Trajectories in Hand Synergy Spaces for in-hand manipulation

Jorge Telo
Email: jorge.c.telo@tecnico.ulisboa.com

***Abstract*—Postural synergies are an important case of study to facilitate the control of dexterous artificial robot hands. To achieve this, latent spaces (synergy spaces) are generated and trained from grasp postures and are used to directly control a robot hand in this space. In this work, we base ourselves in an already achieved model and propose the addition of specific Riemannian metrics to achieve better results than the original model. To evaluate the results, the values for reconstruction error and accumulated smoothness of paths in a space are used to evaluate in-hand regrasping tasks. Different Riemannian metrics lead to different results and should be used with different tasks in mind. The arguments presented in this work are validated by calculating accumulated smoothness based on a set group of in-hand regrasping tasks.**

I. Introduction

In an age characterized by rapid technological advancements, engineers have been driven to explore innovative avenues in replicating the intricate functions of the human body through artificial means. This endeavor has many applications, ranging from the creation of medical prosthetics to enhancing industrial processes.

A notable area of focus in this realm is the human hand, which is often perceived as a deceptively simple tool for gripping objects and executing various movements. However, beneath this apparent simplicity lies a complex system of muscles, tendons, nerves, and bones, replete with a multitude of Degrees of Freedom (DoF) [1]. Controlling such a system within a computational environment demands substantial processing power. Fortunately, nature has incorporated inherent constraints into this system. These constraints, such as the inability of fingers to bend backward and the interconnected nature of muscles and tendons, effectively reduce the number of DoF [1], [2]. This reduction in control space simplifies the task of implementing a control system for a robotic dexterous hand.

Our study builds upon prior work, particularly [3], with the objective of further enhancing the model's solution to the dimensionality challenge. This project leverages the Conditional Variational Auto-Encoder (CVAE) model, originally developed in [3], to generate a smoother latent space. By introducing metrics during the model's training process and focusing on characteristics of the real space representing the DoF in robotic hands, we aim to create a smoother latent space. Ultimately, this will make regrasping more efficient and cost-effective.

In essence, our contributions involve refining the model from [3] by incorporating Riemannian metrics into the generated latent space. This endeavor seeks to improve the modeling of latent spaces associated with robotic hands, with the ultimate goal of facilitating regrasping with reduced effort and cost. Additionally, it lays the groundwork for more intricate latent space modeling and helps us understand the impact of various metrics and parameter adjustments.

II. Related Work

In the pursuit of understanding human hand control and its application to robotic systems, numerous studies have highlighted the presence of synergies. Synergies refer to the coordinated actions of neural signals, muscles, and kinematic constraints in the human hand that allow for efficient grasp and movement. These findings have broad implications for enhancing robotic control and grasp strategies.

Studies such as [1] and [2] have demonstrated that individual finger control is not necessary for various hand grasps and movements. Instead, the human brain leverages synergistic frameworks that exploit neural signals, muscle coordination, and kinematic constraints to streamline hand function.

Building upon these insights, research in [4] and [5] employed Principal Component Analysis (PCA) to discover low-dimensional representations for 12 distinct hand grasps and postures. These representations, known as *eigengrasps*, offer a basis for creating new grasps and postures. Furthermore, it was observed that control algorithms performed more effectively when utilizing these low-dimensional spaces to search for stable grasps.

Another study, [6], applied PCA to precision grasps achieved by human operators controlling dexterous robotic hands. The research represented the discovered synergies in robot joint angle space, demonstrating that even complex precision grasps could be effectively expressed in 5 or 6 dimensions.

[7] and [8] compared various dimensionality reduction methods for encoding robotic hand control in 2D subspaces. They evaluated these methods based on representation consistency and continuity. Additionally, [7] proposed a method for denoising graphs of embedded grasp postures, ultimately favoring the Isomap method for its ability to generate more consistent and continuous embeddings.

In [9] and [10], a more advanced approach utilizing Latent Variable Models (LVM), specifically Gaussian Process Latent Variable Models (GPLVM), was introduced. These models shifted the problem from a deterministic to a probabilistic nature. The evaluation considered the reconstruction error and semantic evaluation. Notably, GPLVM outperformed PCA in

terms of reconstruction error, implying that the nature of hand synergy is nonlinear and may benefit from a more fitting nonlinear approach.

These studies collectively suggest that while PCA can demonstrate the existence of dimensionality reduction, more complex models may be advantageous when aiming to improve grasp encoding and decoding for multifaceted tasks. Additionally, introducing back constraints to GPLVM, as seen in [11], helps maintain proximity between points in both high-dimensional and latent spaces.

As computational power has grown, researchers have explored more advanced models. The use of Auto-Encoders (AE) in [12] and [13] exemplified the benefits of nonlinear, deterministic dimensionality reduction methods based on neural networks. This approach improved performance in terms of reconstruction and enabled the incorporation of additional information, such as object size and shape, expanding the possibilities for robotic hand simulations and grasping strategies.

III. Background

This section is divided into two sections. One related for the Dimensionality problem and the last one related to Riemannian Geometry, specifically Riemannian Metrics.

A. Reduction of Dimensionality

In the context of replicating hand synergies in robotic systems, a significant challenge involves reducing the dimensionality of the control space. Dimensionality reduction techniques aim to transform a high-dimensional space, denoted as real space R^{d_x}, into a lower-dimensional space, known as the latent space R^{d_z}, where $d_z < d_x$. This transformation is typically governed by two parameterized mappings, ϵ and d, represented by vectors θ and ϕ. Various methods use these mappings to minimize selected optimization criteria with a given dataset X.

