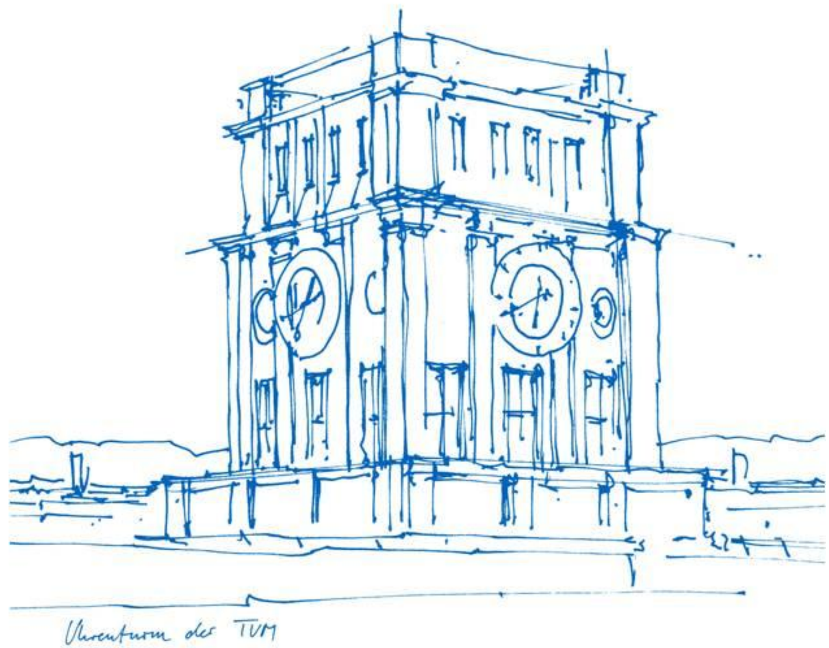
Invertible Neural Networks for Inverse Kinematics

