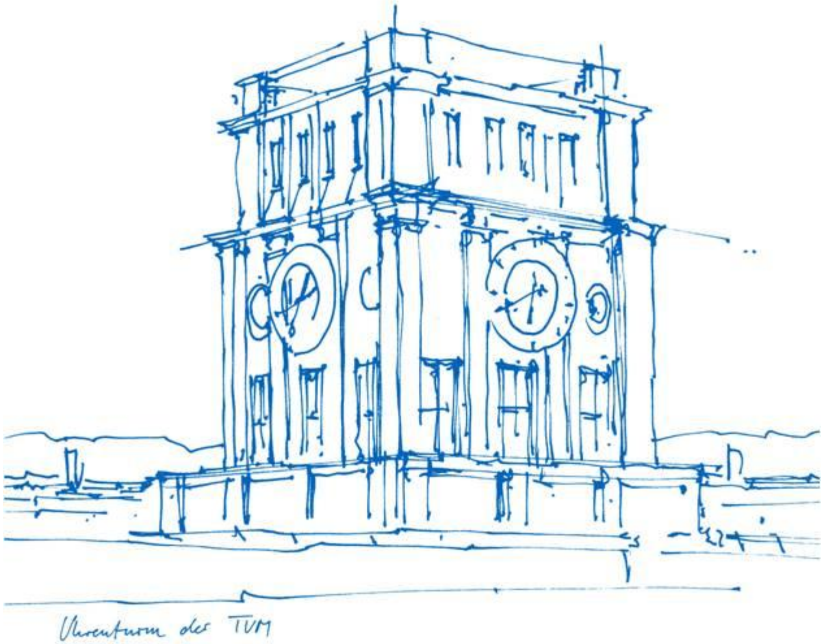
Neural Networks for Inverse Kinematics Problems in Robotics

Franziska Schwaiger and Thomas Brunner

Technische Universität München

10 March 2021

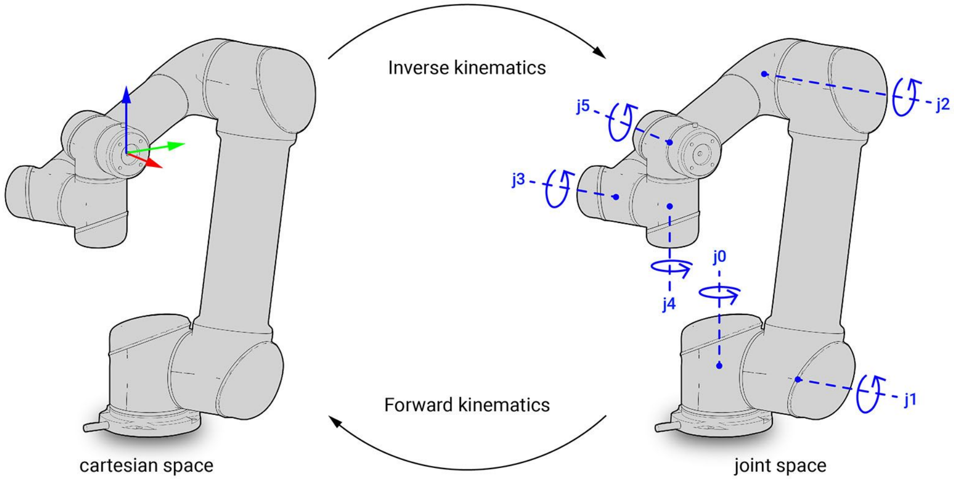


Motivation - Inverse Kinematics

Determining the joint variables corresponding to an end-effector position and orientation