The problem of implementing hand synergy in robotic hands is often approached as an optimization problem related to dimensionality reduction, solvable through software solutions.

PCA is a fundamental method that assumes a linear mapping, ϵ from the high-dimensional space to the low-dimensional space. This mapping is represented as $\epsilon X = XW$, where $W \in R^{d_x \times d_z}$. The most common optimization criteria in PCA is the minimization of the mean squared error between the original input and the reconstructed signal from the embedding. This optimization problem can be analytically solved to compute the optimal value for W, which corresponds to the principal components of X.

The Auto-Encoder (AE) is another widely used method, involving an artificial neural network for unsupervised learning. An AE learns to represent data efficiently by mapping high-dimensional inputs, x to low-dimensional signals, z using an encoder, and then reversing the operation using a decoder. Unlike the PCA model, the relationship between high-dimensional and low-dimensional spaces in AE is nonlinear. However, both methods utilize the same optimization criteria, the mean squared error.

Latent Variable Models (LVMs) offer a probabilistic approach to dimensionality reduction. These models assume that high-dimensional inputs,x, are generated by unobserved latent variables,z, through a probability distribution. LVMs aim to represent this mapping, typically involving the parameters to be learned θ, which can be mathematically formulated as $p(x, z; \theta)$. One common LVM is the Latent Variable Model based on Gaussian Processes (GPLVM), which employs Gaussian Processes to connect latent and high-dimensional spaces. Optimization in GPLVM often focuses on maximizing the likelihood of the data through gradient-based methods. For these models, the mapping ϵ has the following distribution:

$$P(\mathbf{Z}|\mathbf{X}, \theta) = \prod_{j=1}^{D} \frac{1}{(2\pi)^{\frac{N}{2}} |\mathbf{K}|^{\frac{1}{2}}} e^{-\frac{1}{2}\mathbf{z_j}^T \mathbf{K}^{-1} \mathbf{z_j}} \tag{1}$$

While LVMs offer a robust approach to dimensionality reduction, encoding new data points can be computationally expensive due to the iterative optimization process required for inference.

In [10] and [12], the authors utilized a variant of GPLVM, with back constraints added to the model, [11], for synergy extraction. This model enforces a constraint on the latent variables that ensures that points that are close in the original space remain close in the latent space.

In this work, as mentioned earlier, the dimensionality problem is addressed using a Conditional Variational Auto-Encoder (CVAE) model, based on the one presented in [3]. The CVAE is derived from the Variational Auto-Encoder (VAE), which is a variant of the AE architecture and offers an efficient framework for parameter learning. Both VAE and CVAE consist of an encoder network $q_\phi(z|x)$ and a decoder network $p_\theta(x|z)$.

The key difference between AE and VAE is that VAE samples the latent variable,z, from a Gaussian distribution with mean, μ, and variance, σ, computed by the encoder when given an input data point x. VAE training employs variational inference to minimize the Kullback-Leibler (KL) divergence between the encoder's input and the decoder's output. This is achieved by maximizing the evidence lower bound (ELBO), which is computationally more manageable.

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta \left(x|z\right)\right] - D_{KL} \left(q_\phi \left(z|x\right) \| p(z)\right) \tag{2}$$

The ELBO in equation 2 comprises two terms: the first term represents Mean Squared Error (MSE), while the second term captures the KL divergence, preserving the distribution of the real space in the latent space.

In [3], the author extends this model by introducing a conditional variable, denoted as c following the procedure described in [14], thereby transforming the model into a CVAE. This c variable is used to condition both the encoder and the decoder network. It can be either continuous or discrete, which modifies the ELBO as shown in equation 3.

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x,c)}\left[\log p_\theta\left(x|c,z\right)\right] - D_{KL}\left(q_\phi\left(z|x,c\right)\|p(z)\right) \tag{3}$$

To optimize the loss function indicated by equation 3, standard gradient descent algorithms are applied, similar to those presented in [15].

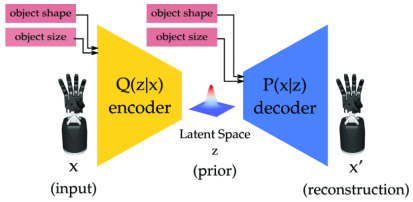

Fig. 1. CVAE architecture as proposed in [3]

This study utilizes the dataset provided in [6], and the author's conclusions highlight that although the CVAE model may exhibit a worse result in terms of MSE as a reconstruction metric compared to approaches like PCA, it excels in learning a smoother latent space. This property translates into improved performance in tasks that involve transitioning between grasps smoothly.

B. A glimpse into Riemannian Geometry

One challenge in dimensionality reduction problems is the potential loss of geometric relations between points in the real space when transitioning to a latent space. This discrepancy can hinder efficient task completion when traversing the latent space. To address this issue, Riemannian Geometry is employed.

Riemannian Geometry, [16], is a branch of differential geometry that focuses on Riemannian Manifolds, well-defined metric spaces where the inner product is locally defined and varies smoothly across the entire space.

In this context, a smooth manifold is envisioned as a non-intersecting surface embedded in an ambient space. Tangent spaces to the manifold at specific points play a crucial role in this geometry. The Riemannian metric, which describes the relationships between points, can be flexible and designed to encode high-level information.

The utility of Riemannian Manifolds lies in simplifying calculations. By utilizing a Riemannian metric to design a lower-dimensional space $Z \in R^{d_z}$ that captures proximity information from the high-dimensional space $X \in R^{d_x}$, shortest paths in X can be computed in Z), reducing computational energy requirements. Furthermore, introducing this metric into the learning model for creating the latent space enables the discovery of more sophisticated relations between points in the real space.

