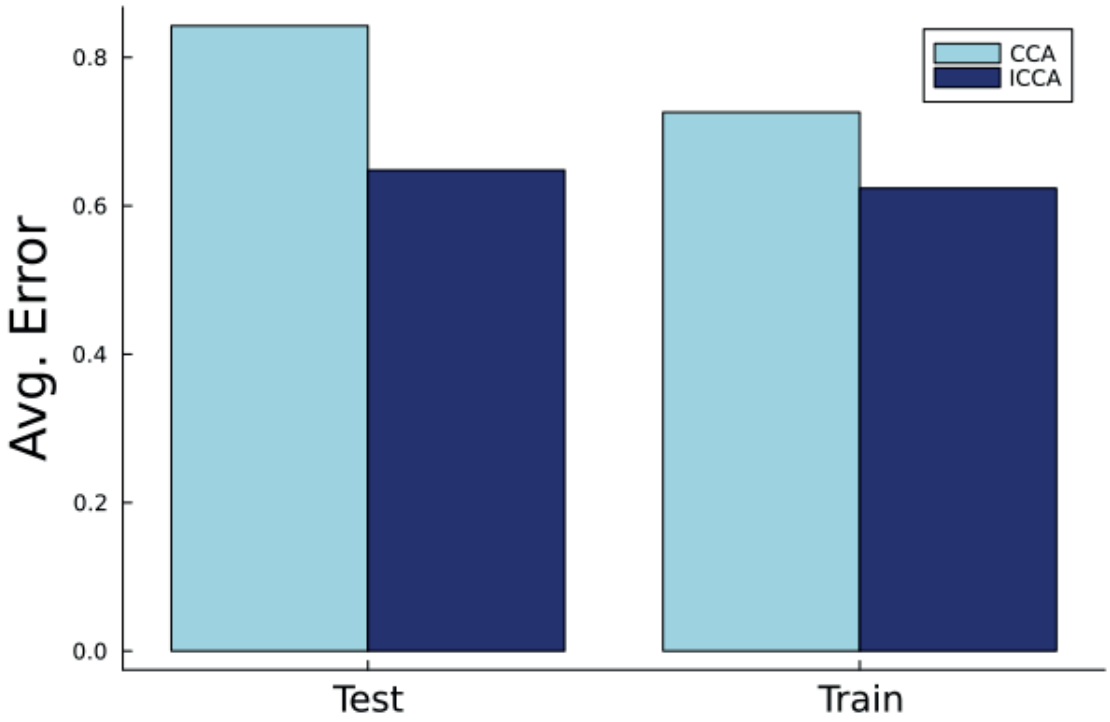

Fig. 3. MSE between the true and the reconstructed configurations of the anthropomorphic robotic arm for CCA and ICCA in training and testing.

many datasets, as a considerable portion of these datasets exhibit greater simplicity within their correct spatial representations.

