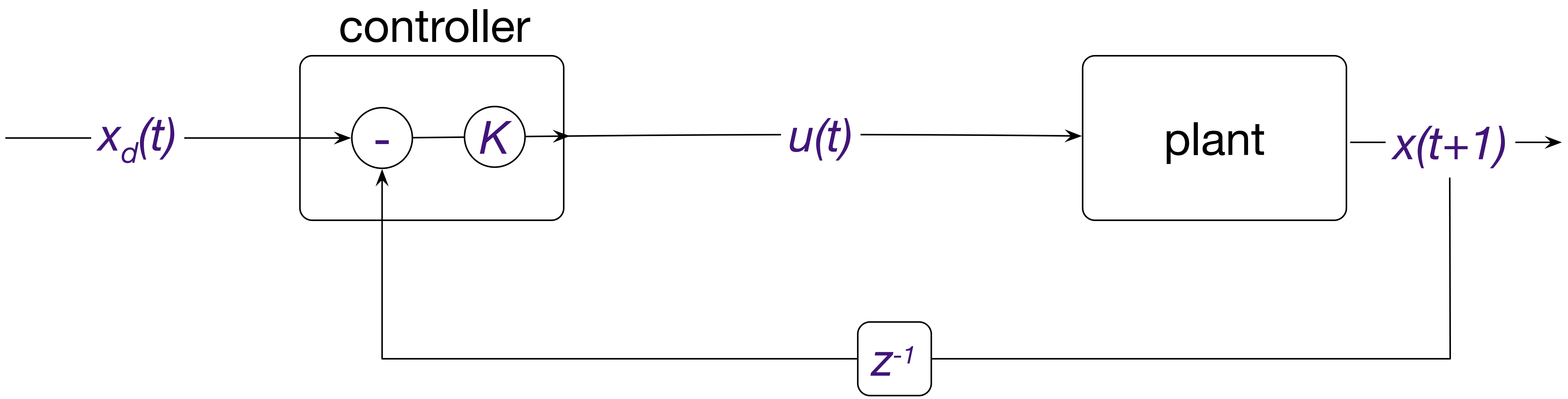
on variational inference and optimal control