Previous works have demonstrated the advantages of Riemannian metrics. In [17], the author explains how these metrics are valuable for calculating shortest paths in manifolds derived from high-dimensional datasets.

In [18], the consideration of the high-dimensional space as a Riemannian manifold, rather than a Euclidean space, is explored. This approach allows the encoding of domain knowledge through well-defined Riemannian metrics, facilitating the definition of appropriate shortest paths in the latent space. Carefully designed metrics tailored to specific tasks can ensure well-behaved shortest paths, even in cases of deterministic generators that often suffer from misleading biases.

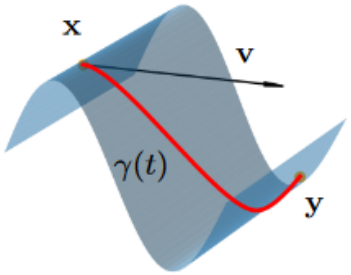

Fig. 2. Example of a Riemannian manifold

IV. Methodology

The methods described in this section will be employed to enhance the encoding model of a high-dimensional space. The objective is to create a new model that captures more intricate relationships among data points in a lower-dimensional space.

In practical terms, this means that when navigating the latent space, the new model should offer shorter paths between encoded grasps compared to the original CVAE model.

The metric used to measure the shortest path in this work is calculated as the sum of differences in each dimension among a set of grasps within a given path. This metric is applied in the high-dimensional space, which corresponds to the operational space of the hand to be controlled.

A. CVAE Model

The base model used in this work is based on the model presented in [3], and the grasp dataset employed aligns with the one used by the author in [6].

The CVAE model is grounded in the VAE architecture. It consists of an encoder responsible for transforming high-dimensional data into a lower-dimensional representation and a decoder for the reverse process.

In the VAE model, a $D_x = \{x_1, x_2, x_3, ..., x_n\}, x_i \in R^{d_x}$ generated from the space $X = R^{d_x}$ (containing saved postures from [6]), is used to train an encoder. This encoder models a probability distribution $q(z|x)$, enabling the encoding of a data point x_i from the high-dimensional space into a lower-dimensional data point z_i from $Z = R^{d_z}, d_z \leq d_x$. The decoder is trained to perform the inverse operation, transforming a lower-dimensional data point z_i into a higher-dimensional data point x_i using a dataset $D_z = \{z_1, z_2, z_3, ..., z_n\}, z_i \in R^{d_z}$.

The loss function employed to train the model, as given in equation 2, comprises two terms: the first term measures the reconstruction error between the encoder input and decoder output, while the second term quantifies the KL divergence

between the latent variable's probability distribution and a standard Gaussian distribution. Optimizing this second term of the loss function results in a smoother latent space and reduced reconstruction error. This optimization process is carried out using standard gradient descent methods.

The produced latent space $Z = R^{d_z}$ represents the synergistic components of the hand, necessitating decoding to achieve a physical representation by moving the robotic hand.

In generating the CVAE model, an additional variable, denoted as c, is introduced as a condition. This variable contains information about the object to be grasped, specifically its size and shape. This addition requires adjustments to the dataset formulations and probability densities. The high-dimensional dataset becomes $D_x = \{(x_1, c_1), (x_2, c_2), (x_3, c_3), ..., (x_n, c_n)\}, x_i \in R^{d_x}, c_i \in R^{d_c}$. The encoder now models the distribution $q(z|x, c$. The latent space dataset becomes $D_z = \{(z_1, c_1), (z_2, c_2), (z_3, c_3), ..., (z_n, c_n)\}, z_i \in R^{d_z}, c_i \in R^{d_c}$, and the decoding distribution is $p(x|z, c$.

As a result of these changes, the loss function undergoes modifications, as illustrated in equation 3. The inclusion of the conditional variable in the model enhances the robustness of the latent space by associating specific grasps with particular object sizes and shapes. This can lead to a more effective learning pattern for uncovering hidden relationships in the original data.

B. Riemannian Metrics

To capture more intricate information about the geometric relationships between points in the high-dimensional space, Riemannian metrics, as described in [18], were employed.

The Riemannian metric used serves to define the geometric properties of the high-dimensional space. This metric determines how distances between points in the space should be measured, taking into account the underlying structure and relationships within the dataset. To promote smoothness in the latent space, the metric is designed to preserve neighborhood relationships between points in the high-dimensional space. In essence, if two points are neighbors in the original space, the metric ensures that they also remain neighbors in the latent space when training the CVAE model.

Once the metric is defined, a square matrix $G, d_z \times d_z$ is derived from it and used to calculate geodesic distances. These distances represent the shortest paths along the curved surface of the Riemannian manifold, corresponding to the original high-dimensional space. With this matrix, additional terms are incorporated into the Loss function of the CVAE model during training, taking the metric into account.

The metrics considered in this context take into consideration both the distance between points, denoted as Dz, and the distance between each individual point and the center of the dataset, denoted as C_z. As a result, the Loss function adopts the following formula:

$$\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x,c)}\left[\log p_\theta\left(x|c,z\right)\right] - \\ - D_{KL}\left(q_\phi\left(z|x,c\right)\|p(z)\right) + D_z + C_z \quad (4)$$

C. Regrasping Algorithms

To evaluate the trained model, three different regrasping algorithms are utilized. These algorithms provide insights into how the model performs under varying training parameters. The resulting steps in a regrasping task are subsequently decoded to quantify their smoothness.