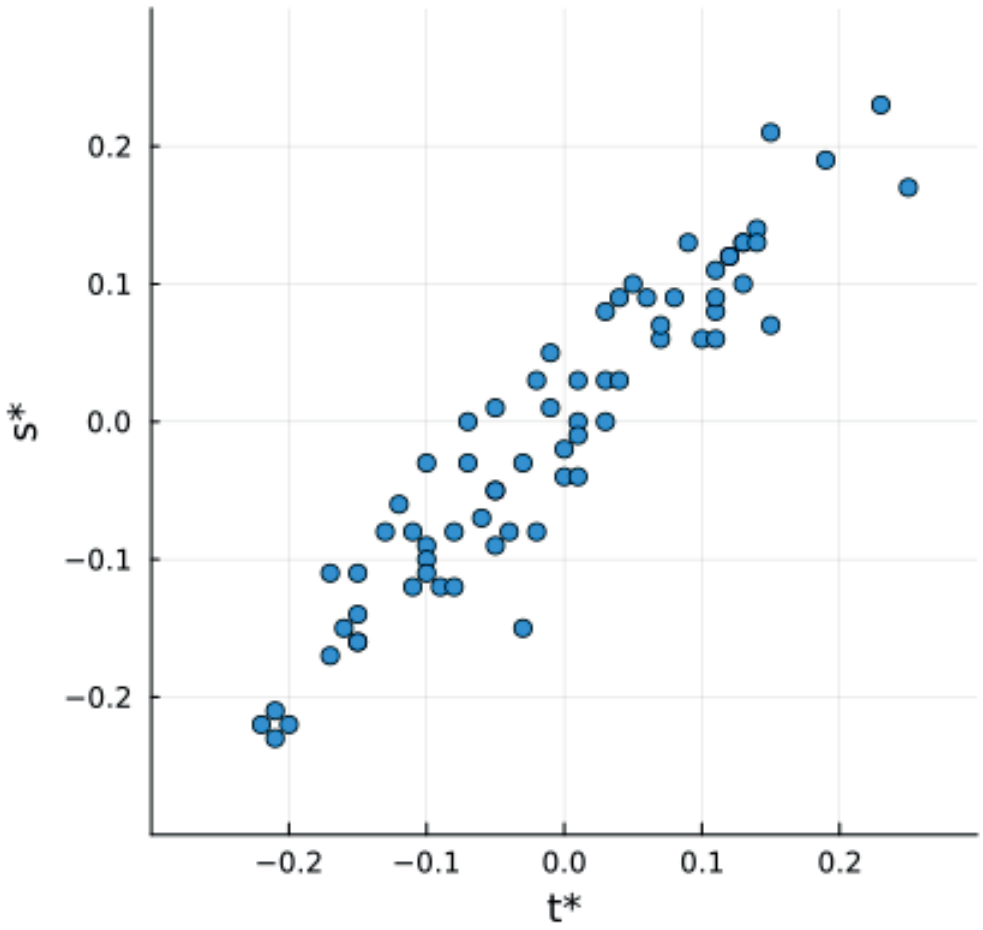

Fig. 4. **Optimal Projection Time**. t^* vs s^* comparison that uses the canonical pair to map to the original data. t^* and s^* show a linear relationship between one another. Thus a simple linear regression model would be suitable to map t^* to s^*.

