



**HAL**  
open science

# Learning to model the grasp space of an underactuated robot gripper using variational autoencoder

Clément Rolinat, Mathieu Grossard, Saifeddine Aloui, Christelle Godin

## ► To cite this version:

Clément Rolinat, Mathieu Grossard, Saifeddine Aloui, Christelle Godin. Learning to model the grasp space of an underactuated robot gripper using variational autoencoder. IFAC-PapersOnLine, 2021, 19th IFAC Symposium on System Identification SYSID 2021, 54 (7), pp.523-528. 10.1016/j.ifacol.2021.08.413 . cea-03347668

**HAL Id: cea-03347668**

**<https://cea.hal.science/cea-03347668v1>**

Submitted on 17 Sep 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

# Learning to Model the Grasp Space of an Underactuated Robot Gripper Using Variational Autoencoder

Clément Rolinat, Mathieu Grossard,\*  
Saifeddine Aloui, Christelle Godin\*\*

\* *Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France*  
(e-mail: {clement.rolinat, mathieu.grossard}@cea.fr).

\*\* *Université Grenoble Alpes, CEA, Leti, F-38000, Grenoble, France*  
(e-mail: {saifeddine.aloui, christelle.godin}@cea.fr)

---

**Abstract:** Grasp planning and most specifically the grasp space exploration is still an open issue in robotics. This article presents a data-driven oriented methodology to model the grasp space of a multi-fingered adaptive gripper for known objects. This method relies on a limited dataset of manually specified expert grasps, and uses variational autoencoder to learn grasp intrinsic features in a compact way from a computational point of view. The learnt model can then be used to generate new non-learnt gripper configurations to explore the grasp space.

*Keywords:* multi-fingered gripper, grasp space exploration, variational autoencoder

---

## 1. INTRODUCTION

Grasping is fundamental in most of the industrial manufacturing processes such as pick-and-place, assembly or bin picking tasks. The grasp planning question is still an active research topic. It aims at finding a gripper configuration that allows to grasp an object reliably. From a geometrical point of view, the chosen grasp configuration needs to be kinematically reachable and collision-free with respect to the environment, while, from a dynamics point of view, the grasp needs to ensure object stability and resistance against external perturbations. Finding such a grasp configuration requires to explore the grasp space, that is the subset of gripper configurations that effectively grasp the object. Thus, grasp planning is both object dependent and robot hardware dependent. Taking into account those constraints during the exploration is not straightforward, as objects can have sophisticated shapes, and gripper-arm combination can have complex kinematics.

This is even more true for underactuated or compliant multi-fingered gripper, for which the adaptive underactuated system generates object-dependent grasp configurations. This type of grippers are often chosen for grasping tasks (Townsend, 2000). Indeed, such architecture allows to reduce the controller complexity by reducing the number of controlled degrees of freedom, while retaining sufficient kinematic abilities. Moreover, it tends toward producing robust grasps by their mechanical structure.

The grasp planner should be able to find in the high dimensional and highly constrained grasp space a configuration that fulfills a given criterion. There are two main ways to achieve this: analytic approaches and data-driven approaches (Sahbani et al., 2012). Analytic approaches rely on an analytic description of the grasping problem (see Berenson et al. (2007), Roa et al. (2008), Xue et al.

(2007)). Data-driven approaches depend on machine learning methods to predict grasps from object depth map or point cloud (see Zhao et al. (2020), Pinto and Gupta (2016), Depierre et al. (2018), Levine et al. (2018), Mahler et al. (2017)).

A shared issue is the grasp dataset creation, that is the grasp space exploration. A variety of high quality grasps needs to be discovered by exploring the space of possible grasp configurations. There are two main approaches regarding this exploration (Xue et al., 2007): contact point approaches, and gripper configuration approaches. In the first case the grasp space exploration comes down to test various combinations of contact point locations on the object surface. However there is no guarantee that a given combination is a priori kinematically admissible for a given gripper, and the inverse kinematics can even be intractable for underactuated or adaptive grippers. In the second case, the grasp space is explored by testing several gripper spatial configurations. This is more suited for underactuated grippers. Nevertheless, there is no assurance that a given gripper configuration is a priori in contact with the object without realizing extensive simulation trials beforehand.