The first algorithm, a Naive Neighbor approach, explores the possibility of correctly modeling neighboring pairs in the high-dimensional space within the low-dimensional space. This algorithm seeks to identify whether the total smoothness of a path in latent space is consistent when using either high-dimensional or low-dimensional neighbors.

The second algorithm, based on the A* pathing algorithm, is employed to assess the efficiency of pathing between grasps. It aims to minimize unnecessary movement and find the shortest path between two points in the dataset. This algorithm investigates the model's capability for efficient pathing and reducing superfluous motion.

The final algorithm, an interpolation method, creates intergrasps by specifying the number of steps in a path. Notably, this algorithm generates new grasps not present in the training data. Its purpose is to test whether the trained model can generate previously unknown grasps and incorporate them into the task.

These algorithms serve as valuable tools for assessing the model's performance and its ability to provide smooth, efficient paths for grasping tasks.

V. Implementation of Metrics

Incorporating Riemannian metrics into the CVAE model involved the definition of three different metrics to serve various objectives:

- Metric 1: Designed to create clear clusters in the latent space based on object type, achieved by considering distances from points to their respective object type cluster center and distances between different cluster centers.
- Metric 2: Similar to Metric 1 but with an emphasis on smoother transitions between clusters, resulting in less distinct separations between object types.
- Metric 3: Focused on preserving relationships between points from the real space into the latent space, regardless of object type, allowing closest points to stay close and farthest points to remain distant.

To implement these metrics, square matrices G were created, one for each Metric.

For Metric 1 the diagonal entries are computed as the sum of the mean distances of each point from the same object type to its cluster center. These entries reflect how close points of the same object type should be to their cluster center. The off-diagonal entries represent the mean distances between cluster center points of different object types. These entries emphasize

the separation of clusters in the latent space. Scaling factors are introduced to control the influence of each term in the G matrix, allowing adjustments based on specific objectives.

Metric 2 is similar to Metric 1, but aims to allow smoother transitions between clusters in the latent space. The diagonal entries are calculated as the sum of mean distances of each point to its respective cluster center. These entries help create distinct clusters. The off-diagonal entries are determined by calculating the Euclidean distance between the closest points of different object types. This encourages smoother transitions between clusters. Scaling factors are applied to control the impact of each term. And for Metric 3 that focuses on preserving pairwise relationships between data points from the high-dimensional space into the latent space, regardless of object type. The diagonal entries represent the average distance of each point in the latent space to its closest point in the high-dimensional space, while the off-diagonal entries are computed as the distances between the two points in the high-dimensional space that have the maximum distance from each other. The scaling factors are used to adjust the influence of each term in the G matrix.

These G matrices play a crucial role in integrating Riemannian metrics into the model by connecting latent space distances with real space distances, allowing for fine-tuning of the model for different objectives.

The Riemannian metrics impacted the loss function by connecting latent space distances with real space distances. The new loss function, called the "Riemannian loss" integrated these metrics into the previous CVAE loss. It compared high-dimensional space pairwise distances with geodesic pairwise distances in the latent space. An additional metric measured the distance of data points to the center of the high-dimensional space and the geodesic distance of latent space data points to their center. The final loss function included weights for each term, allowing fine-tuning of the model for different objectives. It took the form:

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(x) = \mathbb{E}_{q_\phi(z|x,c)}\left[\log p_\theta\left(x|c,z\right)\right] \\ - w_{KL} \times D_{KL}\left(q_\phi\left(z|x,c\right)\|p(z)\right) \\ + w_{D_z} \times D_z + w_{C_z} \times C_z \end{aligned} \quad (5)$$

These weights determined the importance of each term, enabling customization of the model based on specific objectives.

A. Smoothness as a Evaluation Metric

The goal of this thesis is to find a way to achieve a better representation of a high dimensional space in a lower dimensional space, to make regrasping procedures more efficient. Efficiency in the context of this work is considered as smoothness. Smoothness is given by 6, where $f(z_i$ is the joint angle of a decoded posture.

$$S(z_1, z_2) = ||f(z_1) - f(z_2)||_2 \quad (6)$$

This perception of smoothness is presented in [19]. This means that the goal is to try and minimize this equation and can be described as trying to make it so neighboring points in the high dimensional space are also neighboring points in the latent space. If this is achievable, then when transversing the latent space utilizing paths, that are groups of grasps in a regrasping task, from one initial grasp to a final grasp, if one uses the shortest path, than that path leads to the smallest movement of the joint angles of the controlled hand.

VI. Results

A. Dataset Preparation

The data used for this thesis is based on the grasps postures registered in [6]. The models will be trained and tested for both datasets, one for the iCub robot hand and one for the Shadow Robot hand. For the iCub robot hand, the dataset presents 536 grasps and each of these grasps has the information of the object it was used on. This information is used to create the conditional variable, also called object type for purposes of implementation, used for training. For the Shadow robot hand, the dataset contains 438 grasps and their respective object information.

B. Reconstruction Error

The reconstruction error, reflecting the dissimilarity between the input dataset passed to the encoder and the dataset decoded by the decoder, is depicted in figure 3 for the iCub robot hand and in figure 4 for the Shadow robot hand. .

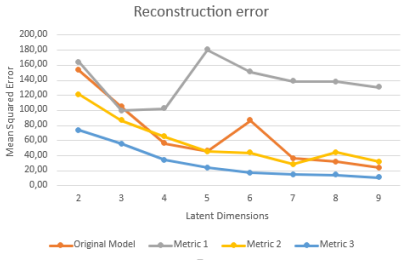

Fig. 3. Reconstruction Error of the model using iCub robot hand.