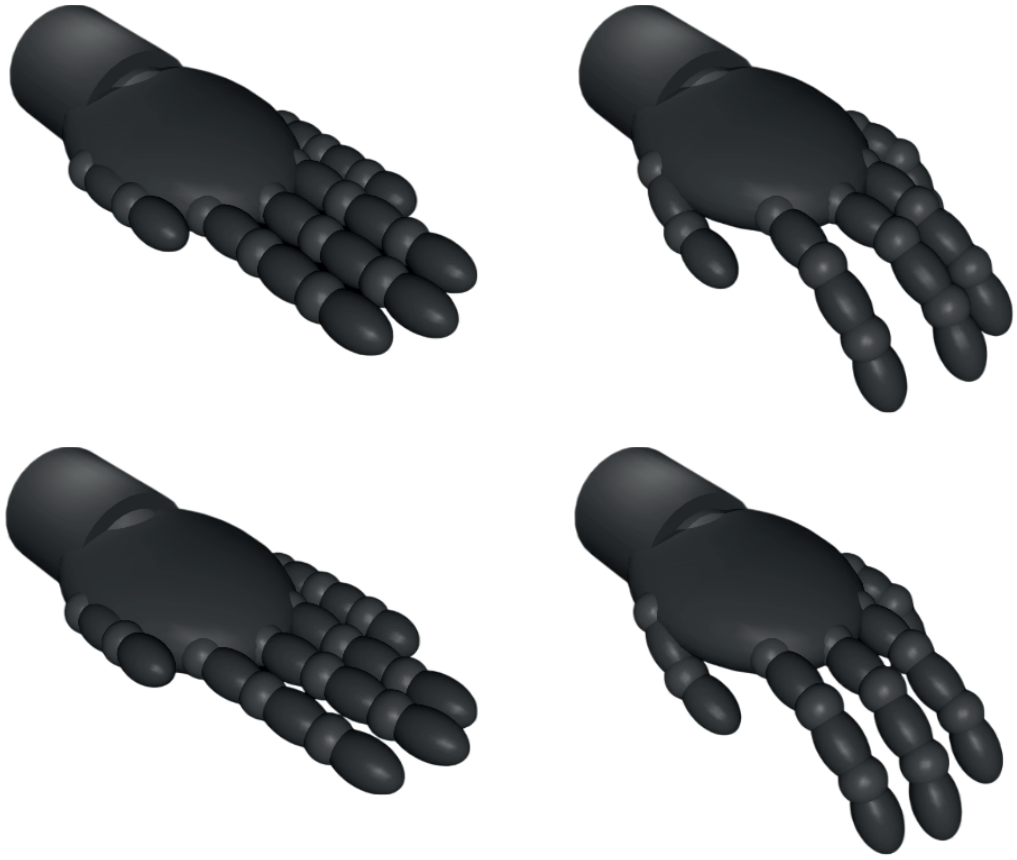

Fig. 5. Anthropomorphic Robotic Hand used in Sec.6. The images above show one of the simulations along with its ICCA reconstruction. Top-left represents the original state. Bottom-left represents the initial configuration (the neutral state with noise). Top-right represents the ground-truth configuration. Bottom-right represents the ICCA reconstruction of the final configuration given the initial configuration.