To circumvent the dimensionality issue related to the huge size of the grasp space, numerous contact point approaches limit their search to fingertip contacts (see Roa et al. (2008), Zhao et al. (2020)), and gripper configuration approaches often use a bi-digital gripper and limit their search to planar grasps (see Pinto and Gupta (2016), Depierre et al. (2018), Levine et al. (2018), Mahler et al. (2017)). For more complex grippers, a human input is often required. For example, in Santina et al. (2019), the authors identified a set of ten grasp primitives from human examples, and reduced the grasp space to those primitives only. In Choi et al. (2018), the authors proposed to limit the search space by discretizing it.

This article presents a method to model the grasp space of a multi-fingered and underactuated gripper for known objects using a variational autoencoder. It relies on a limited set of human-chosen primitive gripper configurations. This model can then be used to generate new gripper configurations that are likely to belong to the grasp space. This allows to explore the grasp space considering inspiration from already specified human-based complex grasps strategies.

The section 2 is dedicated to the problem statement and a presentation of the framework useful for our work. Then, in section 3 the method used to model the grasp space is described. Finally, in section 4 we show how to tune the variational autoencoder hyperparameters to learn the best grasp space model. To conclude, this work is discussed and the planned future works are presented.

## 2. PROBLEM STATEMENT & FRAMEWORK

### 2.1 Simulation Setup

The three-fingered gripper considered in the following has an underactuated and adaptive behavior that allows its natural adaptation to the object geometry, without the need to carefully control each joint, thus increasing the robustness of the grasp.

This gripper has two joints on each finger and one actuator per finger to control both joints. The second (distal) phalanx starts moving when the applied effort on the finger is above a given force threshold. A fourth actuator allows to control the spread angle  $\theta$  between two fingers (see Fig. 1).

This gripper is mounted as end effector of a six degrees of freedom industrial robot arm.

The simulation setup described above is implemented with Gazebo simulator (Koenig and Howard, 2004). A picture of this simulated setup is displayed in Fig. 1.

### 2.2 Problem Statement

An object is placed on a table in the workspace of the considered robotic setup. It is assumed that the object geometry is known, as well as the pose in the scene of its associated frame  $F_{obj}$  thanks to exteroceptive vision system such as in Drost et al. (2010) for example.

In the following, a grasp configuration is a gripper configuration that is able to grasp the object without colliding with the table. It is defined as follows by eight parameters:

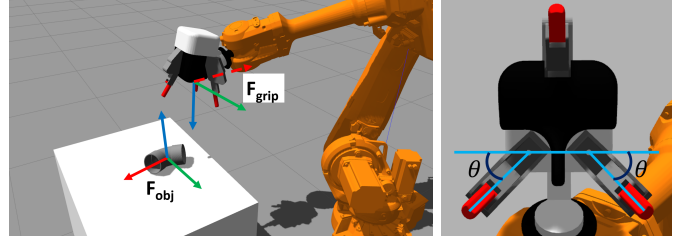
- the pose of the gripper frame  $F_{grip}$ ,  

$$(x, y, z, q_x, q_y, q_z, q_w) \in \mathbb{R}^3 \times SO(3)$$
with the orientation expressed in quaternion convention;
- the spread angle  $\theta$ , as shown in Fig. 1.

The dimensionality of this configuration space is high (seven dimensions), but this allows to fully leverage the grasping ability and kinematic potential of the gripper. Thus, the grasp space is a subset of this gripper configuration space, with an additional constraint that every gripper configuration is able to grasp the object without colliding

with the table. The goal is to model and explore this grasp space.

To locate the gripper, a dedicated frame  $F_{grip}$  situated between the fingers in front of the palm is used. This frame is displayed in Fig. 1. Gripper poses are expressed relatively to the object frame  $F_{obj}$ , in order to be invariant to object poses.



(a) Pose of the frame  $F_{grip}$  used to locate the end-effector relatively to  $F_{obj}$ . (b) Spread angle  $\theta$ .

Fig. 1. Gripper frame and spread angle.

### 2.3 Variational Autoencoders

Variational autoencoders (VAE) are a derivative of classic autoencoders. In addition to learning a compressed representation of the training data as a classic autoencoder, a VAE allows to generate consistent data from its latent space reliably.

The goal of a VAE is to infer the latent variables that are behind the training dataset, so that a latent space with fewer dimensions than the original space can be created. Sampling in it will generate a new data distribution that resembles to the training dataset one. In this work, this allows to create a model of the grasp space, from which new gripper configurations can be generated.