ARGMAX.ai
VOLKSWAGEN GROUP AI RESEARCH

Patrick van der Smagt
Director of AI Research
Volkswagen Group
Munich, Germany

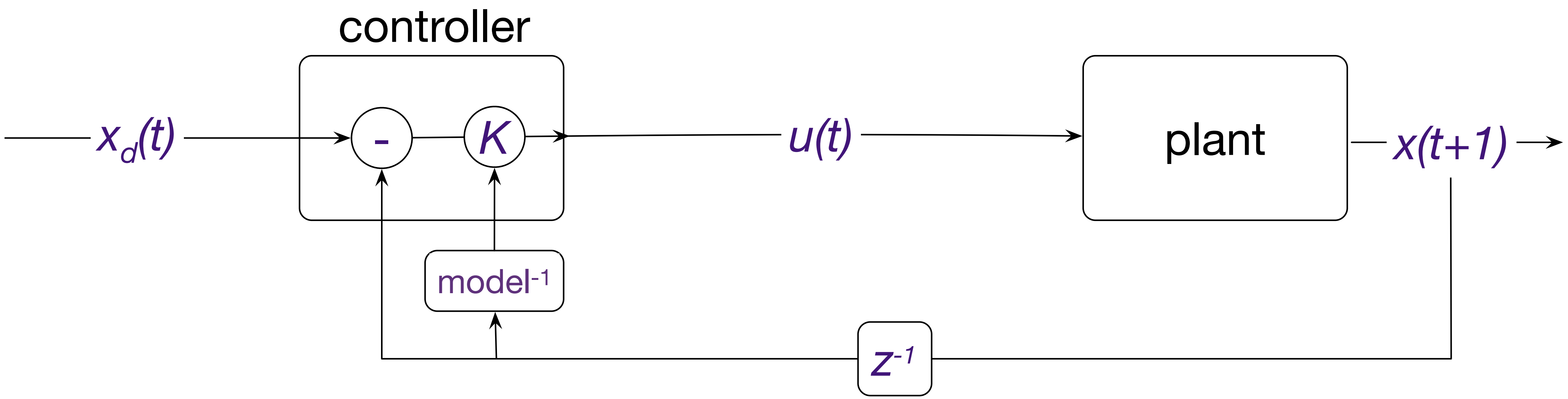
https://argmax.ai

control approach #1: feedback control



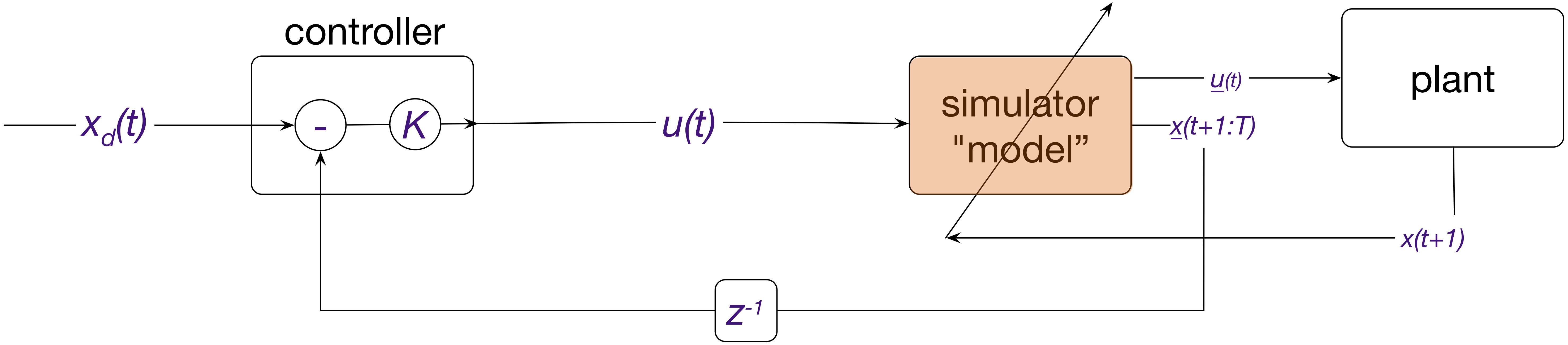
problem: requires *very* fast feedback loop

control approach #2: model-based feedback control



problem: requires fast feedback loop and inverse model

control approach #3: model-reference control



simulator "dreams" the future, aka **predictive coding**

problem: how do I get this model?

problems

1) engineered models are expensive to set up

2) engineered models are expensive to compute

3) engineered models do not scale

we really want to represent $p(x)$

we can write

$$p(x) = \int p(x \mid z)\, p(z)\, dz$$

we really want to represent $p(x)$

$$p(x) = \int p(x \mid z)\, p(z)\, dz$$

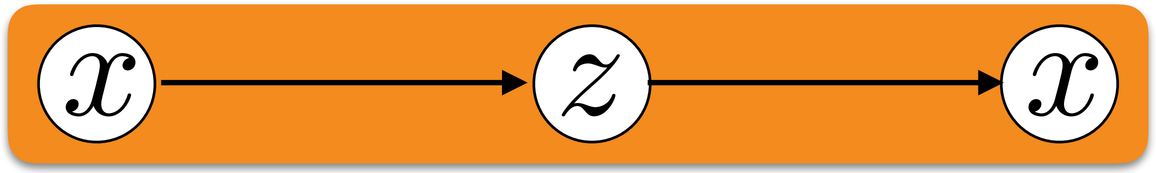
Two problems:

(1) how do we shape z to carry the right information of x? A: We don't hand-design it. Assume it is a Gaussian pd.