Complex problem: nonlinear, closed-form solution only for few cases

Problem can be ill-posed: single, multiple, infinite or no solutions



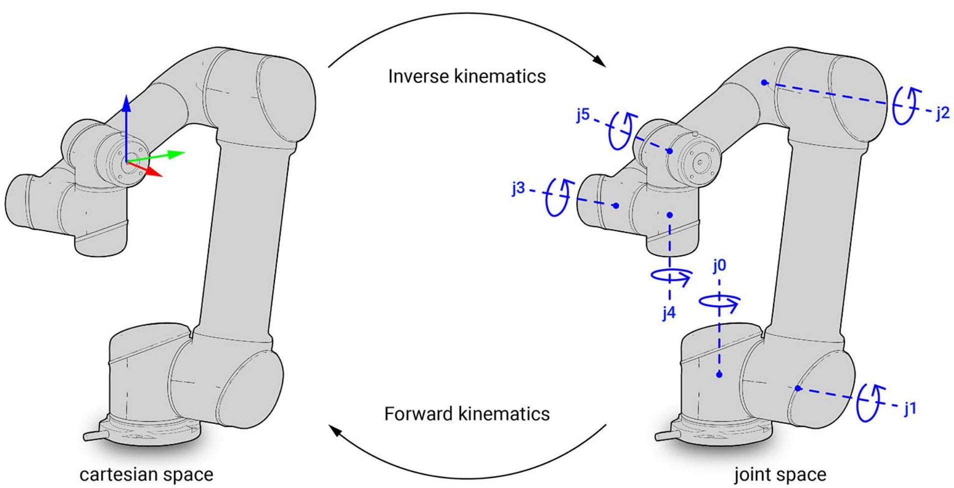
Motivation - Inverse Kinematics

Classical approaches:

analytical solution → not always possible

numerical solution → expensive, limited to few solutions

Can we use neural networks to learn inverse kinematics?



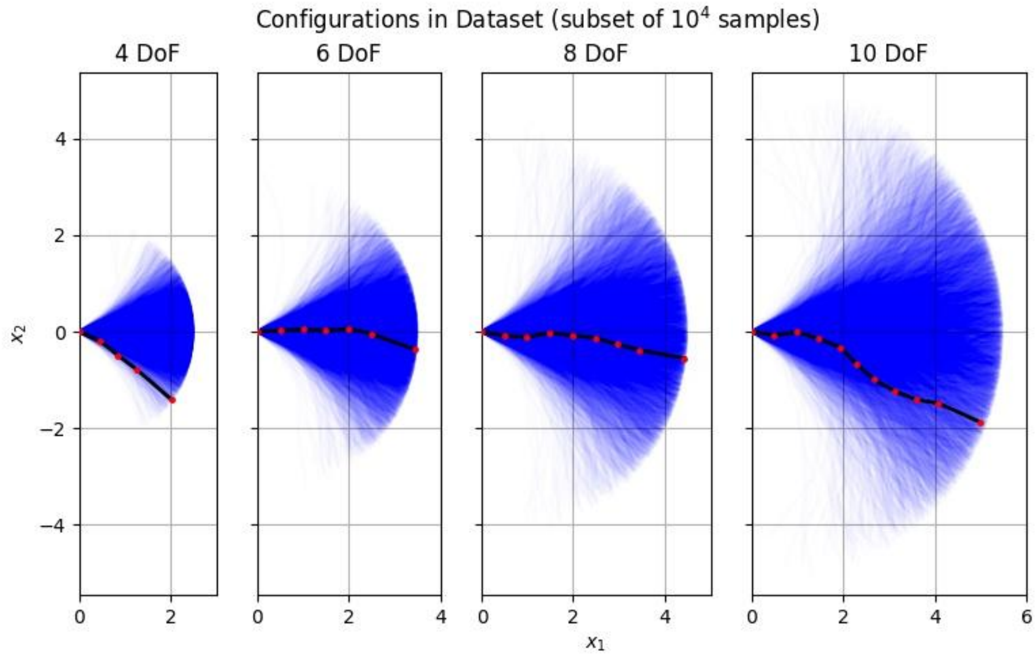
Methods - Dataset

Joint orientations from a normal distribution: $\theta_i \sim \mathcal{N}(\mu = 0, \sigma = 0.2)$

10^6 samples of joint configurations and TCP coordinates

$$x_{\mathrm{TCP}} = \sum_{i=1}^{N} l_i \cos\left(\sum_{j=1}^{i} \theta_j\right)$$

$$y_{\mathrm{TCP}} = \sum_{i=1}^{N} l_i \sin\left(\sum_{j=1}^{i} \theta_j\right)$$



Methods - Hyperparameter Search

Compare performance of networks in fair setting

Hyperparameter search using Ray Tune and Scikit Optimize

Train 100 samples using subset of original dataset (10^4 samples)

TABLE I
RANGES OF INN HYPERPARAMETERS USED DURING HYPERPARAMETER SEARCH.