In a VAE, among other features, a supplementary term is added in the loss function: the Kullback-Leibler (KL) divergence (Kingma and Welling, 2014). This term helps the data to be represented as a normal distribution in its latent space, and thus regularizing it.

## 3. METHOD DESCRIPTION

The proposed method has two main steps : an object depend primitive dataset building step, and a Human-initiated Grasp Generator VAE (HGG) training step.

### 3.1 Primitive Grasp Dataset

To leverage the human ability to find gripper configurations belonging to the grasp space, an object dependent primitive grasp dataset is built. A primitive grasp is a handcrafted gripper configuration, with its pose and spread angle human-chosen so that it is collision free and likely to grasp the object. The spread angle is chosen between four discrete values corresponding to main gripper internal layouts:  $\theta = 0$ ,  $\theta = \pi/6$ ,  $\theta = \pi/4$ , and  $\theta = \pi/2$ .

In this work, such primitives are gathered on three different objects:

- a connector bent pipe

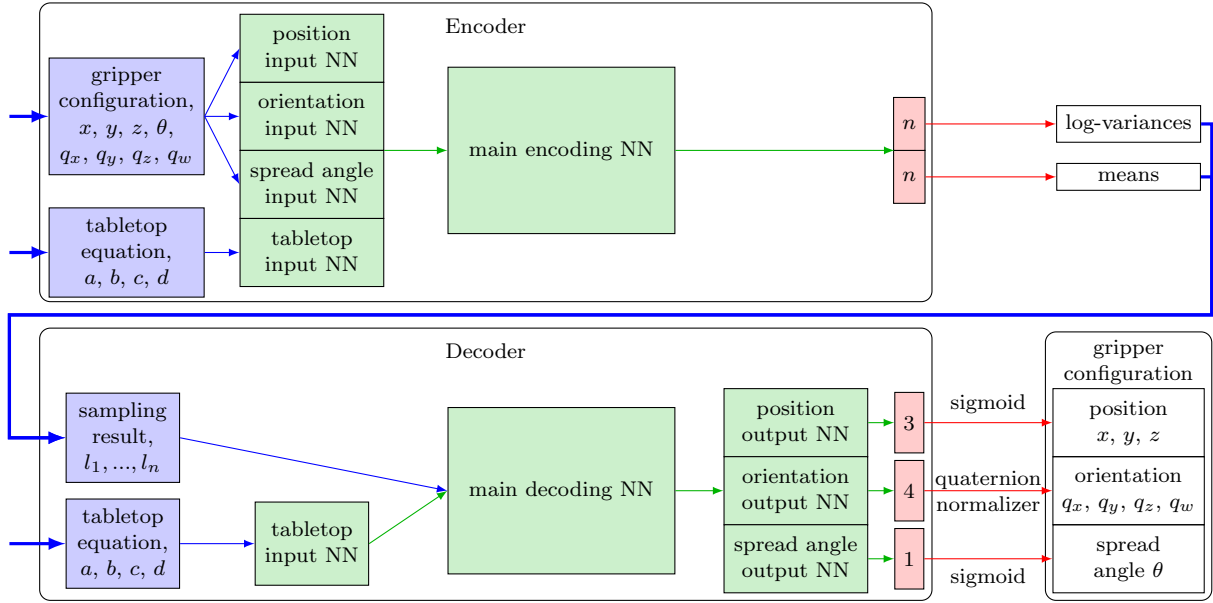


Fig. 2. HGG architecture. In blue the input layers, in green the hidden neural networks (NN) and in red the output layers. The hidden NN inner layers are fully connected layers, with hyperbolic tangent activation function. The latent space dimension, that is the number of latent variables, is  $n$ . The main encoding and main decoding NN have symmetrical inner architecture. The supplementary input for the tabletop equation ensures that the generated grasp depends on it (Sohn et al., 2015). This architecture is implemented with Tensorflow (Abadi et al., 2015) and Keras (Chollet et al., 2015) python libraries.

- a pulley
- a small cinder block

Their 3D meshes used in the simulation in their different stable positions are visible on Fig. 3. Those objects were chosen for their relative complexity and diversity in terms of shapes.