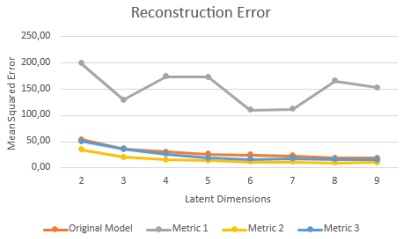

Fig. 4. Reconstruction Error of the model using Shadow robot hand.

In the iCub simulations, it's evident that both metric 2 and metric 3 yield lower reconstruction errors compared to the original CVAE model when using lower latent dimensions. This suggests that the new models can produce data points that more closely resemble the original input data. However, for metric 1, which enforces the separation of points into clusters, it leads to a higher reconstruction error than the

original model at higher latent dimension values, although it performs similarly at lower dimensions. This suggests that, while metric 1 promotes the creation of clusters in the latent space, the model can still utilize the information effectively to recreate the dataset.

In the case of the Shadow hand, except for the model using metric 1, all three attempts exhibit similar values for the reconstruction error. However, the use of metric 1, which emphasizes cluster formation, leads to higher error values. This issue arises from how the clusters are generated and influences the decoding process, resulting in increased errors.

C. Smoothness Testing

To assess whether the metrics achieve the desired results, pairs of grasps of the same object type are utilized, with one serving as the starting point and the other as the ending point. These pairs are used to generate paths in the latent space. The evaluation of these paths relies on the concept of *Accumulated Smoothness* which represents the cumulative differences in joint angles along a path. The number of paths evaluated corresponds to the number of tested pairs.

To present the results effectively, two methods are employed. The first method displays all the accumulated smoothness values for the various mechanisms within each model. The second method showcases the values obtained in each model for each pathing mechanism.

The Accumulated Smoothness metric is essential for understanding how smoothly the paths are executed within the latent space. While the Reconstruction Error provides insights into the model's ability to recreate the input data, the Accumulated Smoothness metric is crucial for evaluating the efficiency of the metrics and pathing mechanisms.

D. Accumulated Smoothness in a Model

In this section, the results of different pathing algorithms for each model are presented, with the objective of achieving the smallest accumulated smoothness, indicating minimal energy (joint angle differences) required for regrasping tasks.

The results are organized in the following sequence: starting with the original model's simulations, followed by the outcomes for metrics 1, 2, and 3, in that order. Graphs illustrate the accumulated smoothness for the various pathing mechanisms used.

Fig. 5. Original Model Pathing results on iCub robot hand.

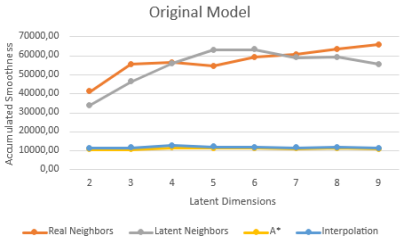

Fig. 6. Original Model Pathing results on Shadow robot hand.

Figures 5 and 6 depicts the results for the original model's pathing, showcasing similarities in accumulated smoothness for iCub and Shadow hands when using the Naive Neighbors algorithm, regardless of whether neighboring relationships are based on real space or latent space. The A* and Interpolation algorithms yield the smoothest paths, with the interpolation method also competing effectively with data-driven methods.

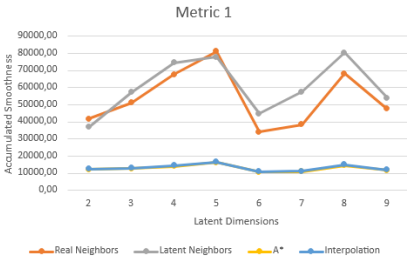

Fig. 7. Metric 1 Model Pathing results on iCub robot hand.

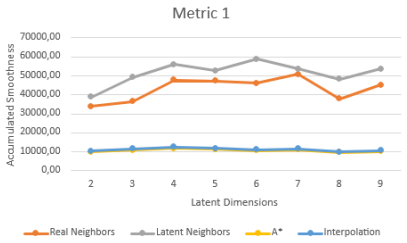

Fig. 8. Metric 1 Model Pathing results on Shadow robot hand.

Figures 7 and 8 presents the results for Metric 1, where points are grouped in the latent space based on their object type, and the centers of these clusters are encouraged to be distant from other centers. The most notable outcomes remain with the A* and Interpolation algorithms, demonstrating the benefits of both known data-driven methods and the interpolation technique.

For Metric 2 results, emphasizing object type-based point grouping and increased distances between the two closest points of different object types, the findings can be seen in figures 9 and 10. Once again, the A* and Interpolation approaches show favorable results, with the naive neighbor approach exhibiting varied performance based on the number of latent space dimensions.

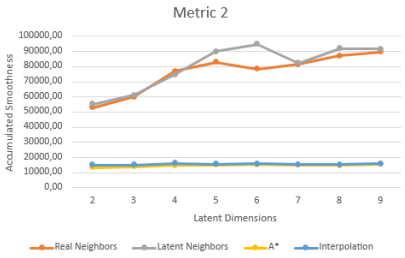

Fig. 9. Metric 2 Model Pathing results on iCub robot hand.

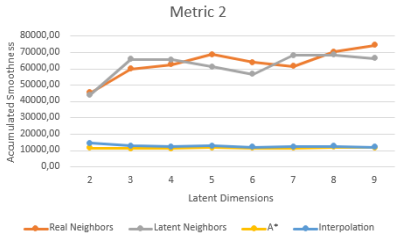

Fig. 10. Metric 2 Model Pathing results on Shadow robot hand.

