Milestone Report

Franziska Schwaiger
Matriculation number: 03658670

Thomas Brunner
Matriculation number: 03675118

DATASET

METHODS

The following two model architectures have been implemented in PyTorch and are inspired by GitHub repositories [1], [2].

Conditional Variational Autoencoder

In the lecture, we have already discussed Variational Autoencoders (VAE) [3] in detail. One drawback of this type of generative architecture is that there is no control over the data generation process. In our case this would mean that although we can generate random sample configurations we cannot control in which end-effector position these configurations would result. Conditional Variational Autoencoders (cVAE) [4] solve this problem by conditioning the latent space z. For inverse kinematics this means that random samples are drawn from $p(z) \sim N(0,1)$ and the predicted posterior of the joint angles is then generated conditioned on the end-effector position. During training, the position (x, y) is concatenated with both x and z and then fed into the encoder and decoder, respectively.

The cVAE is trained in the same manner as the VAE based on the ELBO (Evidence Lower Bound) loss $L_{ELBO} = L_y + L_x$:

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

The Kullback-Leibler divergence measures the distance between the predicted probability distribution $q_\phi(z|x,y)$ of the latent space z and the standard normal distribution. The reconstruction loss is defined as follows:

$$L_x = E_{q_\phi(z|x)}[log(p_\theta(x|z,y))] = \sum_{i=0}^{N} MSE(v_i \cdot \tilde{v}_i, 1) \tag{2}$$

Here, N is denoted as the number of joints. For representing the joint angles of the robot, we use a vector-based representation: $V = (sin(\theta), cos(\theta))$ to avoid singularities at the boundaries of the joint angles. $v_i \cdot \tilde{v}_i$ is denoted as the scalar product between the normalized predicted posterior point estimate $\tilde{v}_i = (sin(\tilde{\theta}), cos(\tilde{\theta}))$ and the ground truth vector $v_i = (sin(\theta), cos(\theta))$ of the ith joint.

DoF	$e_{posterior}$	e_{resim}	Trainable Parameters	Model
2	0.077	**0.003**	164,808	cVAE
3	**0.045**	0.045	370,214	
4	0.063	**0.006**	373,220	
2	**0.061**	0.012	169,632	INN
3	0.066	**0.036**	369,660	
4	**0.044**	0.075	374,960	