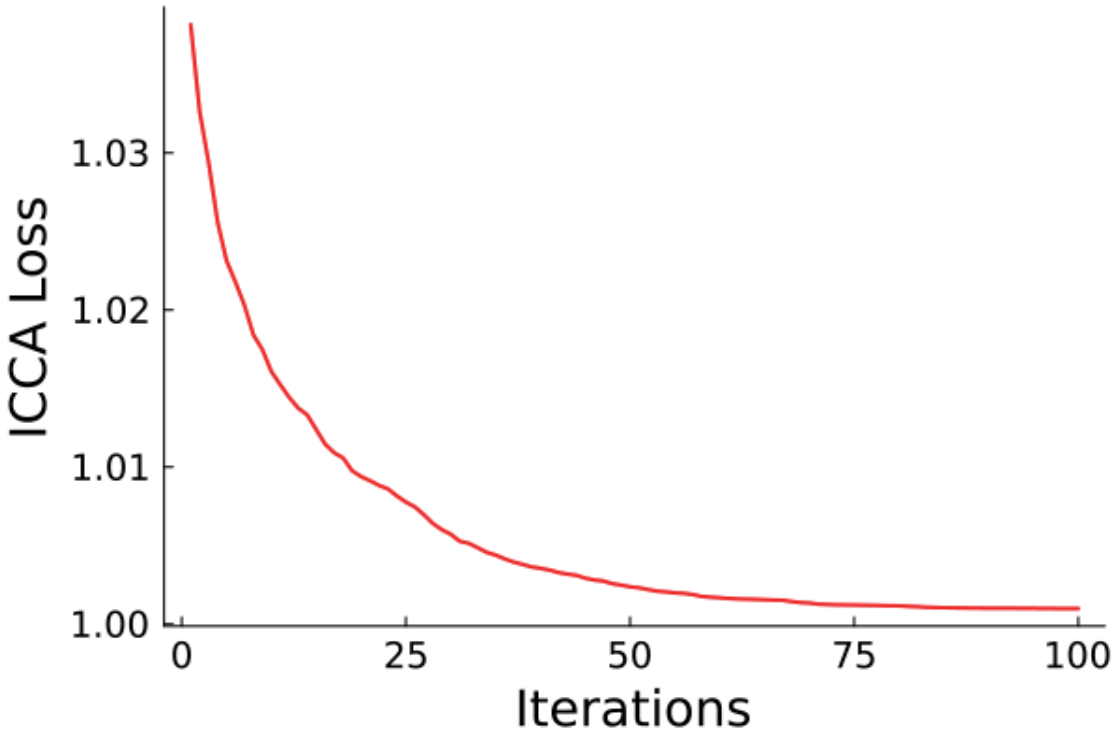

Fig. 6. Convergence of the ICCA loss for Grasping Hand in the iteration stage. The loss plot is created by taking the average on 10 sets of experiments on the grasping hand task. The curve is based on loss in the iteration stage of ICCA Decomposition. There is an additional loss decrease in the initialization stage.

7 FUTURE WORK

In this work, we were able to successfully apply a non-linear approach to CCA. From this, we were able to significantly improve the performance by simplifying the compression by representing the data in its inherent structure (Lie manifold). However, our work only focuses towards CCA in terms of the Lie group. We defer to future work the theoretical study of the relationship between the optimal projection times, which was found to be linear in the Lie group and the extension of the introduced ICCA to the data on different manifolds which are not necessarily Lie groups.

8 CONCLUSIONS

This work presents a novel method to generalize CCA to the nonlinear setting of a Lie group. The distance optimality, projection criteria, and subspace concepts generalize naturally to the Lie setting via intrinsic Riemannian distances and geodesics, respectively. This setting is a central application in the context of articulated robotic devices [41].

The projection-based approach opens doors to symmetry-aware methodologies, among these, the learning of parameters in a transformation model.

Our formalism expressly uses the group-theoretic properties of the Lie group rather than merely approximating the Lie group, e.g. via kernel-based approaches. However, further refinements are based on explicitly symmetry-respecting learning methods, such as kernels [42]. With the presented generalization, the option to incorporate further direct tools from the theory of groups has now become available to enhance the quality of treatment of systems with intrinsically symmetric structures.

We also emphasize that, like the whole family of PCA, CCA, and their informational generalization, the Information Bottleneck methods, the ICCA method enables a controlled hierarchical dimensional reduction of dynamical control systems respecting the constraint manifold and thus allows us to control complexity without losing the structural guarantees enforced by the constraints, and thus to substantially limit the performance loss due to the approximation.

All the experiments and results can be reproduced by our code repository:
https://github.com/JWK7/ICCA

Literature Cited

[1] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *37th Annual Allerton Conf. on Communication, Control, and Computing*, 2000, pp. 368–377.

[2] N. Slonim, "The information bottleneck: Theory and applications," Ph.D. dissertation, Hebrew Univ. of Jerusalem, Israel, 2002.

[3] T. Katayama *et al.*, *Subspace Methods for System Identification*, vol. 1. Communications and Control Engineering, 2005.

[4] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, "Information bottleneck for gaussian variables," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds., 2003, pp. 1213–1220.

[5] F. Creutzig, A. Globerson, and N. Tishby, "Past-future information bottleneck in dynamical systems," *Physical Review E*, vol. 79, no. 4, pp. 41 925–41 929, 2009.

[6] N. Amir, S. Tiomkin, and N. Tishby, "Past-future information bottleneck for linear feedback systems," in *54th IEEE Conf. on Decision and Control*, 2015, pp. 5737–5742.

[7] W. Chung, D. Polani, and S. Tiomkin, "Dimensionality reduction of dynamics on lie groups via structure-aware canonical correlation analysis," in *American Control Conf. 2024*, April 2024.

[8] A. Klami, S. Virtanen, and S. Kaski, "Bayesian canonical correlation analysis," *Journal of Machine Learning Research*, vol. 14, no. 4, pp. 965–1003, 2013.

[9] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Int. Conf. on Machine Learning*, 2013, pp. 1247–1255.

[10] W. Wang, R. Arora, K. Livescu, and N. Srebro, "Stochastic optimization for deep cca via nonlinear orthogonal iterations," in *53rd Annual Allerton Conf. on Communication, Control, and Computing*, 2015, pp. 688–695.

