Task level disentanglement learning in robotics using βVAE

Original Scientific Paper

Midhun M S
Department of Electronics,
Cochin University of Science and Technology, Kerala, India
midhunms@cusat.ac.in

James Kurian
Department of Electronics,
Cochin University of Science and Technology, Kerala, India
james@cusat.ac.in

Abstract *– Humans observe and infer things in a disentanglement way. Instead of remembering all pixel by pixel, learn things with factors like shape, scale, colour etc. Robot task learning is an open problem in the field of robotics. The task planning in the robot workspace with many constraints makes it even more challenging. In this work, a disentanglement learning of robot tasks with Convolutional Variational Autoencoder is learned, effectively capturing the underlying variations in the data. A robot dataset for disentanglement evaluation is generated with the Selective Compliance Assembly Robot Arm. The disentanglement score of the proposed model is increased to 0.206 with a robot path position accuracy of 0.055, while the state-of-the-art model (VAE) score was 0.015, and the corresponding path position accuracy is 0.053. The proposed algorithm is developed in Python and validated on the simulated robot model in Gazebo interfaced with Robot Operating System.*

Keywords*: Machine Learning, Robotics, Neural Networks, Variational Autoencoder, beta-VAE*

1. INTRODUCTION

With the emergence of Artificial Intelligence (AI), trajectory planning in robotics has been solved for many scenarios with different methods [1]. However, task planning with generative models is still because of the complex nature of the joint trajectory, joint constraints, self-collision and collision with the workspace objects. Numerous problems in robotics are solved based on reinforcement learning, established in real-time feedback from sensors. Serial manipulator robots possess multiple joints and links where each joint is controlled by one or many actuators using link actuator signals. Numerous models propose modelling the lower-dimensional joint values with sensory feedback. This approach models are an open-loop higher-dimensional abstraction of the lower-dimensional joint values.

Generative models are best suited for generating data from the same distribution. Generative Adversarial Network (GAN) [2] and Variational Autoencoder (VAE) [3, 4] are the two most common forms of generative deep learning networks. VAEs were picked because their training is more stable than GAN (no mode collapse). VAE has two models, namely encoder and decoder. The encoder maps the input into a higher-dimensional latent space, and the decoder rebuilds it.

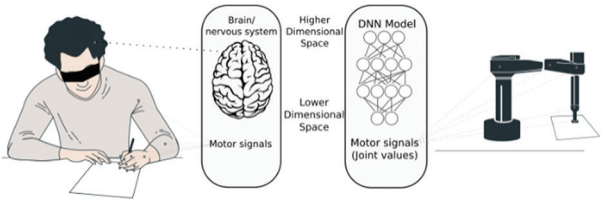

Fig. 1. The Proprioception intelligence model for human and robots. Higher and lower-dimensional space exists in the nervous system and DNN for human and robots. Lower dimensional signals directly control the joints using the motor signals.

Proprioception is the ability of a human to sense position, orientation, joint angle etc. If a robot model has these features? Fig. 1 shows the model of a human drawing an image on paper blindfolded and a robot without any visual feedback - both project the higher dimensional planning to lower-dimensional action.

A change in an independent factor in a higher dimension only affects a single factor in output is called disentanglement [5]. Disentanglement representation of data has been getting more critical in the machine learning community in recent years. The human brain coded each object based on colour, shape, size, etc.;

similarly, if the robot can learn the task's underlying nature, human interpretability, predictive performance and compressed representation will benefit.

In the proposed disentanglement robot model, the encoder generates the mapping function from lower dimensional raw trajectory data to higher dimensional representation, and all the interpretable higher-dimensional vectors generate the mapping function from higher-dimensional representation to lower-dimensional motor signals in the decoder, which encapsulates all the kinematics complexities, sequence, and task information. The main contributions in the work are as follows

- The model generates a generative model for robot task planning
- Human interpretable, disentangled latent space is learned by the model, which is an effective way to make new data from the underlying factors of variations.

The rest of the paper is organised as follows. Section 2 explains related works in the field; Section 3 describes the system implementation for disentanglement representation. Section 4 presents the simulation setup. Section 5 discusses the results obtained, and Section 6 explains the conclusions.

2. RELATED WORK

Disentanglement models generate data from the independent factors of variation. Since *βVAE* [6], disentanglement learning is getting strong community attention. Disentanglement learning is divided into supervised and unsupervised. The supervised method needs the dataset to contain all the factors of variations. Supervised and unsupervised disentanglement models gain much attention in the image, audio and video domains. Most real-world datasets do not have variation factors, so the proposed work implements the unsupervised model.