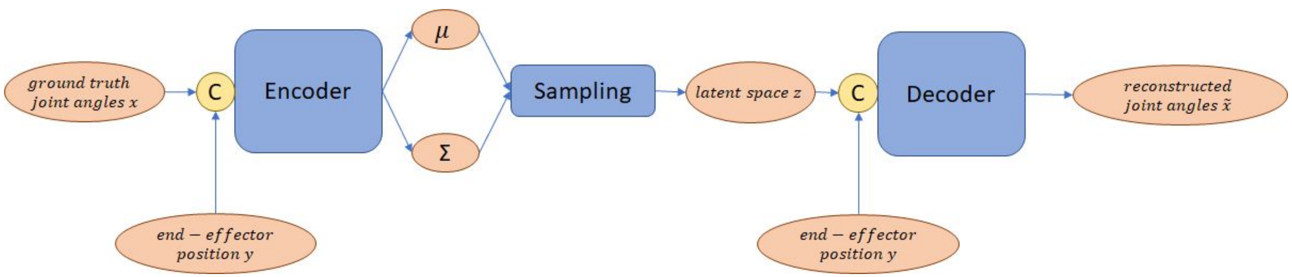
	Learning rate	# Subnet layers	# Coupling layers	# Neurons per layer	Weight decay
Min	0.0001	3	6	100	0.00001
Max	0.005	7	10	300	0.001

TABLE II
RANGES OF CVAE HYPERPARAMETERS USED DURING HYPERPARAMETER SEARCH.

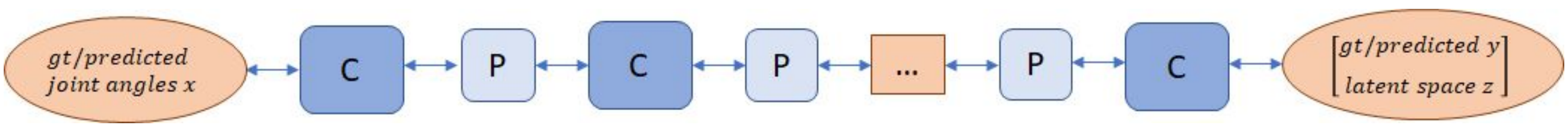
	Learning rate	# Layers	# Neurons per layer	Weight decay
Min	0.0001	3	200	0.00001
Max	0.01	15	500	0.001

Recap: Model Architectures

Conditional Variational Autoencoder (cVAE)



Invertible Neural Network (INN)



Recap: Evaluation

1. Average mismatch between true posterior and predicted posterior

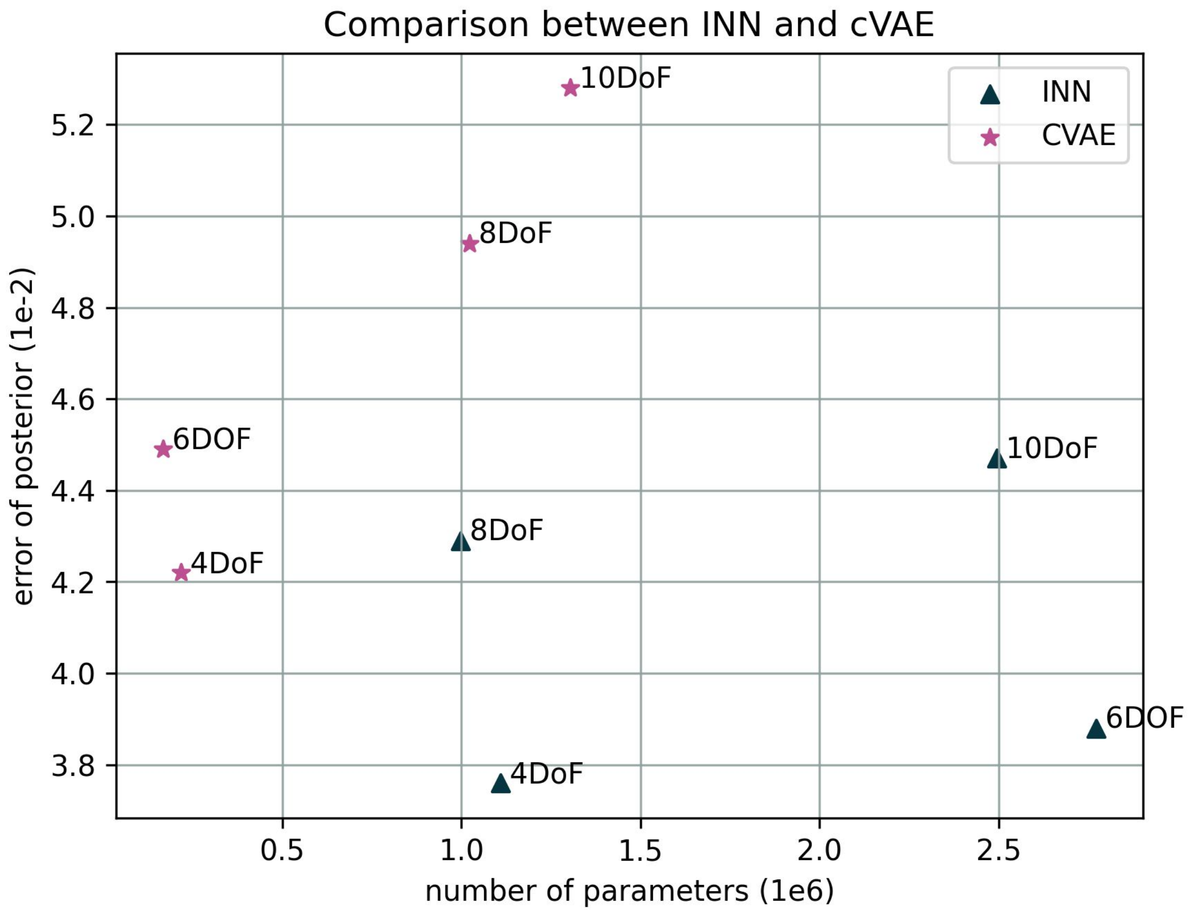
$$e_{posterior} = MMD(\tilde{p}(x|y_{gt}), p_{gt}(x|y_{gt}))$$