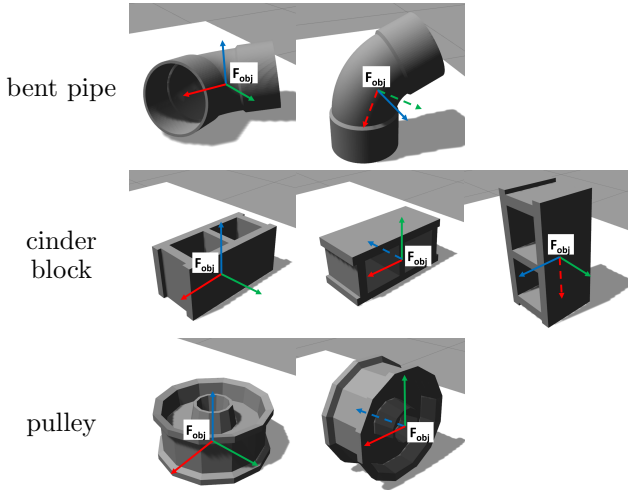


Fig. 3. The chosen objects and their frame  $F_{obj}$  in their different stable positions.

A set of primitive gripper configurations is determined for each of those objects for each of their stable position. These primitive gripper configurations can be sorted in different grasp types presented on Fig. 4. For each of these grasp type, several variants are manually created.

For each object is gathered the following number of primitive grasps:

- bent pipe: 145 samples
- cinder block: 141 samples
- pulley: 118 samples

Around one hour is needed for a human operator to register the primitives for a given object.

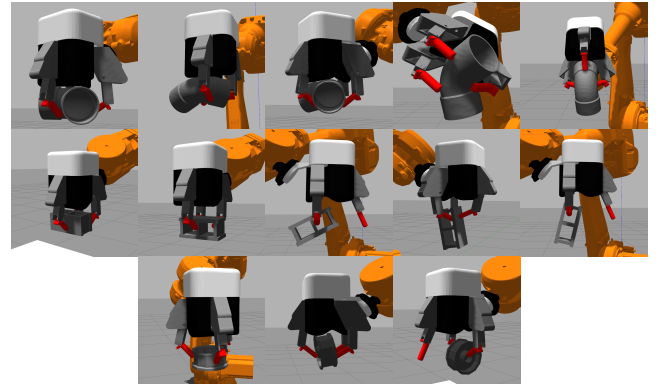


Fig. 4. Primitive grasp types for the three chosen objects. On the first row the grasp types for the bent pipe, on the second row for the cinder block, and on the third row for the pulley

The dataset stores the eight parameters describing each primitive grasp along with the four parameters of the tabletop plane Cartesian equation in object frame  $F_{obj}$ . Indeed, many objects have different possible stable positions on the table. This is a critical information to avoid collisions with it. Some grasps may collide with the table in a given stable position, while being suitable for another stable position.

Expressing the grasp configuration in the object frame is still useful as it allows an invariance to a position change and to a rotation around a vertical axis.

### 3.2 Human-initiated Grasp Generator VAE (HGG)

The goal of the Human-initiated Grasp Generator VAE (HGG) is to infer the correlations existing between the parameters of different grasp primitives to learn a model of the grasp space. Such correlations exist, as primitive grasps are in the grasp space, and this space is a subset of the gripper configuration space. The HGG is able to use those correlations to map the grasp space in its latent space. This model can be used to generate efficiently new configurations that are likely to be in the grasp space.

A distinct HGG is trained on the primitive grasp dataset for each object. Its inputs and outputs are shown in Fig. 2 along with its global architecture. Before the training, the inputs and outputs data are normalized. This allows a faster training as the network does not have to scale its data by itself. For the gradient descent during the training, a Mean Square Error (MSE) is computed for each gripper parameter. Each of these errors is averaged on each batch. The global loss is computed as the sum of these averaged errors together with the KL divergence loss.

To make sure that the quaternion outputs by the decoder is a unit one, a custom activation function is used to normalize it on the output layer of the decoder.