Disentanglement in robotics usually processes the input image and learns the disentanglement on those. Y. Hristov et al. [7] presented the robot learns from demonstrations from the captured scene. Mobile robot path planning and execution are demonstrated with disentanglement scene representation by V.A.K.T Rajan et al. [8]. Learning the changing surroundings by mobile robots in [9, 10] uses image-based disentanglement. M.Wulfmeier et al. [9] represent an improved reinforcement learning approach for better perception and exploration with the help of disentanglement. J. Pajarinen et al. [11] presented a probabilistic approach to disentangle the objects from an image and waste sorting using the state of the art machine learning algorithms with a robotic arm. M. Zolotas et al. [12] presented a robotic wheelchair that uses a disentangled Variational sequence encoder for trajectory planning and execution with a joystick and laser scanner inputs. Robot disentanglement models produce closed-loop control systems with cameras, sensors, user inputs, or combinations. The proposed implementation uses an open-loop control system, which learns and produces the task without feedback.

3. SYSTEM IMPLEMENTATION

The robot trajectory contains the sequence of moving joint values. Each channel in the data includes a single joint motion and collectively moves to achieve a particular goal. The data is recorded in a simulator/emulator environment and used as the dataset.

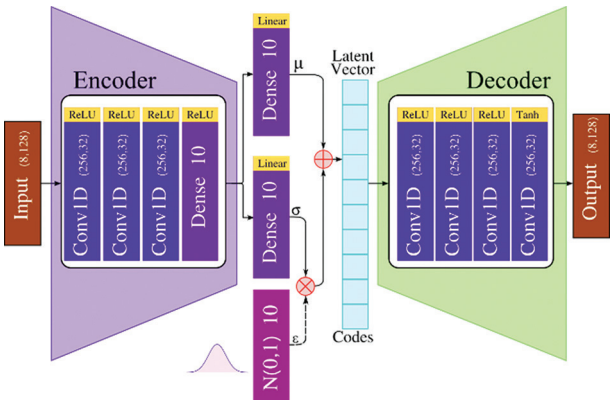

Fig. 2. The Variational Autoencoder model architecture for 1-D robot task sequence data.

3.1. THE PROPOSED NETWORK ARCHITECTURE

Autoencoder (AE) network consists of two networks named encoder and decoder. One dimensional Convolution layer captures the hidden features in the data and generates the model. The architecture is portrayed in Fig. 2. CNN learns features from the raw data while training, sparsely connected layers make it more efficient to learn large networks than the densely connected Multi-Layer Perceptron (MLP). Also, they have low computational requirements and are immune to small changes in translation, scaling and distortion in the input. Hence 1-D CNNs are used for learning the underlying data structures of robot tasks with non-linear Rectifier Linear Unit (ReLU) activation functions ($max(0, x)$). The initial layers of the encoder network learn the simple joint-trajectory features in its kernels, and the higher layers model the complex task level representation.

Table 1. DH parameters of SCARA robot.

Parameter	Link1	Link2	Link3	Link4
Link length (a)	0.45 m	0.45 m	0	0
Link Twist(α)	0	π	0	0
Joint distance(d)	0	0	d_3	0
Joint Angle(θ)	θ_1	θ_2	0	θ_4

The decoder network/generative network uses the transposed convolution layers to model the latent dimensional vector into the robot task data. The probabilistic nature of the model makes it a generative network for robot-task data. The model is trained using a variant Stochastic Gradient Descent optimiser called Adam, which is relatively computationally efficient and less prone to noise.