[11] P. T. Fletcher, S. Joshi, C. Lu, and S. M. Pizer, "Gaussian distributions on lie groups and their application to statistical shape analysis," in *18th Int. Conf. on Information Processing in Medical Imaging*, 2003, pp. 450–462.

[12] M. Moakher, "Means and averaging in the group of rotations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 1–16, 2002.

[13] A. Bonfiglioli and R. Fulci, *Topics in Noncommutative Algebra: The Theorem of Campbell, Baker, Hausdorff and Dynkin*, vol. 2034. Springer Science & Business Media, 2011.

[14] P. Fletcher, C. Lu, and S. Joshi, "Statistics of shape via principal component analysis on lie group," in *2003 Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 95–101.

[15] A. Agrachev and Y. Sachkov, *Control Theory from the Geometric Viewpoint*, vol. 87. Springer Science & Business Media, 2013.

[16] P. Agarwal *et al.*, "The banana distribution is gaussian: A localization study with exponential coordinates," in *Robotics: Science and Systems VIII*, 2013, pp. 265–272.

[17] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "From intrinsic optimization to iterated extended kalman filtering on lie groups," *Journal of Mathematical Imaging and Vision*, vol. 55, pp. 284–303, July 2016.

[18] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[19] J. Hoffmann *et al.*, "Training compute-optimal large language models," in *36th Conf. on Neural Information Processing Systems*, 2022, pp. 30 016–30 030.

[20] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd Int. Conf. on Learning Representations*, 2013.

[21] A. Alemi, I. Fischer, J. Dillon, and K. Murphy, "Deep variational information bottleneck," in *5th Int. Conf. on Learning Representations*, 2019.

[22] N. Slonim and N. Tishby, "The power of word clusters for text classification," in *23rd European Colloquium on Information Retrieval Research*, 2001.

[23] A. Wieczorek, M. Wieser, D. Murezzan, and V. Roth, "Learning sparse latent representations with the deep copula information bottleneck," in *6th Int. Conf. on Learning Representations*, 2018, pp. 612–616.

[24] R. Dubey, P. Agrawal, D. Pathak, T. L. Griffiths, and A. A. Efros, "Investigating human priors for playing video games," in *35th Int. Conf. on Machine Learning*, 2018, pp. 1349–1357.

[25] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[26] J. Cahill, D. Mixon, and H. Parshall, "Lie pca: Density estimation for symmetric manifolds," *Applied and Computational Harmonic Analysis*, vol. 65, pp. 279–295, 2023.

[27] T. Kim, S. Wong, and R. Cipolla, "Tensor canonical correlation analysis for action classification," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[28] R. Arora and K. Livescu, "Kernel cca for multi-view learning of acoustic features using articulatory measurements," in *Proc. of the Symposium on Machine Learning in Speech and Language Processing*, 2012, pp. 33–37.

[29] G. Lisant, I. Masi, and A. Bimbo, "Matching people across camera views using kernel canonical correlation analysis," in *Proc. of the Int. Conf. on Distributed Smart Cameras*. ACM, 2014, pp. 1–6.

[30] L. Chen *et al.*, "Adaptive asynchronous control system of robotic arm based on augmented reality-assisted brain-computer interface," *Journal of Neural Engineering*, vol. 18, no. 6, p. 66005, 2021.

[31] L. Melas-Kyriazi, "The mathematical foundations of manifold learning," Ph.D. dissertation, Harvard Univ., Boston, MA, 2020.

[32] A. N. Gorban and I. Y. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philisophical Transactions of the Royal Society A*, vol. 376, p. 237, 2018.

[33] Y. Ni, P. Koniusz, R. Hartley, and R. Nock, "Manifold learning benefits gans," in *2022 Conf. on Computer Vision and Pattern Recognition*, 2022, pp. 11 255–11 264.

[34] M. Herrmann, D. Kazempour, F. Scheipl, and P. Kröger, "Enhancing cluster analysis via topological manifold learning," *Data Mining and Knowledge Discovery*, vol. 38, pp. 840–887, 2022.