### 3.3 Latent Space Produced with Two Latent Variables

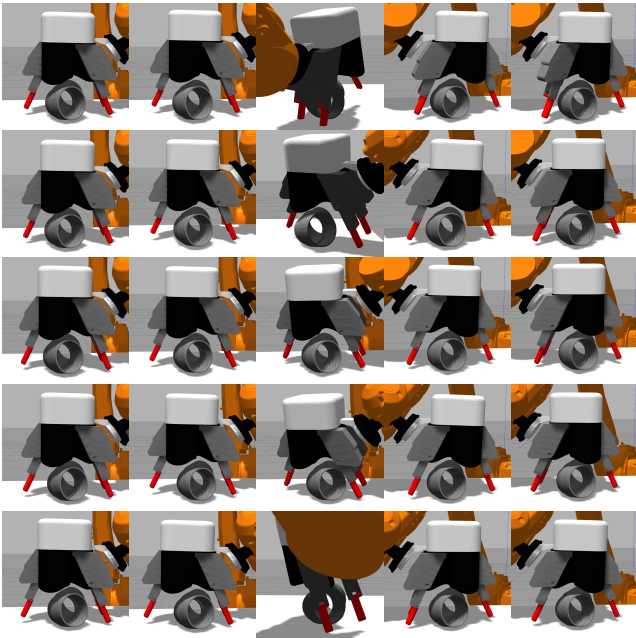


Fig. 5. Gripper configurations generated when visiting a two dimensional latent space. The central image is the point  $(0, 0)$  in latent space. The inner and outer image rings around it correspond to points evenly distributed on circles of diameter respectively 0.5 and 1 in latent space. Here, translations along the image plan normal are not visible, which explains some visually almost identical configurations.

The HGG learns to model the grasp space in its latent space. By sampling values in it, one can generate new gripper configurations that are likely to belong to the grasp space. On Fig. 5 is shown the obtained gripper configurations when visiting a two dimensional latent space for one stable position of the bent pipe. As some configurations may not lead to successful grasps, or may be in collision with the object or the environment, only pre-grasp configuration are shown (that is before closing the fingers), with collisions disabled.

For this stable position, it appears that the latent variable displayed on the horizontal axis on Fig 5 encodes mainly the direction from which the bent pipe will be grasped: the two rearrangeable fingers on the concave side or on the convex side. The other latent variable seems to encode mainly translations. On the top-right corner appears a configuration corresponding to the first bent pipe primitive grasp type (shown on Fig. 4). The rest of the right side configurations correspond to the third grasp type, and the left side configurations to the second grasp type.

## 4. HGG TUNING TO MODEL EFFICIENTLY THE GRASP SPACE

The HGG has three main hyperparameters that can be tuned to improve the learnt grasp space model:

- the network size;
- the latent space dimension;
- the KL divergence loss component coefficient (Higgins et al., 2017).

Several indicators can be monitored to assess the effect of those hyperparameters on the performances of the HGG:

- the reconstruction error;
- the KL divergence loss component value;
- the number of used latent variables, that is the number of latent variables with a high KL divergence;
- the share of generated successful grasps.

Various learning trials were conducted with different hyperparameters combinations. A summary of the effects of the hyperparameters is given in the following subsections.

### 4.1 Trade-Off Between Reconstruction and Regularity

One of the distinctive features of a VAE is that its loss function combines a reconstruction cost and a regularization cost, the KL divergence cost. This leads the training process to converge to a trade-off between reconstruction and regularity. The reconstruction is the ability to accurately reproduce on the output the input data. The regularity is the fact that the input data are homogeneously distributed in each latent variable (here, following a normal distribution), and that latent variables are disentangled. A side effect of those constraints is that the network is pushed to use as few latent variables as possible to represent the data. For the HGG, both terms are important: a good reconstruction is needed as it allows to capture faithfully all the primitive data variability, and a good regularity is also needed as it reduces the data distribution sparsity, and thus the risk of generating inconsistent configurations.

In Higgins et al. (2017), the authors introduced a coefficient on the KL divergence term that allows to adjust this trade-off. Increasing this coefficient will put higher priority on the KL divergence term, and thus increases the regularity at the expense of the reconstruction. A too high coefficient on this term can push the network to ignore the data variability along a given axis to homogenize the data in its latent variables and disentangle them, leading to poor reconstruction. Conversely, a too low coefficient will allow a very accurate reconstruction, but the latent variables will be more entangled and the data distribution in them will be sparser.

The optimal value of the coefficient depends mainly on the latent space dimension. In Higgins et al. (2017), they recommend a value greater than 1, but they use the VAE for image generation, which involve generally a latent space of greater dimension than for the presented use case. For the HGG, a value below 1 is mandatory to keep an acceptable reconstruction loss.