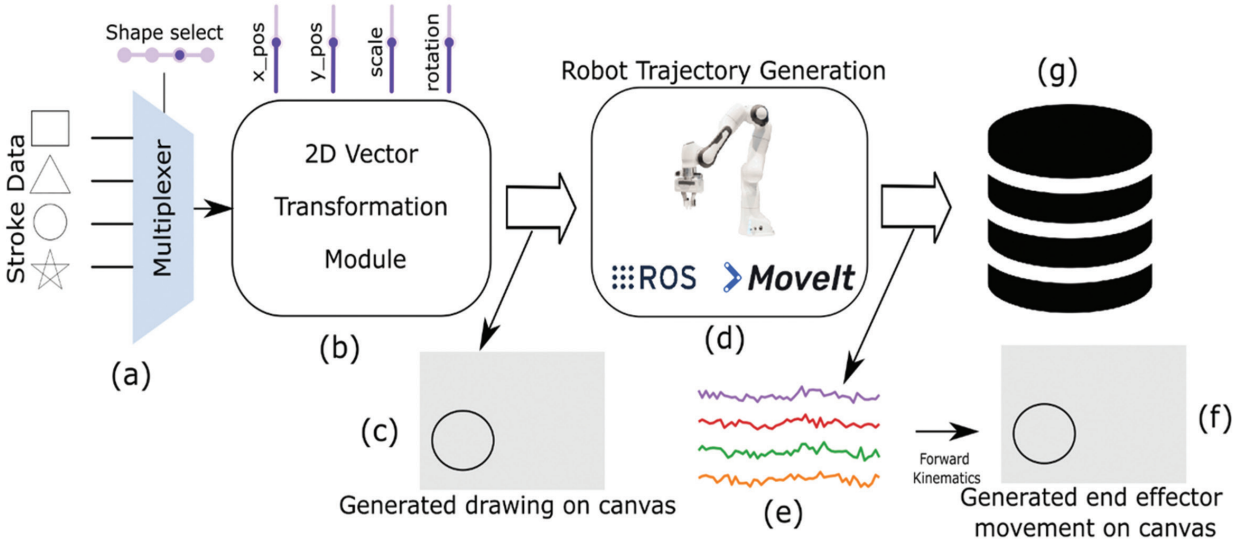

Fig. 3. The disentanglement robot task dataset generation. a) Input path generated using interpolation of points and selected using a multiplexer module, b) 2D vector transformation module with translation, rotation and scaling operations c) The generated trajectory is shown - grey colour represents the canvas d) Robot trajectory generation module e) The generated joint values f) Task space path is plotted by applying forward kinematics to the joint-values g) stored in a dataset.

Table 2. Factors of variation

Factor	Values	Count
Shape	Circle, Square, Triangle, Star	4
Scaling	0.5, 0.6, 0.7, 0.8, 0.9, 1.	6
Orientation	0, 9, 18 … 351	40
Position_x	-70.0, -68.5 … 70	32
Position_y	-70.0, -68.5 … 70	32
Total configurations		**983040**

3.2. SIMULATION SYSTEM DESIGN

Selective Compliance Assembly Robot Arm (SCARA) [13] is a 4-degree of freedom (DOF) serial manipulator robot used in the proposed work. The simulated model of the SCARA robot is developed as a physical linkage system with a Unified Robotics Description Format (URDF) file based on Denavit–Hartenberg (DH) [14] parameters described in Table 1. Robot Operating System (ROS)[15] interfaced with Gazebo simulator with Open Dynamics Engine (ODE) physics engine is used for simulating the model with joint, link, visual and collision parameters in the URDF file. Since the simulator is computationally complex, a low-footprint kinematics model is also developed with the robotics toolbox for Python [16] for evaluating the model performance in the evaluation phase.

Considering $X \in \{x_i\}$ as input and $Z \in \{z_i\}$ as the latent space vector in the network, Evidence Lower Bound (ELBO) [3] in VAE is defined as

$$\log p(X) \geq E[\log p(X|Z)] - D_{KL}[q(Z|X)||p(Z)] \quad (1)$$

where $p(X/Z)$ and $p(Z/X)$ are two probability distributions. The term $E[logp(X/Z)]$ represents the reconstruction and $D_{KL}[q(Z/X)||p(Z)]$ represents the similarity between the two probability distribution. The goal of the network is to maximise the ELBO, i.e., maximise the similarity while keeping the prior and posterior distribution closure as possible.

The model ϕ and θ represent the weights and biases of the probabilistic encoder and decoder, respectively, and its corresponding distribution is defined as $q_\phi(z/x)$ and $p_\theta(x/z)$. The final objective is to minimise the loss as

$$L_{VAE}(\theta, \phi) = -E_{z\, q_\phi(z|x)} \log p_\theta(x|z) + D_{KL}(q_\phi(z|x)||p_\theta(z)) \quad (2)$$

The gradients cannot backpropagate since the sampling operation exists in the model. Re-parameterisation trick is used to model the z as

$$z = \mu + \sigma \odot \epsilon \quad (3)$$

where fully connected layers model the mean (μ) and variance (σ) of the prior representation $p_\theta(z)$ and a sampling layer with a sampling normal vector ($\epsilon \sim N(0,1)$) is utilised. ($\odot$ represents element-wise product) The latent vector is called codes in disentanglement representation. The compressed human interpretable vector is learned - each robotic task generated with varying human interpretable factors like position, orientation, constraints etc.