2. Average re-simulation error

$$e_{resim} = E_{x \sim \tilde{p}(x|y_{gt})}(||f(x) - y_{gt}||_2^2)$$

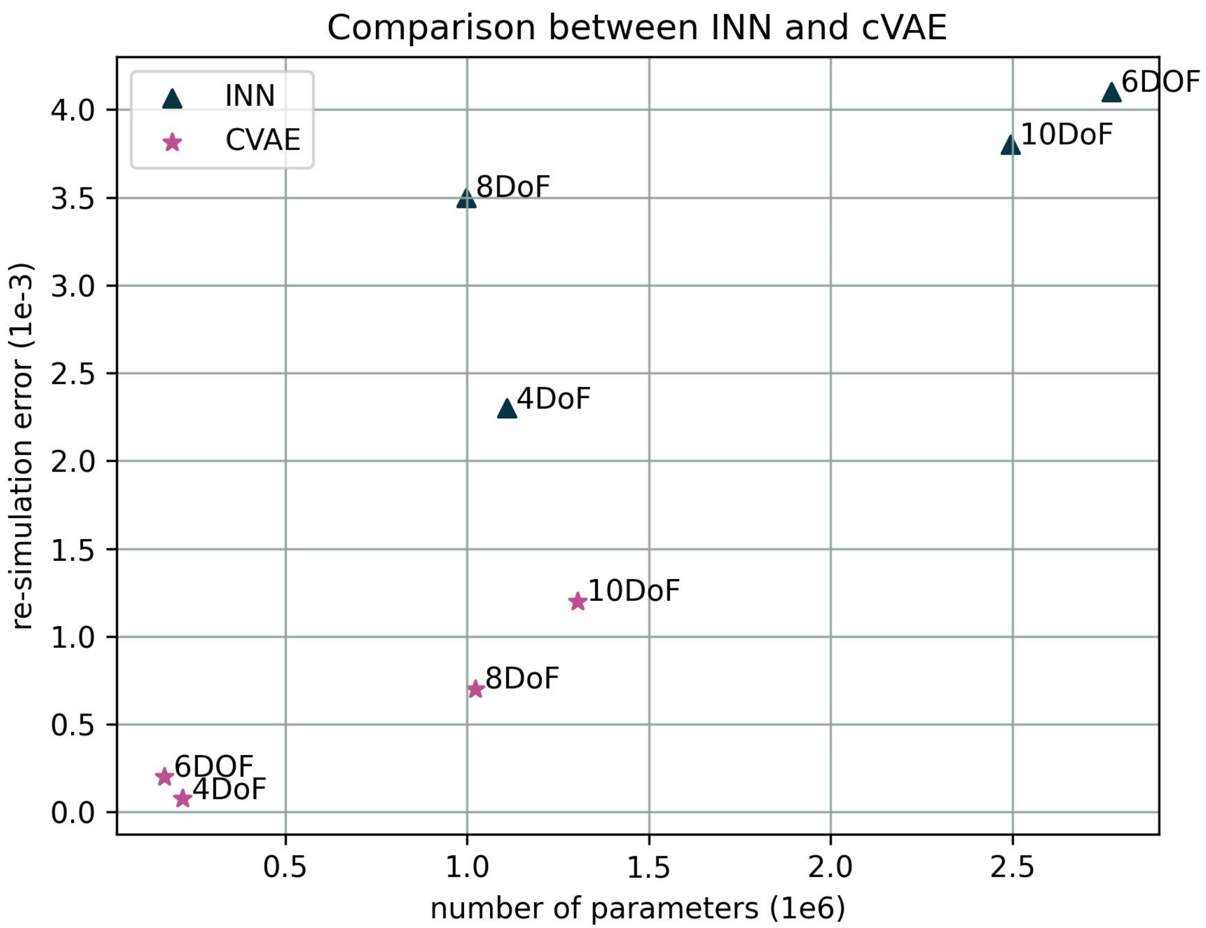
Results

Average mismatch between true posterior and predicted posterior

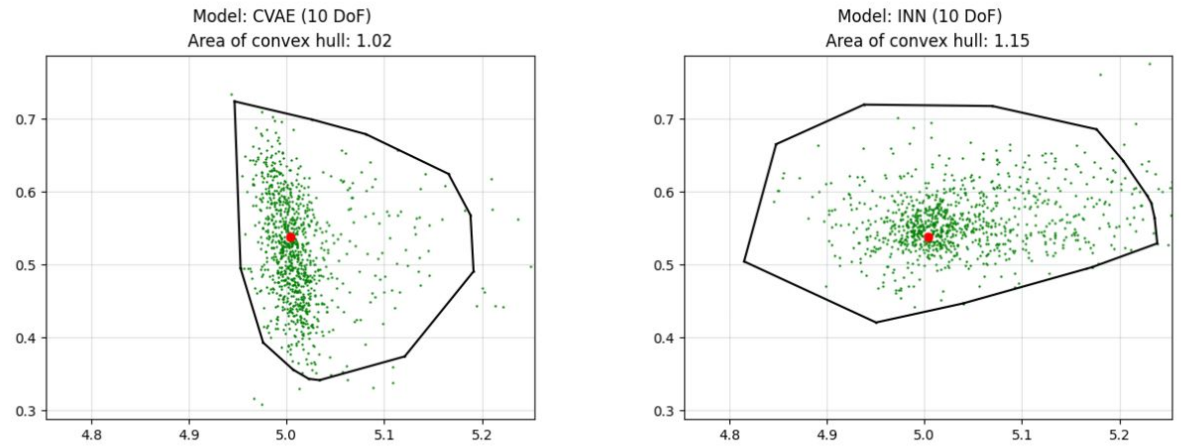


Results

Average re-simulation error