Franziska Schwaiger and Thomas Brunner

Technische Universität München

14 January 2021

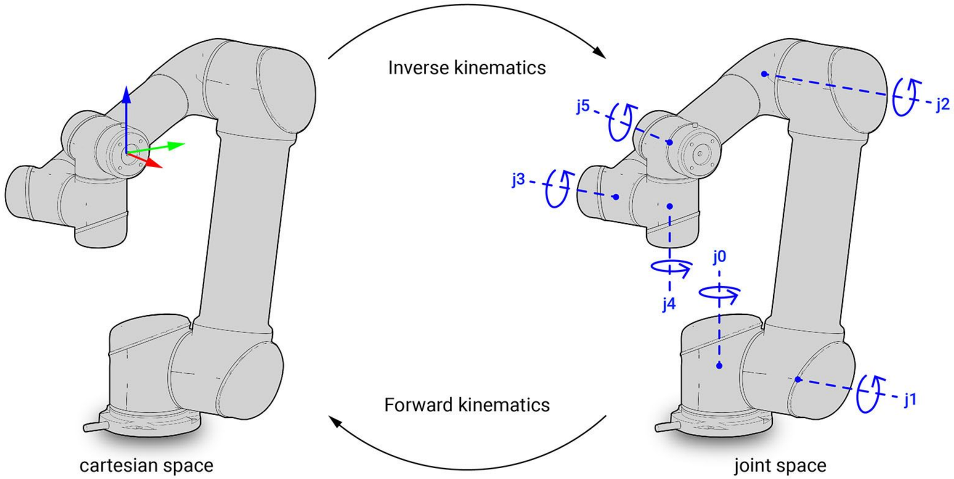


Motivation - Inverse Kinematics

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



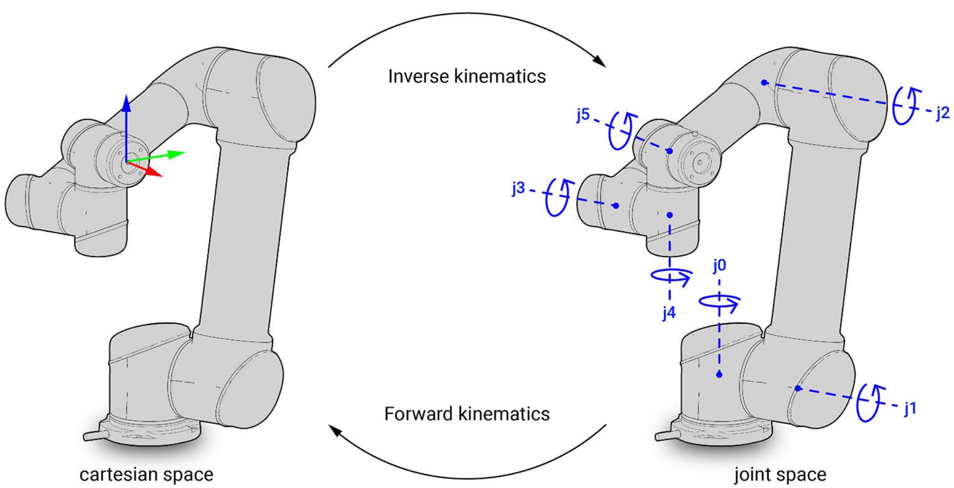
Motivation - Inverse Kinematics

Classical approaches:

analytical solution → not always possible

numerical solution → expensive, limited to few solutions

Can we use neural networks to learn inverse kinematics?