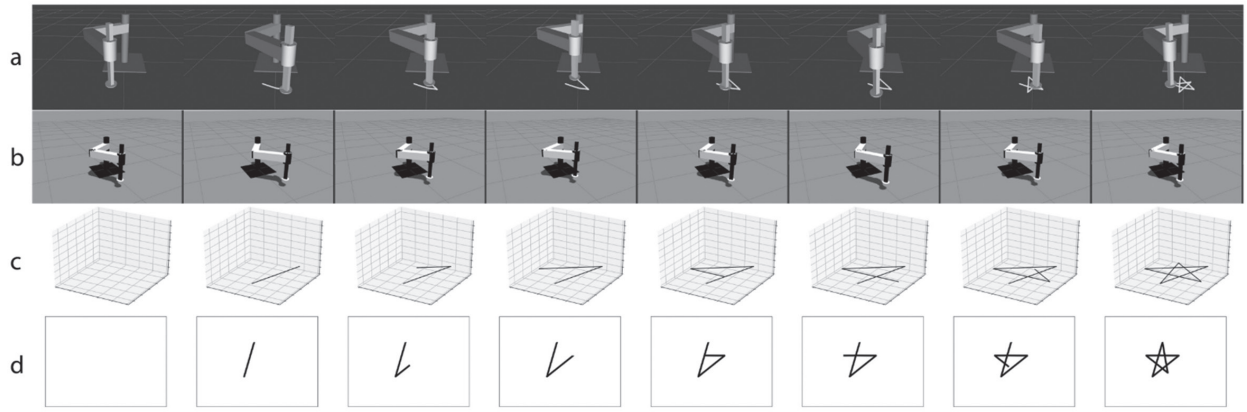

Fig. 4. The disentanglement robot task dataset generation (equidistant samples) of a sample (drawing a star). a) ROS robot visualisation (Rviz) with end-effector movement is shown, b) Gazebo simulated robot, c) The generated trajectory in three-dimensional space is shown d) Generated plot in canvas.

Adding more importance to the KL loss term in equation 2 with a new-hyper-parameter $\beta(> 1)$ will enhance the divergence factor and improves the disentanglement performance. The network is called βVAE, and the new loss is given by,

$$\mathrm{L}_{\beta VAE}(\theta, \phi, \beta) = -\mathrm{E}_{\mathrm{z}\, \mathrm{q}_{\phi}(\mathrm{z}|\mathrm{x})} \log \mathrm{p}_{\theta}(\mathrm{x}|\mathrm{z}) + \beta \mathrm{D}_{\mathrm{KL}}(\mathrm{q}_{\phi}(\mathrm{z}|\mathrm{x})||\mathrm{p}_{\theta}(\mathrm{z})) \quad (4)$$

As the β increases, the representation becomes more suitable, but the reconstruction loss increases, leading to lower precision in robotic tasks, which is not advisable.

4. SIMULATION SETUP

4.1 DATASET

Disentanglement testing Sprites dataset (dSprites) [15] is a popular dataset with images and its underlying factors of variations. The robot version of the dSprites-like dataset is developed using the SCARA robot with a straightforward task - "draw a shape on a canvas", as shown in Fig. 3. Four different shapes were picked - box, circle, triangle and star. Each shape creates multiple instances by varying the independent factors – Position (x, y), scaling (s), and rotation (θ). The transformation is accomplished by using a 2D vector algebra equation.

The dataset preparation is carried out in two steps. The task space trajectory is generated using the transformation metric in the first phase and the generation of the joint space trajectory in the second phase. Four basic shapes are selected in the first phase - circle, square, triangle and star. The vector drawing of each shape is generated using linear algebra equations. Then interpolate the points in the shape and create a sequence in the task space of the robot canvas. Each point is transformed using the equation

$$\begin{bmatrix} x_o \\ y_o \\ z_o \\ s_o \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{offset} \\ 0 & 1 & 0 & y_{offset} \\ 0 & 0 & 1 & z_{offset} \\ 0 & 0 & 0 & s_{canvas} \end{bmatrix} \begin{bmatrix} cos\theta & sin\theta & 0 & x \\ -sin\theta & cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 0 \\ 1 \end{bmatrix} \quad (5)$$

where x, y, θ and s represent independent factors of variations, x_i, y_i represents the position points in the generated trajectory.

The x_{offset}, y_{offset}, z_{offset} and s_{canvas} projects the points into the robot workspace and the points on the robot task space obtained as $(x_o/s_o, y_o/s_o, z_o/s_o)$.

The values represent the 2D representation of the task as plotted on a canvas, as shown in Fig. 4d. Then the values are translated into the robot workspace, which will be the 3D representation shown in Fig. 4c. The robot trajectory points in Robot visualization (Rviz) and gazebo simulator are shown in Fig. 4a and Fig. 4b.

All possible combinations of the factors of variations are generated, as shown in Table 2. The total configurations are estimated as

$$C_{total} = \prod_{i=1}^{5} factor_i \quad (6)$$

The Robot Trajectory generation uses Cartesian planners available in the Moveit planning library [18]. The generated trajectory is post-processed to remove outliers, and the dataset is created for the training.