(2) how do we compute the integral? It is intractable (we **only** have the data; need MCMC)

we really want to represent $p(x)$

$$p(x) = \int p(x \mid z)\, p(z)\, dz$$



bummer, we don't have it

Trick to do efficient MCMC:

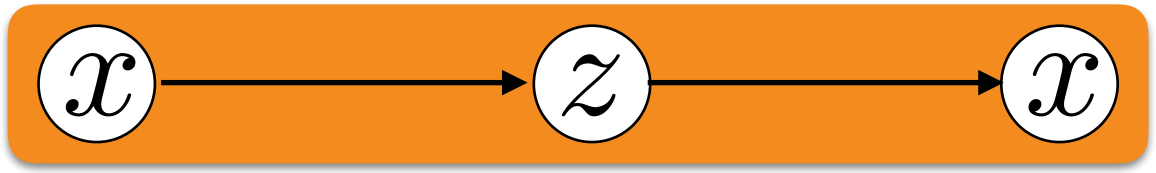
(1) we choose a specific x and look in its neighbourhood (to find z that most likely produced it)

(2) use $p(z \mid x)$ to sample the corresponding z

(3) evaluate $p(x \mid z)$ there

we really want to represent $p(x)$

$$p(x) = \int p(x \mid z)\, p(z)\, dz$$



Trick to do efficient MCMC:

(1) we choose a specific x and look in its neighbourhood (to find z that most likely produced it)

(2) use $q(z \mid x)$ to sample the corresponding z

(3) evaluate $p(x \mid z)$ there

minimise Kullback-Leibler to make ***q*** look like ***p***

$$\mathrm{KL}[q(z|x)\|p(z|x)] = \sum_z q(z|x)\,\log\frac{q(z|x)}{p(z|x)}$$

$$= E[\log q(z|x) - \log p(z|x)]$$

$$= E\left[\log q(z|x) - \log\frac{p(x|z)P(z)}{P(x)}\right]$$

$$= E[\log q(z|x) - \log p(x|z) - \log p(z) + \log p(x)]$$

$$\log p(x) - \mathrm{KL}[q(z|x)\|p(z|x)] = E[\log p(x|z) - (\log q(z|x) - \log p(z))]$$

$$= E[\log p(x|z)] - \mathrm{KL}[q(z|x)\|p(z)]$$

I need this

I can compute this

$$\text{argmax}_\theta \log p(x) - \text{KL}[q(z|x) \| p(z|x)] = E[\log p(x|z)] - \text{KL}[q(z|x) \| p(z)]$$