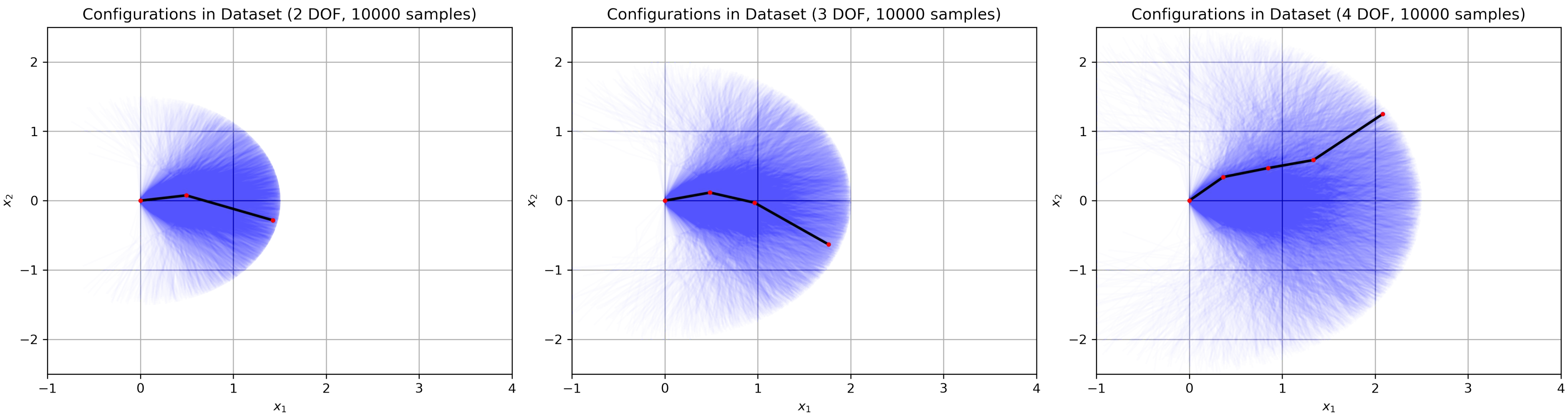


Methods - Dataset

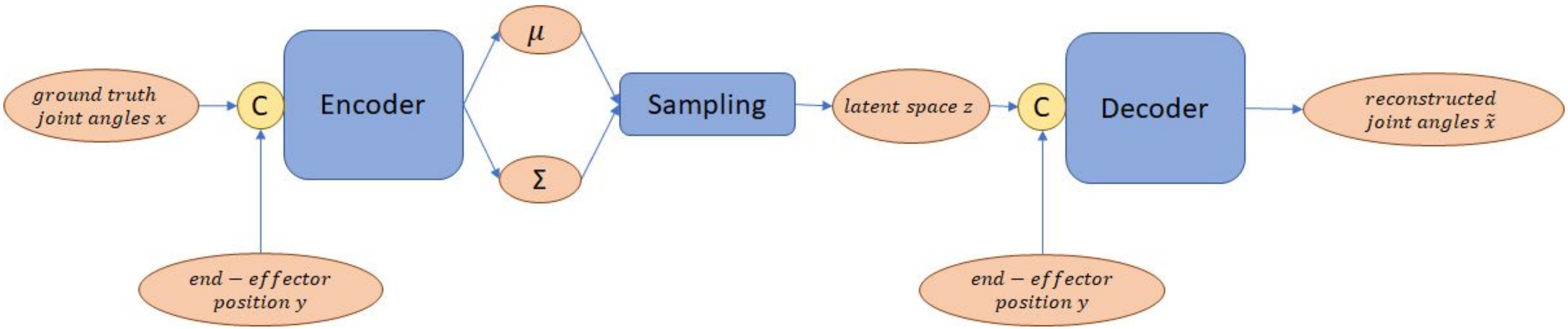
Simulation of different planar robots: 2 DoF, 3 DoF and 4 DoF

Joint orientations from a normal distribution: $\theta_i = \mathcal{N}(\mu = 0, \sigma^2 = 0.25)$

One million samples of robot configurations

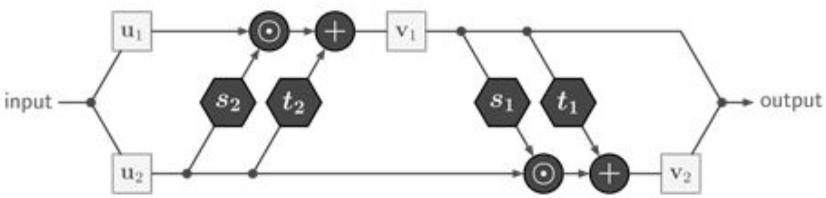


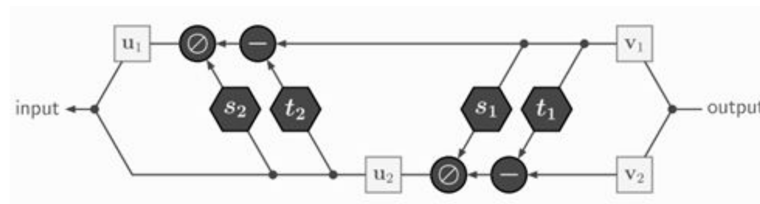
Methods - Conditional Variational Autoencoder (cVAE)



Methods - Invertible Neural Network (INN)

Affine Coupling Layer



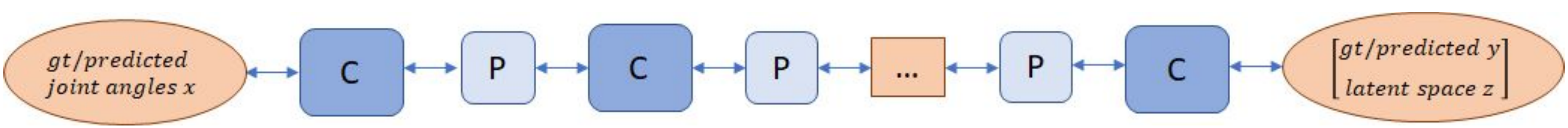


$$\mathbf{v}_1 = \mathbf{u}_1 \odot \exp\big(s_2(\mathbf{u}_2)\big) + t_2(\mathbf{u}_2)$$
$$\mathbf{v}_2 = \mathbf{u}_2 \odot \exp\big(s_1(\mathbf{v}_1)\big) + t_1(\mathbf{v}_1),$$