Each configuration $c_i \in C_{total}$ is taken and generated, the task space path using a 2D vector transformation block with equation 5. The robot Trajectory generation module plans the trajectory and appends it to a dataset. The dataset is normalized based on the corresponding joint limits in the post-processing phase.

The generated task is stored with the corresponding factors and metadata in the database. The data filtering and outlier removal were done by using Python scripts.

4.2 ROBOTICS METRICS

Accuracy and repeatability are the two most common evaluation metrics for robot performance defined in ISO 9283:1998 [19]. Each is calculated by generating n data points. Path position accuracy and repeatability are computed as the maximum pose position accuracy and repeatability value.

4.3 DISENTANGLEMENT METRICS

Each independent element in the labelled data is called a factor, and the varying independent variables in the latent space are called codes. There is no proper way to measure true disentanglement, completeness and informativeness, but the literature suggests many metrics to rely on. βVAE score is one of the first metrics to evaluate the disentanglement's performance, also called the z-min variance score. Later many methods were suggested with different advantages.

βVAE [6] and FactorVAE [20] are some of the initial disentanglement representation methods which measure the variance in codes. Mutual information Gap (MIG) [5] is used to evaluate the disentanglement by using the mutual information between the true underlying factors and generated codes, which uses the difference between the most prominent two variations. *jemmig* is introduced in [21], which measures the modified *MIG score*, including all the factors of variation instead of top2 as in *MIG score*.

Later disentanglement is represented using three terms disentanglement, completeness and informativeness (DCI) [22]. DCI run k linear repressor and evaluates the metrics. Disentanglement represents the amount of disentangling the underlying data variations. Completeness measures the amount of data a single variable captures, and informativeness represents how informative the latent vector is. Disentanglement library functions are used for evaluating the metrics [23].

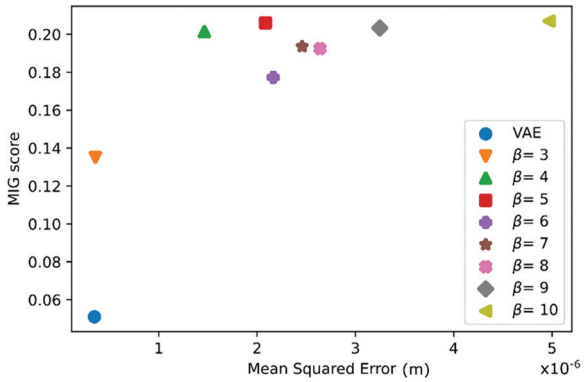

Fig. 5. The disentanglement score (MIG) vs. reconstruction error (MSE) plot for different models.

5. RESULTS AND DISCUSSIONS

In order to evaluate the performance, each model is trained for 100 epochs with a batch size of 128. The resulting z (codes) is evaluated against the factors in the dataset and different metrics calculated for measuring various disentanglement metrics. Fig. 5 represents the reconstruction loss (mean squared error) with a MIG disentanglement score of VAE and βVAE models with different beta values. βVAE models provide better disentanglement by compromising reconstruction quality. The work aims to find the trade-off between reconstruction loss and disentanglement. The VAE model achieves a minimum reconstruction metric (3.4x10-5 m), with a better reconstruction loss but poor disengagement (MIG score = 0.015). $\beta VAE_{(\beta=5)}$ model has a better disentanglement score of 0.206 and a reconstruction error of 2x10^{-4} m. The figure shows that the $\beta VAE_{\beta=10}$ model has a slightly greater MIG score (0.207) than the $\beta VAE_{\beta=5}$ model, but has a higher reconstruction error of 4.9x10^{-4} m. For the performance evaluations, models in VAE, $\beta VAE_{\beta=5}$ and $\beta VAE_{\beta=10}$ are considered.

The evolution of disentanglement metric and reconstruction error over epoch are shown in Fig. 6. MIG metric shows a massive improvement over traditional VAE models but will affect the reconstruction performance. As MIG is not directly linked with the loss function, it does not monotonically increase. Fig. 7 shows the reconstruction performance of models, and the corresponding evaluation metrics are shown in Table 3.