[35] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[36] D. Belanger and A. McCallum, "Structured prediction energy networks," in *33rd Int. Conf. on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48, 2016, pp. 983–992.

[37] G. Durrett and D. Klein, "Neural crf parsing," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 302–312.

[38] J. D. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *18th Int. Conf. on Machine Learning*, 2001, pp. 282–289.

[39] J. Selig, "Lie groups and lie algebras in robotics," in *Computational Noncommutative Algebra and Applications*, 2004, pp. 101–125.

[40] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[41] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[42] R. Kondor, "Group theoretical methods in machine learning," Ph.D. dissertation, Columbia Univ., New York, NY, 2008.

[43] R. Howe, "Very basic lie theory," *The American Mathematical Monthly*, vol. 90, no. 9, pp. 600–623, 1983.

[44] B. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*, ser. Graduate Texts in Mathematics. Springer, 2015.

[45] G. Bourmaud, R. Mégret, M. Arnaudon, and A. Giremus, "Continuous-discrete extended kalman filter on matrix lie groups using concentrated gaussian distributions," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 209–228, 2015.

[46] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer, 2014.

Appendix A

A finite-dimensional Lie group is a group that is at the same time a differentiable manifold [43], [44]. For every point $x \in G$ on the manifold, there exists a tangent linear vector space TG_x. The tangent space at the identity element, TG_e, is special. It is called the *Lie algebra*, $\mathfrak{g}$, of the Lie group, G. One can map elements of the Lie group (manifold) to those of its algebra (linear vector space with an associative bilinear product) and vice versa by:

$$\mathfrak{g} \underset{\exp}{\overset{\log}{\rightleftharpoons}} G \tag{29}$$

where 'exp' and 'log' are calculated via the corresponding Taylor series of the operators. Concretely, $\forall g \in \mathfrak{g}$, $\exp(g) = \sum_{n=0}^{\infty} \frac{g^n}{n!}$, where $g^n = g \circ g \circ \cdots \circ g$ is the n-fold product of g in the Lie algebra and the sum taken in the tangent space. Here we interpret the expression in terms of the matrix representation of the algebra.

The Lie algebra, as a linear vector space, is spanned by a basis of k elements, $E = \{E_1, E_2, \ldots, E_k\}$, where k is the dimension of the manifold G and $E_i \in \mathbb{R}^k$. Thus every element in the algebra, $g \in \mathfrak{g}$, interpreted as k-dimensional vector, is represented by a unique linear combination of the basis elements

$$g(\alpha) = \sum_{i=1}^{k} \alpha_i E_i, \text{ with } k \text{ scalars }, \alpha = \{\alpha_i\}_{i=1}^{k}. \tag{30}$$

Eq. (29) and (30) induce a mapping of the collection α to the Lie group, G, as follows: $G(\alpha) = \exp(g(\alpha)) \in G$.

The formalism is of particular interest for the special case of articulated robotic systems, since its configuration space (under multiple concatenated links) is represented by the Lie groups of rotations and translations, reviewed below.

A.0.1 Groups of rotation and translation

The Lie groups of 2D/3D spatial rotations, $SO(2)$/$SO(3)$, and translations, $SE(2)$/$SE(3)$, fully characterize the primitive geometric motions of rigid bodies, and are widely used in