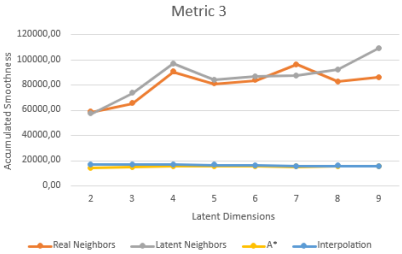

Fig. 11. Metric 3 Model Pathing results on iCub robot hand.

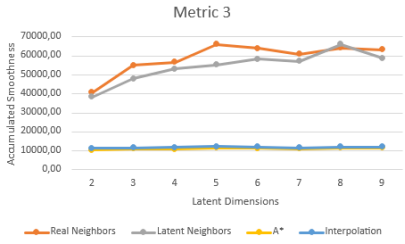

Fig. 12. Metric 3 Model Pathing results on Shadow robot hand.

The final results in this section correspond to Metric 3, focusing on capturing geometric relationships and potential hidden connections in the latent space. Figures 11 and 12 illustrate that the choice of using real space or latent space neighboring relationships depends on the hand being used. For the iCub hand, real space relationships lead to smoother paths, while for the Shadow hand, latent space relationships prove beneficial. This suggests that latent space can be more effective in creating regrasping tasks, potentially simplifying the hand's use by eliminating the need for extensive DoF inputs.

E. Accumulated Smoothness for a Specific Algorithm

This section aims to determine if the models trained with the newly developed metrics perform better or worse than the original CVAE model. Each model is tested using different algorithms, with the desired outcome being a smaller value of accumulated smoothness, preferably for most of the tested latent space dimension sizes. The results are presented in the same order as the algorithms were introduced. Starting with

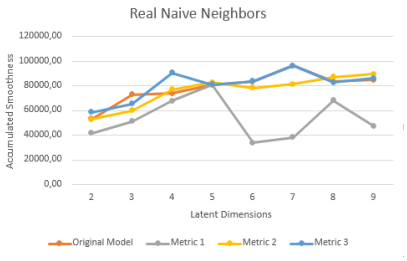

Fig. 13. Real Naive Neighbors Pathing results on iCub Robot Hand.

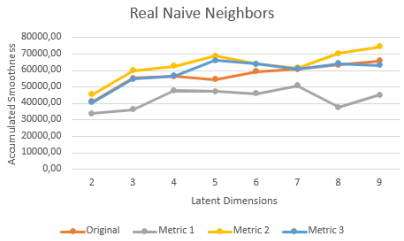

Fig. 14. Real Naive Neighbors Pathing results on Shadow Robot Hand.

the Real Naive Neighbors, where neighboring relationships are determined based on distances in the decoded real space, figures 13 and 14 provide the outcomes for iCub and Shadow Robot hands. For both hands, metric 1 consistently provides the best results for most possible dimension sizes. However, metric 1 exhibits a larger variance in the obtained values, while metrics 2 and 3, along with the original model, maintain more stable results across different sizes of latent space.

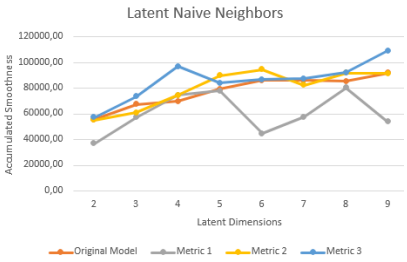

Fig. 15. Latent Naive Neighbors Pathing results on iCub Robot Hand.

Moving on to simulations for Latent Naive Neighbors, where neighboring relationships are based on distances in the latent space before decoding, figures 15 and 16 illustrate the results for both hands. For the iCub hand, metric 1 again shows the best results, with metric 3 presenting the least smooth

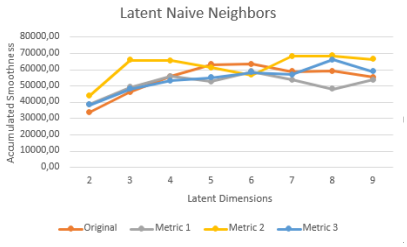

Fig. 16. Latent Naive Neighbors Pathing results on Shadow Robot Hand.

paths. Metrics 2 and the original model exhibit more stable results across varying latent space sizes. For the Shadow hand, the original model performs best for the lowest-dimensional latent space, while metric 3 outperforms the other two metrics when dimensions exceed 7.

The results suggest that creating a latent space where points are grouped by object type and where geometric relationships are promoted within these clusters performs well for regrasping tasks involving unchanged objects.

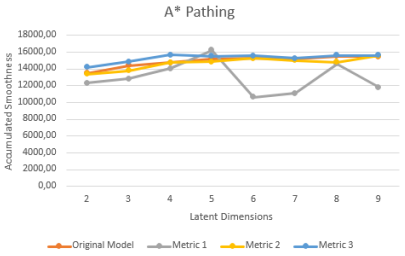

Fig. 17. A* Pathing results on iCub Robot Hand.

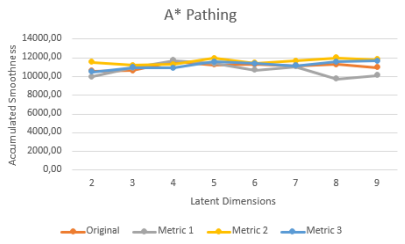

Fig. 18. A* Pathing results on Shadow Robot Hand.

Figures 17 and 18 present results for constructing the smoothest path using known data points with the A* algorithm. For both hands, metric 1 performs best in terms of accumulated smoothness, with results across all models showing consistency and minimal fluctuations as latent space dimensions increase.