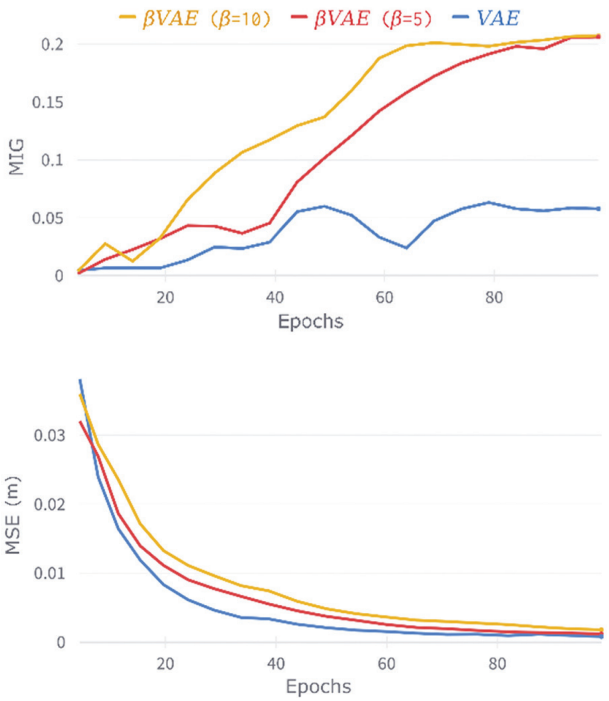

Fig. 6. The evolution of MIG score metric and reconstruction loss (MSE) for different models

Fig. 7. The reconstruction performance of different models - input, VAE, βVAE (β=5), βVAE (β=10).

In order to evaluate the repeatability, input and reconstructed images are plotted in Fig. 8 in the first and second rows, respectively. Joint space reconstruction error in VAE, $\beta VAE_{\beta=5}$ and $\beta VAE_{\beta=10}$ are 4.4 x 10^{-4} ± 1.7 x 10^{-4} m, 5.4 x 10^{-4} ± 4.9 x 10^{-5} m and 3.5 x 10^{-3} ± 2.1 x 10^{-4} m respectively.

And the corresponding robot task space loss (computed by applying forward kinematics) is 8.9 x 10^{-4} ± 2.6 x 10^{-4} m, 3.3 x 10^{-4} ± 5.7 x 10^{-5} m and 2.3 x 10^{-3} ± 2.1 x 10^{-4} m respectively. The joint space loss is less in the case of VAE model, but the tasks space loss is lower in $\beta VAE_{\beta=5}$ model. It is because of the non-linear forward kinematics conversions. The latent dimensional transversal representation of the best-performing model is depicted in Fig. 8, rows 3-12.

Each row shows the transversal of only one code, and the decoded the plot is shown. Fig. 8a, Fig. 8b and Fig. 8c show the transversal in VAE, $\beta VAE_{\beta=5}$ and $\beta VAE_{\beta=10}$ models respectively with Gaussian reconstruction loss. While considering the VAE model in Fig. 8, the 5th and 6th rows show y position transversal and the 5th and 9th rows show x position transversal. Rows 10th and 11th produce similar instances over the changes in the corresponding code.

(a) $\boldsymbol{VAE}$ (b) $\boldsymbol{\beta VAE_{\beta=5}}$ (c) $\boldsymbol{\beta VAE_{\beta=10}}$

Fig. 8. The first row shows the input image, and the second row shows the corresponding reconstructed images for computing the repeatability of the model. Rows 3-12 show the latent transversal performance of the model. The latent transversal performance of each model, each row shows the code, and columns show the transverse in that particular code with all other codes kept constant. The fifth column in each figure shows the reconstructed instance and their transversals generated based on this.

Other rows produce some noise outputs. The $\beta VAE_{\beta=5}$ in Fig. 8b shows x position, y position and orientation transversal in the 5th, 6th and 7th rows, respectively. Scaling is embedded in code in the 3rd and 7th rows. Rows 4th and 8th produce shape transversal, and code variation in 9th -12th rows does not produce much difference. The $\beta VAE_{\beta=10}$ in Fig. 8c produces position x, y and scale transversals in the 5th, 6th and 4th rows, respectively. The 3rd and 8th rows encode code for shape, and rows 7, 9-12 do not produce a visible output difference.

While analyzing the variations, the 5th row in the VAE model changes x and y positions and not all factors of variations are not encoded, while $\beta VAE_{\beta=5}$ and $\beta VAE_{\beta=10}$ encodes the codes and has achieved a higher disentanglement score. It can be observed that the reconstruction performance is quite prominent in lower β values.

Generative models produce samples from a sample distribution, so each time it generates a new sample, it belongs to the same distribution, but some variations exist. It is the property of VAE which causes the variation in standard deviation. Table 3 shows the precision of different models. Robot tasks need to be precise and accurate. The higher value of standard deviation in Reconstruction loss, accuracy and repeatability are due to the generative nature of the model. Accuracy and repeatability are calculated in task space and the Reconstruction loss in the joint space of the robot. The non-linear kinematics operation produces variations between the reconstruction values and the robotics metrics. The model is executed 100 times and generates the plot sown in Fig. 9.