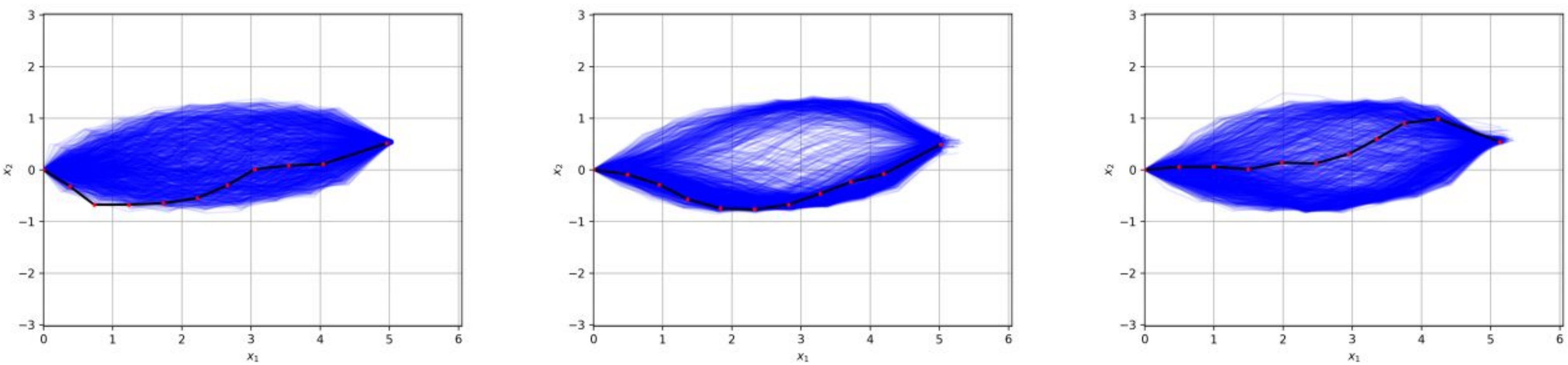


Results



(a) cVAE (b) INN

Fig. 4. Area of the convex hull of the 0.97 percentile of the re-simulated end-effector coordinates with the ground truth end-effector position at $(x, y) = [5.00, 0.53]$ and 1000 samples. Number of DoF: 10.