The interpolation pathing results are evaluated to determine the effectiveness of creating new inter-grasps when performing regrasping tasks. As shown in figure 19, for the iCub hand, the original CVAE model yields promising results, albeit with variations based on latent space dimensions. The other three metrics offer more stable outcomes across different dimension

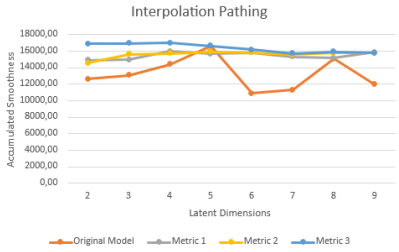

Fig. 19. Interpolation Pathing results on iCub Robot Hand.

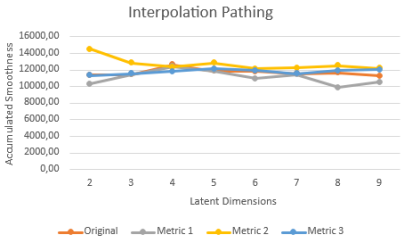

Fig. 20. Interpolation Pathing results on Shadow Robot Hand.

sizes. For the Shadow hand, all three metrics perform competitively with the original model, indicating their adaptability to creating new grasps for regrasping tasks.

F. KL Weight Effects

The KL divergence weight, as seen in equation 5, plays a crucial role in making the model align the distribution of points in the latent space with that of the original space. An increase in this weight theoretically compels the model to prioritize maintaining the distribution of the original space within the latent space. To assess the impact of this weight on the model, a simulation is conducted using Metric 3, as it specifically considers the geometric relationships of points in the real space.

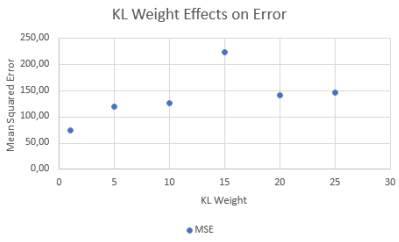

Fig. 21. Effects of KL weight on Reconstruction Error for iCub Robot hand.

Figures 21 and 22 illustrate the reconstruction error of the model using Metric 3 as the KL weight in the loss function is incremented. For the iCub hand, the error value increases with the weight's increments, with a sharp increase at w = 15. In contrast, the Shadow hand experiences a drop in the error value when increasing the weight from 1 to 5, but it slowly rises as the weight further increases. These results highlight the significant impact of the KL weight on the

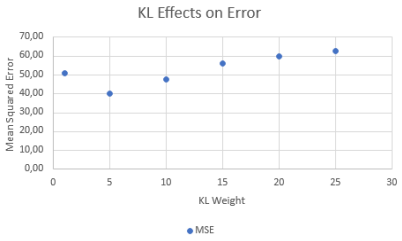

Fig. 22. Effects of KL weight on Reconstruction Error for Shadow Robot hand.

model's performance, emphasizing the importance of fine-tuning this parameter.

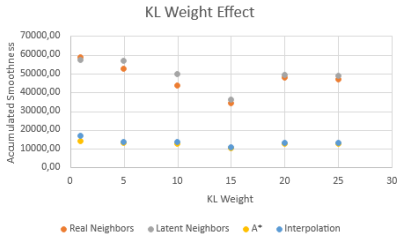

Fig. 23. Effects of KL weight on Smoothness for iCub Robot hand.

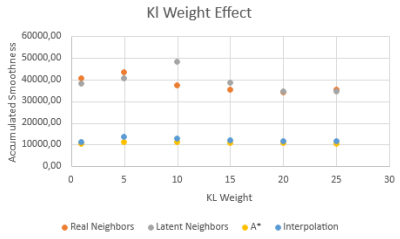

Fig. 24. Effects of KL weight on Smoothness for Shadow Robot hand.

With the reconstruction error assessed, the next step is to examine the effects of the KL weight on the pathing algorithms, as shown in figures 23 and 24. The KL weight indeed influences the results of the model using Metric 3 in terms of Accumulated Smoothness. Notably, the effect is most evident in how the Naive Neighbors approach behaves with increasing weight. Depending on the weight value, the performance can shift between Real Neighbors and Latent Neighbors as the preferred approach. Additionally, it is worth mentioning that with a KL weight of 15 and the iCub hand, the interpolation algorithm provides a smoother path for $R^{d_z} = R^2$ compared to the previous best result in figure 19, although the difference is not substantial. Conversely, the weight does not have a pronounced impact on the accumulated smoothness of the models with the Shadow hand compared to previous attempts.

G. Simulations on NVIDIA Isaac Gym Environment

The simulations in the Isaac Gym environment aim to validate the effectiveness of the models in performing regrasping tasks practically, taking into account the dynamics of contact. The task is considered successful if the object maintains contact with the simulated hand throughout the process. If the object loses contact with the hand at any point, the regrasping task is considered a failure.

Model	Successful Grasp	Failed Grasp
Original	29	1
Metric 1	23	7
Metric 2	28	2
Metric 3	29	1

TABLE I
SIMULATION RESULTS WITH SHADOW ROBOT HAND IN ISAAC GYM ENVIRONMENT

The results indicate that while Metric 1 achieved the best accumulated smoothness value for the Shadow hand in figure 20, it faced challenges with regrasping tasks. In particular, it struggled with applying correct movements to the thumb actuators, leading to a higher failure rate, especially when dealing with objects having a spherical body.