TABLE I
CAPTION TO COME

Invertible Neural Network

EXPERIMENTAL EVALUATION

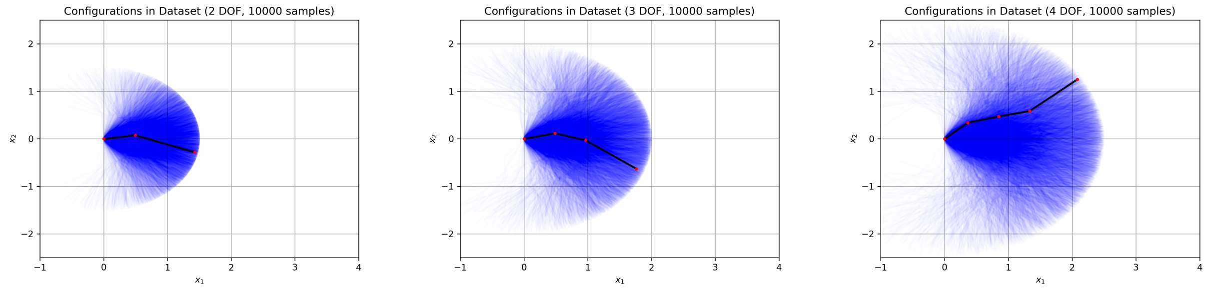
Evaluation protocol

Results

NEXT STEPS

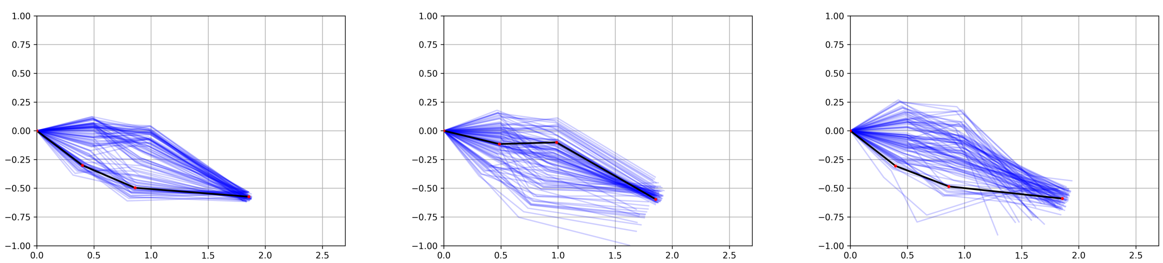
REFERENCES

[1] graviraja, "pytorch-sample-codes," https://github.com/graviraja/pytorch-sample-codes, 2019.
[2] "Freia," https://github.com/VLL-HD/FrEIA/blob/master/FrEIA/, 2020.
[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014.
[4] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015, pp. 3483–3491.
[5] L. Ardizzone, J. Kruse, S. J. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother, and U. Köthe, "Analyzing inverse problems with invertible neural networks," *CoRR*, vol. abs/1808.04730, 2018. [Online]. Available: http://arxiv.org/abs/1808.04730
[6] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014. [Online]. Available: http://arxiv.org/abs/1411.1784
[7] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample problem," *CoRR*, vol. abs/0805.2368, 2008. [Online]. Available: http://arxiv.org/abs/0805.2368
[8] Y. Zhou, C. Barnes, L. Jingwan, Y. Jimei, and L. Hao, "On the continuity of rotation representations in neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
[9] J. Kruse, L. Ardizzone, C. Rother, and U. Köthe, "Benchmarking invertible architectures on inverse problems," Tech. Rep. i, 2019.
[10] B. Choi and C. Lawrence, "Inverse kinematics problem in robotics using neural networks," Tech. Rep., 1992.
[11] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," *CoRR*, vol. abs/1605.08803, 2016. [Online]. Available: http://arxiv.org/abs/1605.08803



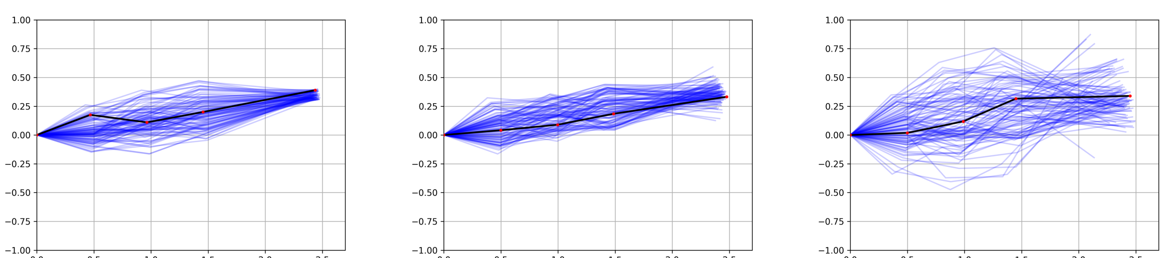
(a) 2 DOF (b) 3 DOF (c) 4 DOF

Fig. 1. Illustration of datasets used during training of models. Only a subset of the samples contained in the datasets is shown here. One configuration in the dataset is highlighted to illustrate the configuration of the robot arm.



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

Fig. 2. Arm configuration of a planar manipulator with 3 revolute joints and end-effector position at $(x, y) = [1.83, -0.57]$. 100 samples are drawn from each model's predicted posterior $\tilde{p}(x|y_{gt})$, one random sample configuration is highlighted.



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

Fig. 3. Arm configuration of a planar manipulator with 4 revolute joints and end-effector position at $(x, y) = [2.44, 0.35]$. 100 samples are drawn from each model's predicted posterior $\tilde{p}(x|y_{gt})$, one random sample configuration is highlighted.