(a) Rejection Sampling (b) cVAE (c) INN

Fig. 6. Arm configuration of a planar manipulator with 10 revolute joints and end-effector position at $(x, y) = [5.00, 0.53]$. 1000 samples are drawn from each model's predicted posterior $\tilde{p}(x|y_{gt})$, one random sample configuration is highlighted. $e_{posterior} = 0.057$ for the cVAE and $e_{posterior} = 0.025$ for the INN.

Conclusion

- Evaluation of INN and cVAE through simulation experiments in the context of inverse kinematics
- Finding best set of hyperparameters via Bayesian Optimization
- cVAE performed better on average re-simulation error
- INN performed better in estimating the posterior distribution of the joint angles
- INN: additional unsupervised backward training to match performance of cVAE

References

[1] Lynton Ardizzone, Jakob Kruse, Sebastian J. Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. CoRR, abs/1808.04730, 2018

[2] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. 2015. Learning structured output representation using deep conditional generative models. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15). MIT Press, Cambridge, MA, USA, 3483–3491.

[3] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. 2010. Robotics: Modelling, Planning and Control. Springer Publishing Company, Incorporated.

[4] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. CoRR, abs/1605.08803, 2016.

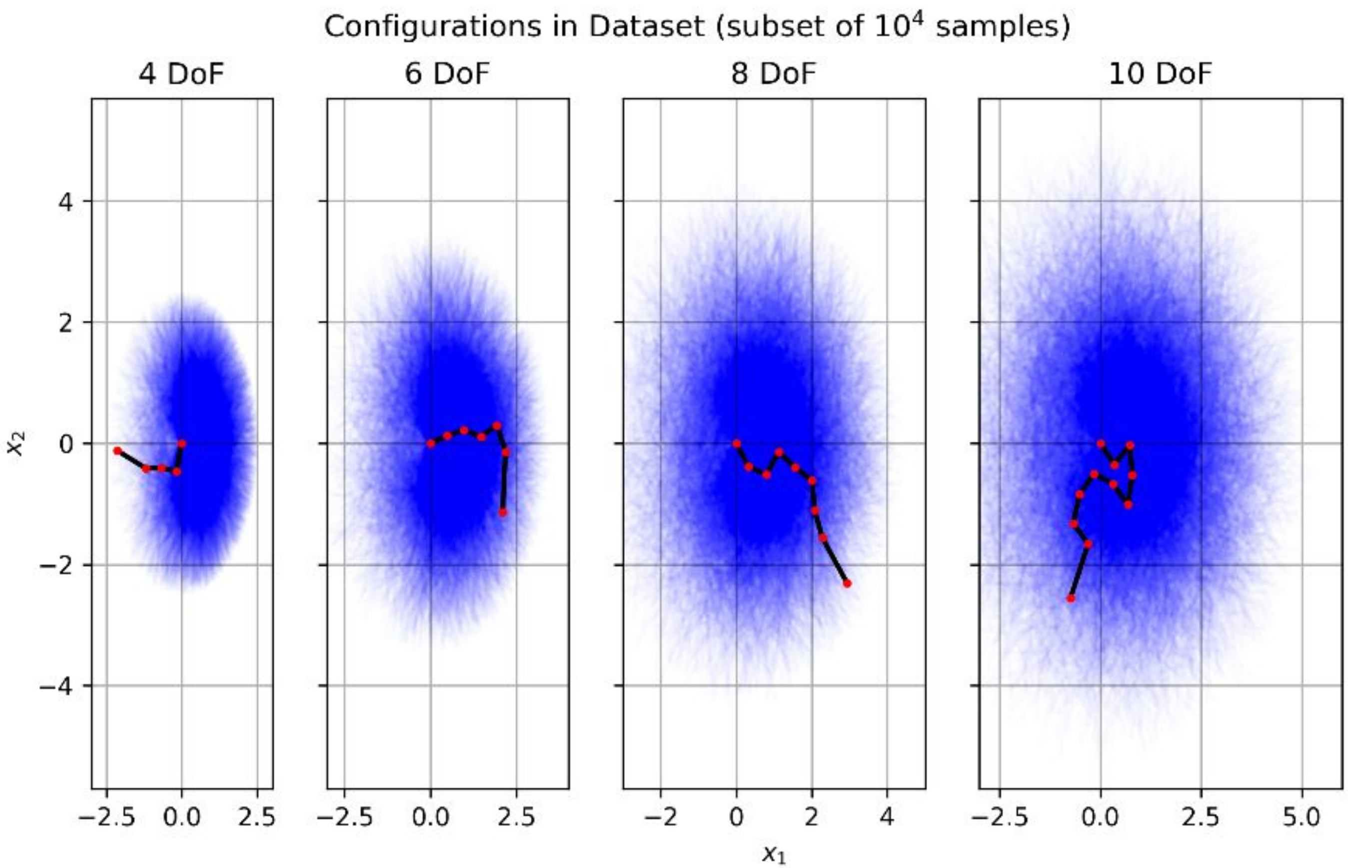
[5] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample
problem. CoRR, abs/0805.2368, 2008.