Table 3. Metrics considered.
(↑ Means higher is better).

Model	VAE	βVAE (β=5)	βVAE (β=10)
Rec_loss ↓(x10^{-3} m)	0.346 ± 0.345	2.085 ± 2.714	4.970 ± 7.019
Accuracy ↓ (m)	0.053 ± 0.013	0.055 ± 0.039	0.080 ± 0.065
Repeatability ↓ (m)	0.017 ± 0.002	0.030 ± 0.007	0.039 ± 0.010
MIG [14] ↑	0.051	0.206	0.207
Disentanglement [37] ↑	0.5	0.914	0.786
Completeness [37] ↑	0.128	0.216	0.302
Informativeness [37] ↑	0.122	0.213	0.21
jemmig [36] ↑	0.192	0.297	0.292
βVAE score [15] ↑	0.546	0.617	0.612
FactorVAE [16] ↑	0.611	0.647	0.759

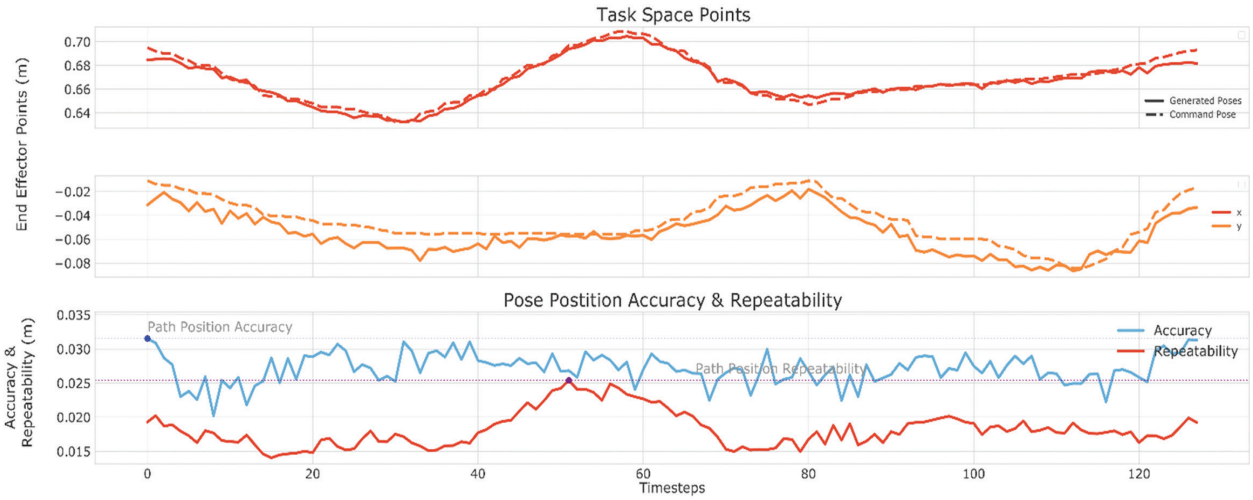

Fig. 9. $\beta VAE_{\beta=5}$ model task space input and output representation in x and y axis is represented (top) and its corresponding accuracy and repeatability is plotted (bottom)

The performance analysis of different disentanglement representations, losses and robot precision are listed in Table 3. Joint space reconstruction loss is lower in the VAE model. As the β value increases, reconstruction performance decreases. However, there is an allowed limit for each task's precision and accuracy range. Optimization of β based on task nature and disentanglement required can be achieved by hyperparameter tuning. Literature shows that the βVAE and FactorVAE scores do not provide practically feasible metrics. This work uses the MIG score as the primary metric for evaluating disentanglement.

6. CONCLUSION

In this work, CNN-based Variational Autoencoder models have been utilized for disentanglement representation of robot tasks. All the models have been trained on the robot disentanglement dataset proposed. Popular disentanglement metrics such as MIG score, DCI, jemmig, VAE and FactorVAE scores are used to evaluate the model performance as well as robot accuracy and precision metrics. From various disentanglement metrics, it has been found that the underlying factors of variation in tasks learn better with disentanglement losses. The model has been found to generate disentanglement representation with a path position accuracy of 0.055, close to the VAE model (0.053) and better disentanglement of 0.206, which is far better than the state-of-the-art VAE model.

The disentanglement generative models can be used as a supervised data generator for training deep learning models and can be directly used in robot task generation applications (e.g. Painting tasks).