#### 4.2 Network Size

To increase reconstruction with less impact on regularity than the KL coefficient, one can increase the network size, that is the number of neurons in the different layers, or the number of layers. Indeed, it increases the number of network trainable parameters, and thus the complexity of the functions that it is able to approximate. However, it increases the computational cost of both the learning phase and the inference phase, and the memory required to store model. Moreover, the more parameters the network has, the more training data it needs for a proper training. For the HGG use case, there are very few training data, which limit the size of the network.

For the architecture presented in Fig. 2, a good trade-off between computation cost and general performances is around 30 000 parameters for the whole VAE, that is both encoder and decoder. Indeed, the reconstruction starts to improve less significantly beyond this threshold.

#### 4.3 Grasp Space Dimension & Latent Space Dimension

The grasp space is a subset of the gripper configuration space (see subsection 2.2). Thus, it has at most 7 dimensions, but its true size is a priori unknown. As the goal is to map the grasp space in the HGG latent space, it is important that the number of latent variables used by the HGG among the available ones is at least equal to the grasp space dimension. Otherwise, there will be information loss due to the compression caused by the projection of the grasp space into a smaller space. Although conservative, it is sub-optimal to let the HGG find by itself the required number of latent variables needed to map the grasp space, by letting the latent space dimension be equal to the one of the gripper configuration space. Indeed, even if the KL divergence term in the cost function will push the network to use as few latent variables as possible, increasing the number of available latent variables increases the chances to converge toward a cost function local minimum where the network uses more latent variable than needed. As the used latent space dimension is greater, the data distribution inside it is also sparser, which have the same effect as a too low coefficient on the KL divergence.

Thus, it is useful to know an approximation of the dimension of the grasp space. Here, a dimensional analysis tool has been chosen, the kernel-PCA (Scholkopf et al., 1999). Kernel-PCA is an extension of the PCA to non-linear relations. Indeed, the grasp space is probably a submanifold of the gripper configuration space, which probably involves non-linear relations between the parameters of this space.

The kernel-PCA implemented in scikit-learn is used (Pedregosa et al., 2011). The algorithm is run for each object, taking as input the list of gripper configurations in the primitive grasp dataset. The grasp space dimension is determined as the number of eigenvectors of the centered kernel matrix needed to retrieve 90% of the information, by looking at their eigenvalues. Indeed, it means that the kernel-PCA can explain 90% of the data variability with the given number of eigenvectors. The outputted result depend on the object: 3 dimensions for the pulley, and 4 dimensions for the bent pipe and the cinder block.

Therefore, this can serve as an upper bound for the optimal latent space dimension. Indeed, the HGG has a supplementary information: the tabletop Cartesian equation, and may use it to learn a more compact representation than the one found by the kernel-PCA.

#### 4.4 Overview

In table 1 is summarized the influence of the three hyperparameters on the chosen indicators. For this evaluation, trials were conducted with hyperparameter combinations among the following ranges:

- network size between 12 000 and 30 000 parameters;
- latent space dimension between 2 and 6;
- KL divergence coefficient between 0.0002 and 0.01.

Table 1. Spearman correlation coefficients between the hyperparameters and the indicators.

|                                                 | latent<br>space<br>dimension | KL<br>divergence<br>coefficient | network<br>size |
|-------------------------------------------------|------------------------------|---------------------------------|-----------------|
| number of used<br>latent variables              | 0.12                         | -0.26                           | 0.55            |
| reconstruction error                            | -0.18                        | 0.18                            | -0.21           |
| KL divergence                                   | -0.75                        | -0.62                           | 0.09            |
| proportion of<br>generated<br>successful grasps | -0.39                        | 0.25                            | 0.39            |

Table 2. performances of the HGG for the selected hyperparameters: 30 000 network parameters, 3 latent variables (that are all used), and a KL divergence coefficient of 0.0005. The mean errors are measured on the training data.

|              | mean<br>position<br>error (m) | mean<br>orientation<br>error (degree) | generated<br>successful grasps<br>share (%) |
|--------------|-------------------------------|---------------------------------------|---------------------------------------------|
| bent pipe    | 0.004                         | 1.94                                  | 68.2                                        |
| pulley       | 0.005                         | 1.32                                  | 84.2                                        |
| cinder block | 0.009                         | 1.1                                   | 93.5                                        |

In table 2 are shown the performances corresponding to the set of hyperparameters achieving the best trade-off

between the reconstruction and the proportion of generated successful grasps. To avoid arm kinematic reachability issues, as gripper configurations are in object frame, each generated configuration is tested for different object orientations relative to the robot. The main cases of failing grasps are found when transitioning between different grasp types, and with the fifth grasp type of the bent pipe (Fig. 4, top right) where one of the bottom finger can collide with the table in the pre-grasp phase for some gripper orientation variations.