[6] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014.

Questions

Backup Slides

Dataset with degenerated configurations (std=1.0)



Backup Slides

Results for HPO

TABLE III
HYPERPARAMETERS FOR CVAE TRAINED ON A PLANAR ROBOT WITH REVOLUTE JOINTS

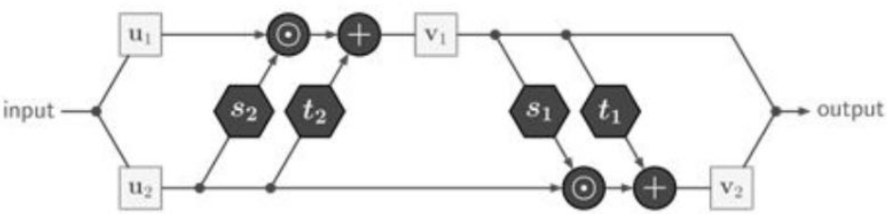
DoF	Learning rate	# Layers	# Neurons per layer	Weight decay	# Trainable parameters
4	0.009	3	230	0.0002	217128
6	0.0041	3	200	0.00001	166814
8	0.0001	3	500	0.00001	1022020
10	0.00063	5	400	0.00029	1303226

TABLE IV
HYPERPARAMETERS FOR INN TRAINED ON A PLANAR ROBOT WITH REVOLUTE JOINTS

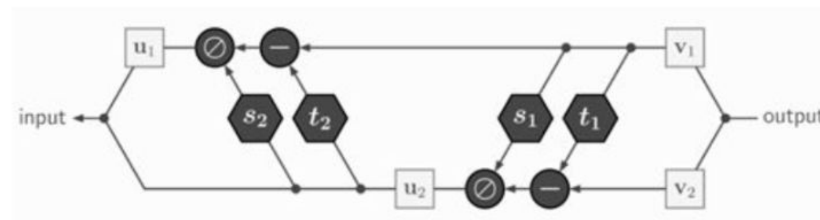
DoF	Learning rate	# Subnet layers	# Coupling layers	# Neurons per layer	Weight decay	# Trainable parameters
4	0.0009	5	9	100	0.00008	1108872
6	0.001	5	7	180	0.0003	2772084
8	0.0014	4	9	115	0.0004	997884
10	0.002	5	7	170	0.0005	2494380

Backup Slides

INN: Affine Coupling Layers



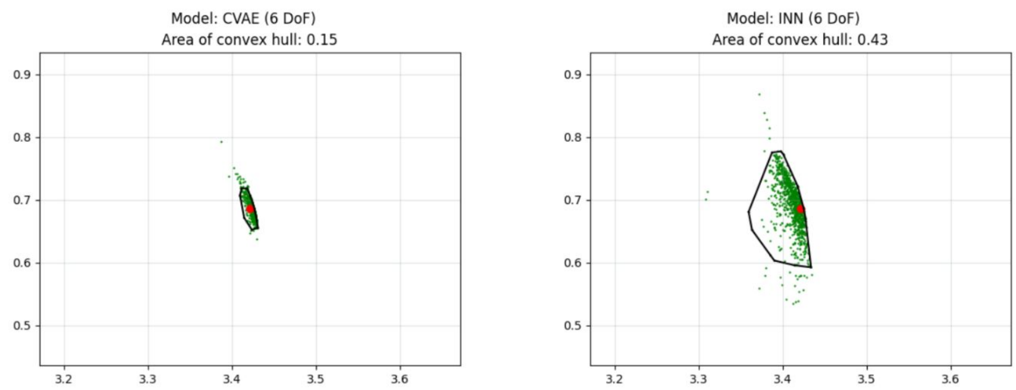
$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp\big(s_2(\mathbf{u}_2)\big) + t_2(\mathbf{u}_2)$$
$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp\big(s_1(\mathbf{v}_1)\big) + t_1(\mathbf{v}_1),$$