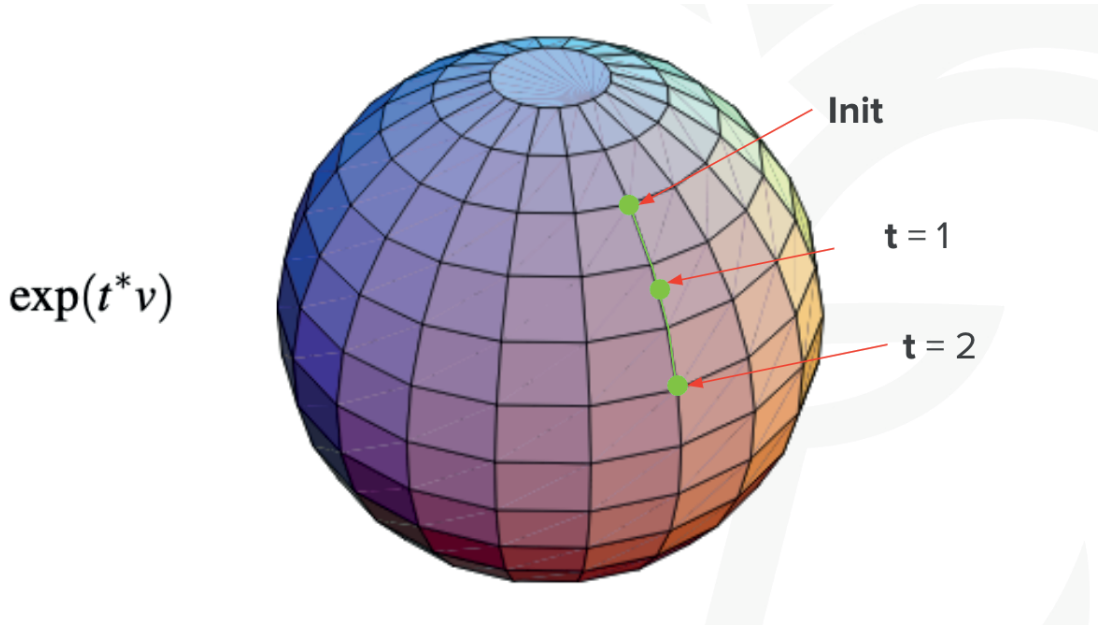

Fig. 7. Here we can see a 3D rotation represented through the $SO(3)$ group. The 3D rotation can be represented through the surface of a sphere. The group movement contains two components, the arbitrary unit vector of the 3D movement t and the scale of the movement t.

robotics [39], [45], [46]. Both groups admit matrix representations [44]. The group composition operator, $\circ$, is the standard matrix multiplication and exp the matrix exponential.

The exponential map from the lie algebra to the corresponding Lie group in Eq. (29) can be explicitly calculated by the Rodrigues rotation formula:

$$R = \exp\big(\theta U\big) = I + U\sin\theta + U^2(1-\cos\theta) \in \mathbb{R}^{3\times 3}, \tag{31}$$

where I is the identity matrix, $\theta \in \mathbb{R}$ the angle of rotation and U the matrix (the element of the Lie algebra) generating the rotation.

The full form can be seen in:

The 'logarithmic map':

$$\log(R) = \frac{\theta(R-R^T)}{2\sin\theta}, \qquad \theta = \cos^{-1}\left(\frac{\text{trace}(R)-1}{2}\right)$$

Together with the Rodrigues rotation formula, it allows us to effectively calculate the intrinsic distance and the projections to subgroups, which are the important components of the proposed ICCA method.

In 3D rotation, the movement in lie algebra space is described by its Euler vectors x, y, z and its angle θ. The lie algebra matrix is represented as:

$$R = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{32}$$

From this, we can use exponential mapping 31 to get:

$$R = \begin{bmatrix} 1-2s^2+2x^2s^2 & 2xys^2-2zsc & 2xzs^2+2ysc \\ 2xys^2+2zsc & 1-2s^2+2y^2s^2 & 2yzs^2-2xsc \\ 2xzs^2-2ysc & 2yzs^2+2xsc & 1-2s^2+2z^2s^2 \end{bmatrix} \tag{33}$$

Where, x, y, z are the Euler vectors, s is $\sin\frac{\theta}{2}$ and c is $\cos\frac{\theta}{2}$ respectively.

From this, we can map the Euler rotations into the group space, where we can intrinsically create 3D rotational movement. We can see the lie group representation of the 3D rotation in Fig. 7.