we can get our generative model…

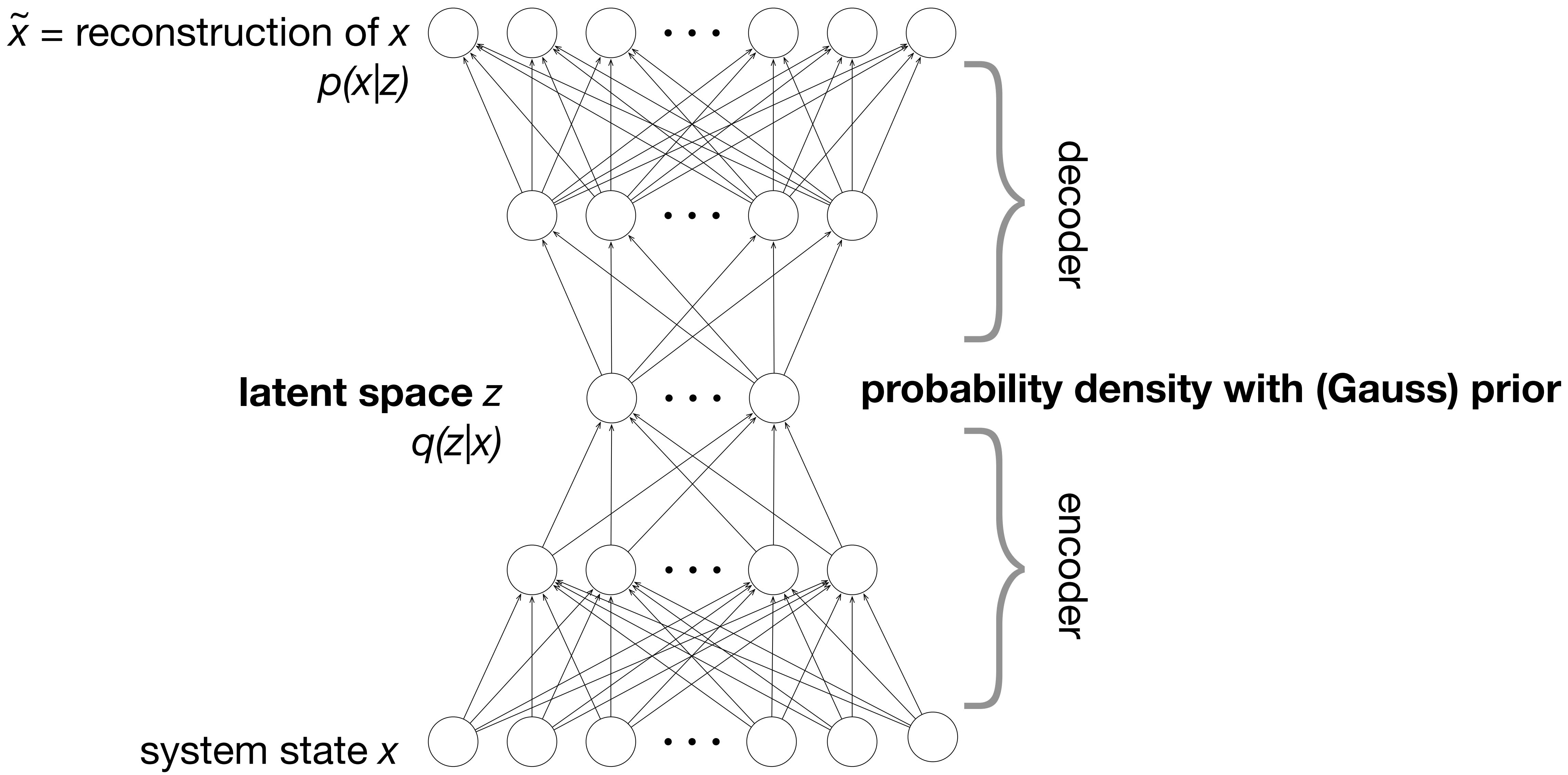
…while we want q to be close to p…

…by maximising the MLE for x given z…
(optimising the reconstruction by sampling)

…and please make z equal to the prior.