$$\mathbf{u}_2 = \big(\mathbf{v}_2 - t_1(\mathbf{v}_1)\big) \odot \exp\big(-s_1(\mathbf{v}_1)\big)$$
$$\mathbf{u}_1 = \big(\mathbf{v}_1 - t_2(\mathbf{u}_2)\big) \odot \exp\big(-s_2(\mathbf{u}_2)\big).$$

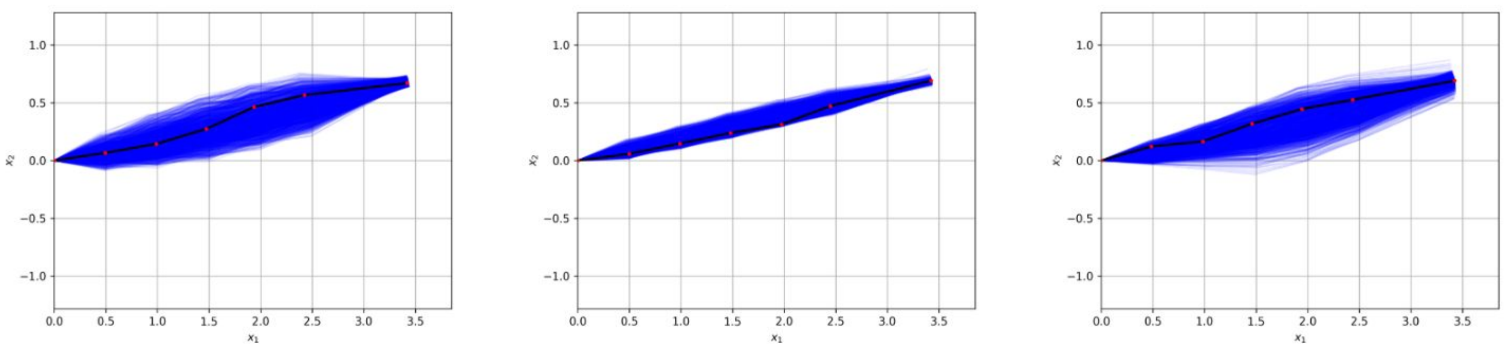
Backup Slides

Additional example



(a) cVAE (b) INN

Fig. 4. Area of the convex hull of the 97th percentile of the re-simulated end-effector coordinates with the ground truth end-effector position at $(x, y) = [3.42, 0.68]$ and 1000 samples. Number of DoF: 6.



(a) Rejection Sampling (b) cVAE (c) INN

Fig. 6. Arm configuration of a planar manipulator with 6 revolute joints and end-effector position at $(x, y) = [3.42, 0.68]$. 1000 samples are drawn from each model's predicted posterior $\tilde{p}(x|y_{gt})$, one random sample configuration is highlighted. $e_{posterior} = 0.012$ for the cVAE and $e_{posterior} = 0.0068$ for the INN.

Backup Slides

Loss for cVAE:

$$L_y = -KL[q_{w_{enc}}(z|x,y)||p(z) \sim N(0,1)]$$

$$L_x = E_{q_{w_{enc}}(z|x,y)}[\log(p_{w_{dec}}(x|z,y))]$$

Loss for INN:

$$[y,z] = [f_y(x), f_z(x)]$$

$$L_y = MSE(y_i, f_y(x_i))$$

$$L_x = MSE(x_i, [f_y^{-1}(y_i), f_z^{-1}(f_z(x_i))])$$

$$L_{p(z)} = MMD(p(f_z(x_i)), p(z) \sim N(0,1))$$

$$L_{p(x)} = MMD(p([f_y^{-1}(y_i), f_z^{-1}(p(z))]), p(x))$$