The original model demonstrated reliability in performing regrasps with various objects, serving as a valuable reference for desired results. Metric 2 experienced one more failed regrasp compared to the original model or Metric 3, specifically when attempting a regrasp using a small cylinder held on its side. Metric 3 showed a performance equivalent to the original model but encountered failures in different pairs. While the original model failed in regrasping a small box, Metric 3 faced difficulties in a regrasp involving a small cylinder placed on its side, particularly in the initial grasping phase.

These findings highlight the practicality and adaptability of the models in handling regrasping tasks in complex environments.

VII. CONCLUSION

The metrics developed in this work are designed to align with specific goals, making them adaptable based on the desired outcomes for dimensionality reduction. In the context of regrasping tasks, where the object to be grasped remains unchanged, the optimal metric separates the latent space representation of the initial dataset into clusters, each corresponding to objects of the same type. As observed, Metric 1 stands out for its ability to create smooth regrasping paths. It's crucial to emphasize that fine-tuning all the model parameters is essential since slight adjustments can lead to varying results, ultimately determining the effectiveness of a metric.

The fact that the interpolation algorithm yields results close to the specially implemented smoothest path discovery algorithm is a positive outcome. It indicates that the models are proficient in generating new, unknown grasps for diverse regrasping tasks. These new grasps are well-represented when decoded by the model, enhancing adaptability.

The incorporation of Kullback-Leibler divergence within the loss function, aimed at preserving the distribution of points from the original real space in the latent space, has a noticeable effect on our experiments. Of particular note is the weight assigned to this parameter, which serves as a pivotal determinant influencing the performance of various metrics.

Notably, this weight impacts metrics designed to prioritize the accurate capture of geometric relationships among points in the original real space over clustering points in the latent space.

The simulations in the NVIDIA Isaac Gym environment served as a practical validation of the obtained results. They revealed that Metric 1, while appearing strong initially, has limitations when applied to real regrasps. In contrast, Metric 3 achieved a level of success comparable to the original model while delivering better accumulated smoothness results, when using an increased KL weight. This demonstrates that, with the right metric implementation, it is possible to create a smoother latent space, ensuring that this latent space can effectively control robotic dexterous hands.

This thesis builds upon an existing model, as presented in [3], by introducing innovative methods to create a smoother latent space representation. The primary aim is to facilitate the recreation of similar tasks with minimal changes in the joint angles of the robot hands employed. All simulations in this work assume a static object during regrasping tasks. The logical progression from these simulations involves the execution of analogous tasks with physical iCub and/or Shadow robot hands, thereby validating the virtual results in a real-world setting.

Future work should also explore the adaptability of the metrics introduced in this research or the development of novel metrics to ensure efficient regrasping across various dexterous robot hands. While current in-hand regrasping tasks entail grasping a single object, the future may introduce scenarios where robot hands must efficiently manipulate different objects in a continuous motion. The pursuit of a smoother latent space distribution constitutes a foundational step toward achieving this capability.

ACKNOWLEDGMENT

The author would like to thank the members at ISR for providing the access to the materials and data used for this work.

REFERENCES

[1] M. Santello, G. Baud-Bovy, and H. Jörntell, "Neural bases of hand synergies," *Frontiers in Computational Neuroscience*, 2013.

[2] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *Journal of Neuroscience*, vol. 18, 1998.

[3] D. Dimou, J. Santos-Victor, and P. Moreno, "Learning conditional postural synergies for dexterous hands: A generative approach based on variational auto-encoders and conditioned on object size and category," vol. 2021-May, 2021.

[4] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dimensionality reduction for hand-independent dexterous robotic grasping," 2007.

[5] ——, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," 2007.

[6] A. Bernardino, M. Henriques, N. Hendrich, and J. Zhang, "Precision grasp synergies for dexterous robotic hands," 2013.

[7] O. C. Jenkins, "2d subspaces for sparse control of high-dof robots," 2006.

[8] a Tsoli and O. Jenkins, "2d subspaces for user-driven robot grasping," *Robotics, Science and Systems Conference: Workshop on Robot Manipulation*, vol. 5, 2007.

[9] J. Romero, T. Feix, C. H. Ek, H. Kjellström, and D. Kragic, "Extracting postural synergies for robotic grasping," *IEEE Transactions on Robotics*, vol. 29, 2013.

[10] K. Xu, H. Liu, Y. Du, and X. Zhu, "A comparative study for postural synergy synthesis using linear and nonlinear methods," *International Journal of Humanoid Robotics*, vol. 13, 2016.

[11] N. D. Lawrence and J. Quiñonero-Candela, "Local distance preservation in the gp-lvm through back constraints," vol. 148, 2006.

[12] J. Starke, C. Eichmann, S. Ottenhaus, and T. Asfour, "Synergy-based, data-driven generation of object-specific grasps for anthropomorphic hands," vol. 2018-November, 2019.

[13] ——, "Human-inspired representation of object-specific grasps for anthropomorphic hands," *International Journal of Humanoid Robotics*, vol. 17, 2020.

[14] K. Sohn, X. Yan, and H. Lee, "Learning structured output representation using deep conditional generative models," vol. 2015-January, 2015.

[15] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.

[16] M. P. do Carmo, *Riemannian Geometry*, 1992.

[17] G. Arvanitidis, L. K. Hansen, and S. Hauberg, "A locally adaptive normal distribution," 2016.

[18] G. Arvanitidis, S. Hauberg, and B. Schölkopf, "Geometrically enriched latent spaces," vol. 130, 2021.

[19] N. Chen, M. Karl, and P. V. D. Smagt, "Dynamic movement primitives in latent space of time-dependent variational autoencoders," 2016.