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

C: coupling layer

P: fixed but random permutation block



Results

1. Average mismatch between true posterior and predicted posterior

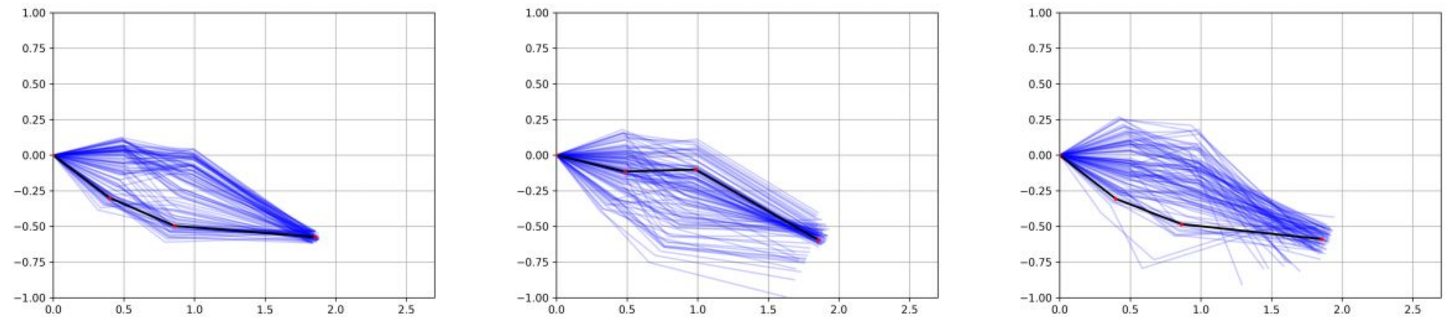
$$e_{posterior} = MMD(\tilde{p}(x|y_{gt}), p_{gt}(x|y_{gt}))$$

2. Average re-simulation error

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

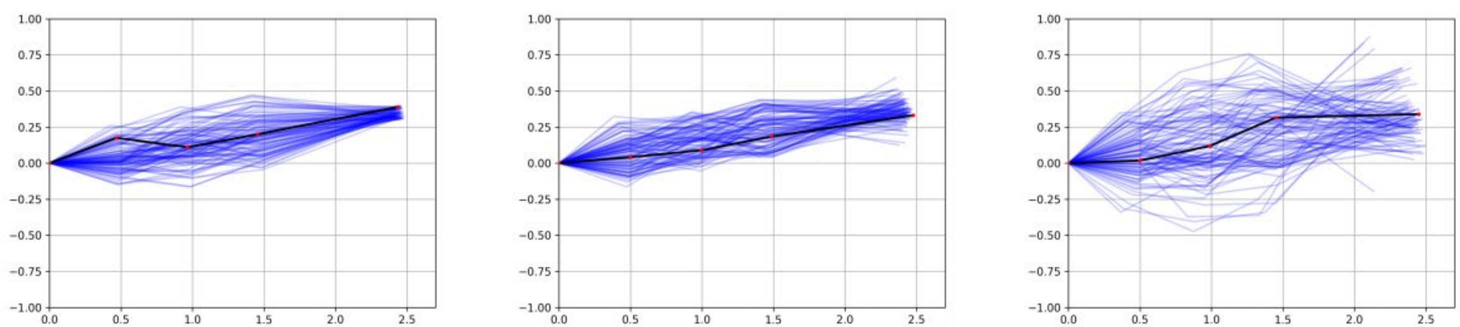
DoF	$e_{posterior}$	e_{resim}	Trainable Parameters	Model
2	0.077	**0.003**	164,808	cVAE
3	**0.045**	0.045	370,214	
4	0.063	**0.006**	373,220	
2	**0.061**	0.012	169,632	INN
3	0.066	**0.036**	369,660	
4	**0.044**	0.075	374,960	

Results



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

Fig. 1: 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. 2: 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.

Next Steps

1. Increase DoFs of 2D robots and see where models break
2. Extend to 3D robots
3. Hyperparameter optimization (Random Search)

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