7. ACKNOWLEDGEMENTS

The authors would like to acknowledge University Grants Commission (UGC), India for providing financial assistance to carry out this research work.

8. REFERENCES

[1] H. Demir, F. Sari, M. R. Tolun, "Review for path planning for robots and its applications", ACADEMIC STUDIES, 1st Edition, 2019, pp. 196-211.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial networks", Proceedings of the Advances in Neural Information Processing Systems, Montreal, Quebec, Canada, 8-13 December 2014, pp. 2672-2680.

[3] D. P. Kingma, W. Max, "Auto-encoding variational bayes", arXiv:1312.6114, 2013.

[4] D. J. Rezende, S. Mohamed, D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models", Proceedings of the 31st International Conference on Machine Learning, Bejing, China, 22-24 June 2014, pp. 1278-1286.

[5] R. T. Chen, X. Li, R. Grosse, D. Duvenaud, "Isolating sources of disentanglement in variational autoencoders", arXiv:1802.04942, 2013.

[6] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, A. Lerchner, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework", Proceedings of the 5th International Conference on Learning Representations, Toulon, France, April 2017, pp. 24-26.

[7] Y. Hristov, D. Angelov, M. Burke, A. Lascarides, S. Ramamoorthy, "Disentangled Relational Rep-

resentations for Explaining and Learning from Demonstration", Proceedings of the Conference on Robot Learning, Osaka, Japan, 30 October - 1 November 2019, pp. 870-884.

[8] V. A. Kumar, T. Rajan, A. Nagendran, A. Dehghani-Sanij, R. C. Richardson, "Tether monitoring for entanglement detection, disentanglement and localisation of autonomous robots", Robotica, Vol. 34, No. 3, 2016, pp. 527-548.

[9] M. Wulfmeier, A. Byravan, T. Hertweck, I. Higgins, A. Gupta, T. Kulkarni, M. Reynolds, D. Teplyashin, R. Hafner, T. Lampe, M. Riedmiller, "Representation matters: Improving perception and exploration for robotics", Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May - 5 June 2021, pp. 6512-6519.

[10] C. Qin, Y. Zhang, Y. Liu, S. Coleman, D. Kerr, G. Lv, "Appearance invariant place recognition by adversarially learning disentangled representation", Robotics and Autonomous Systems, Vol. 131, No. 9, 2020, p. 103561.

[11] J. Pajarinen, O. Arenz, J. Peters, G. Neumann, "Probabilistic approach to physical object disentangling", IEEE Robotics and Automation Letters, Vol. 5, No. 4, 2020, pp. 5510-5517.

[12] M. Zolotas, Y. Demiris, "Disentangled sequence clustering for human intention inference", arXiv:2101.09500.

[13] M. T. Das, L. C. Dülger, "Mathematical modelling, simulation and experimental verification of a SCARA robot", Simulation Modelling Practice and Theory, Vol. 13, No. 3, 2005, pp. 257-271.

[14] J. Denavit, R. S. Hartenberg., "A kinematic notation for lower-pair mechanisms based on matrices", Journal of Applied Mechanics, Vol. 22, No. 2, 1955, pp. 215-221.

[15] M. Quigley, et al. "ROS: an open-source Robot Operating System", Proceedings of the ICRA workshop on open source software, Kobe, Japan, 12-17 May 2009, p. 5.

[16] P. Corke, J. Haviland, "Not your grandmother's toolbox–the robotics toolbox reinvented for python", Proceedings of the IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May -5 June 2021, pp. 11357-11363.

[17] L. Matthey, I. Higgins, D. Hassabis, A. Lerchner, dsprites: "Disentanglement testing sprites dataset", https://github.com/deepmind/dsprites-dataset/ (accessed: 2021)

[18] S. Chitta, I. Sucan, S. Cousins, "Moveit![ros topics]", IEEE Robotics & Automation Magazine, Vol. 19, No. 1, 2012, pp. 18-19.

[19] ISO 9283:1998(en), "Manipulating industrial robots-Performance criteria and related test methods", International Organization for Standardization, Geneva, Switzerland, Technical Report, 1998.

[20] H. Kim, A. Mnih, "Disentangling by factorising", Proceedings of the 35th International Conference on Machine Learning, Stockholm Sweden, 10-15 June 2018, pp. 2649-2658.

[21] K. Do, T. Tran, "Theory and evaluation metrics for learning disentangled representations", Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26 April - 1 May, 2020.

[22] C. Eastwood, C. K. I. Williams, "A framework for the quantitative evaluation of disentangled representations", Proceedings of the International Conference on Learning Representations, Vancouver, Canada, 30 April - 3 May 2018.

[23] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations", Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10-15 June 2019, pp. 4114-412.