## 5. CONCLUSION

This work presents a method to model the grasp space of an underactuated gripper. It generates new gripper configurations that are likely to belong to the grasp space, which allows to explore it. Some insights for a proper hyperparameters tuning are also given.

Various tracks can be investigated in future works. First, a reduction of the number of human inputs required per object would be useful to scale this method to several objects. Moreover, this work was conducted in simulation only, and trials on a real setup should be conducted. Finally, the presented method does not take into account any criterion for the grasp quality. This method and the presented proper hyperparameter tuning can be used to improve other grasp space exploration procedures, which use a grasp quality metric, such as (Rolinat et al., 2021).

## REFERENCES

- Abadi, M. et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from [www.tensorflow.org](http://www.tensorflow.org).
- Berenson, D., Diankov, R., Nishiwaki, K., Kagami, S., and Kuffner, J. (2007). Grasp planning in complex scenes. In *IEEE-RAS International Conference on Humanoid Robots*, 42–48.
- Choi, C., Schwarting, W., DelPreto, J., and Rus, D. (2018). Learning object grasping for soft robot hands. *IEEE Robotics and Automation Letters*, 3(3), 2370–2377. doi:10.1109/LRA.2018.2810544.
- Chollet, F. et al. (2015). Keras. Software available from [www.keras.io](http://www.keras.io).
- Depierre, A., Dellandréa, E., and Chen, L. (2018). Jacquard: A large scale dataset for robotic grasp detection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3511–3516. doi:10.1109/IROS.2018.8593950.
- Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 998–1005.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representation*.
- Kingma, D. and Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.
- Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, 2149–2154. doi:10.1109/IROS.2004.1389727.
- Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International Journal of Robotics Research*, 37(4), 421–436. doi:10.1177/0278364917710318.
- Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., and Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. In *Robotics: Science and Systems (RSS)*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3406–3413.
- Roa, M.A., Suarez, R., and Rosell, J. (2008). Grasp space generation using sampling and computation of independent regions. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2258–2263.
- Rolinat, C., Grossard, M., Aloui, S., and Godin, C. (2021). Human-initiated grasp space exploration algorithm for an underactuated robot gripper using variational autoencoder. In *IEEE International Conference on Robotics and Automation (ICRA)*, in press.
- Sahbani, A., El-Khoury, S., and Bidaud, P. (2012). An overview of 3d object grasp synthesis algorithms. *Robotics and Autonomous Systems*, 60(3), 326–336.
- Santina, C.D., Arapi, V., Averta, G., Damiani, F., Fiore, G., Settini, A., Catalano, M.G., Bacciu, D., Bicchi, A., and Bianchi, M. (2019). Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands. *IEEE Robotics and Automation Letters*, 4(2), 1533–1540. doi:10.1109/LRA.2019.2896485.
- Scholkopf, B., Smola, A., and Müller, K.R. (1999). Kernel principal component analysis. In *Advances in Kernel Methods - Support Vector Learning*, 327–352. MIT Press.
- Sohn, K., Yan, X., and Lee, H. (2015). Learning structured output representation using deep conditional generative models. In *International Conference on Neural Information Processing Systems*, volume 2 of *NIPS’15*, 3483–3491. MIT Press, Cambridge, MA, USA.
- Townsend, W. (2000). The BarrettHand grasper – programmably flexible part handling and assembly. *Industrial Robot*, 27(3), 181–188. doi:10.1108/01439910010371597.
- Xue, Z., Zoellner, J.M., and Dillmann, R. (2007). Grasp planning: Find the contact points. In *IEEE International Conference on Robotics and Biomimetics (RO-BIO)*, 835–840.
- Zhao, Z., Shang, W., He, H., and Li, Z. (2020). Grasp prediction and evaluation of multi-fingered dexterous hands using deep learning. *Robotics and Autonomous Systems*, 129, 103550.