this is why I chose **argmax.ai** for our lab website

efficient computation as a neural network:
the Variational AutoEncoder



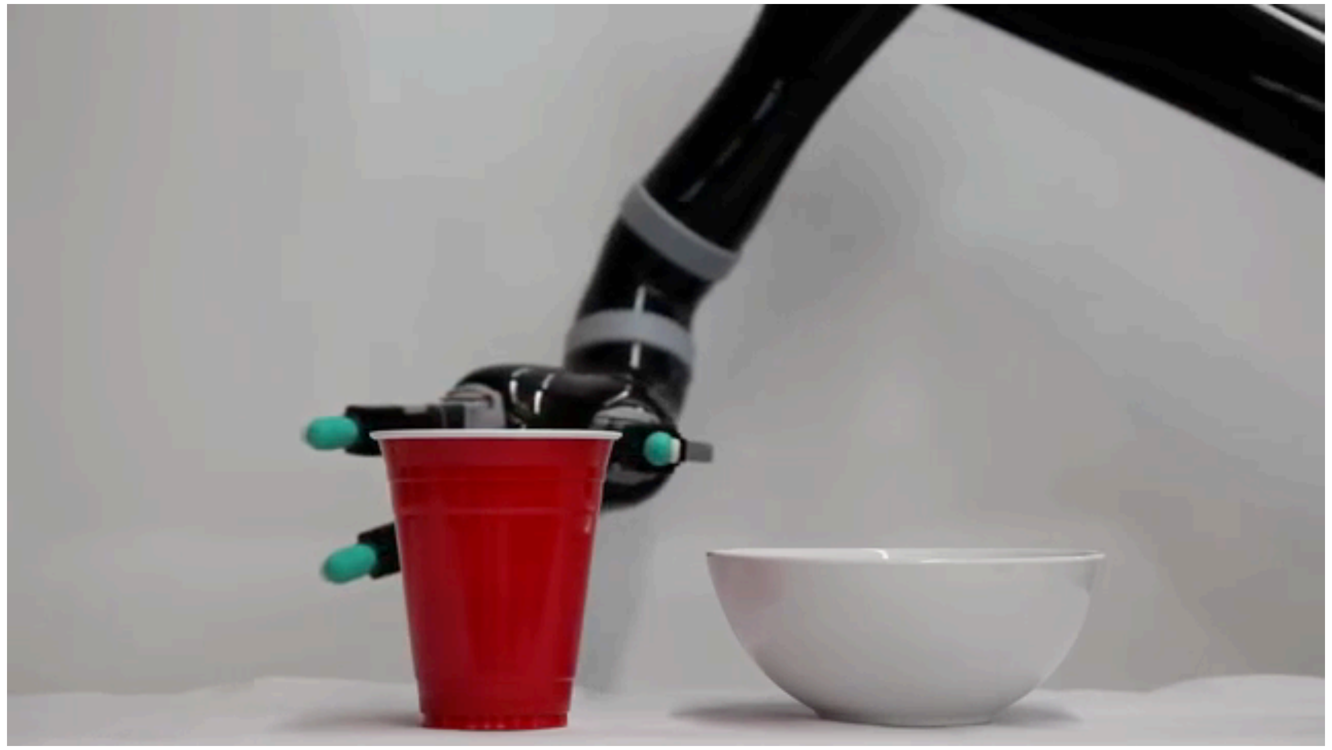
loss = reconstruction loss
+ KL[q(z|x) || prior]

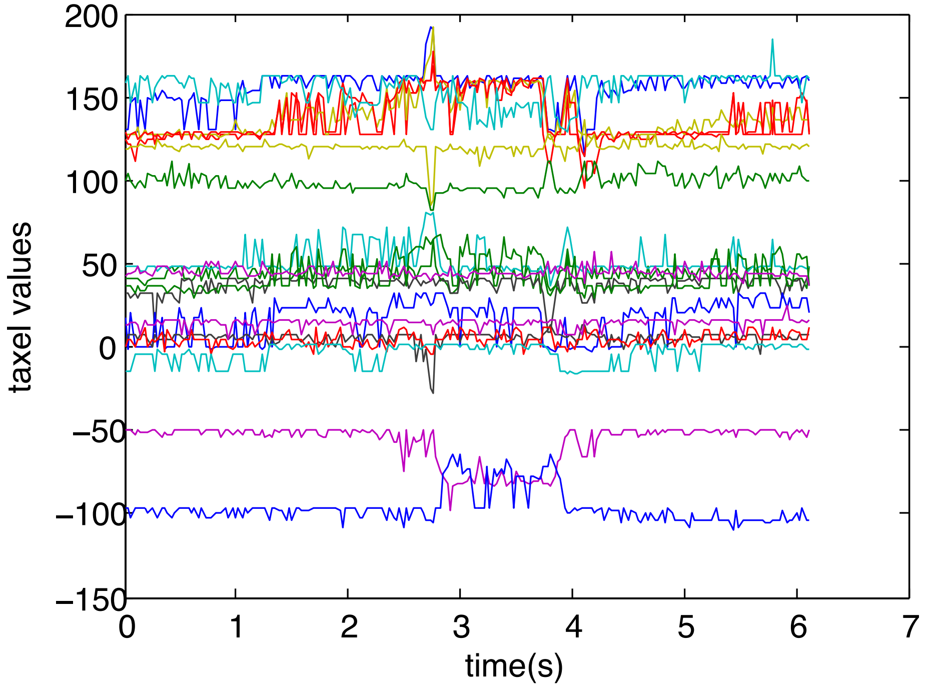
"nonlinear PCA"

Durk Kingma and Max Welling, 2013
Rezende, Mohamed & Wierstra, 2014

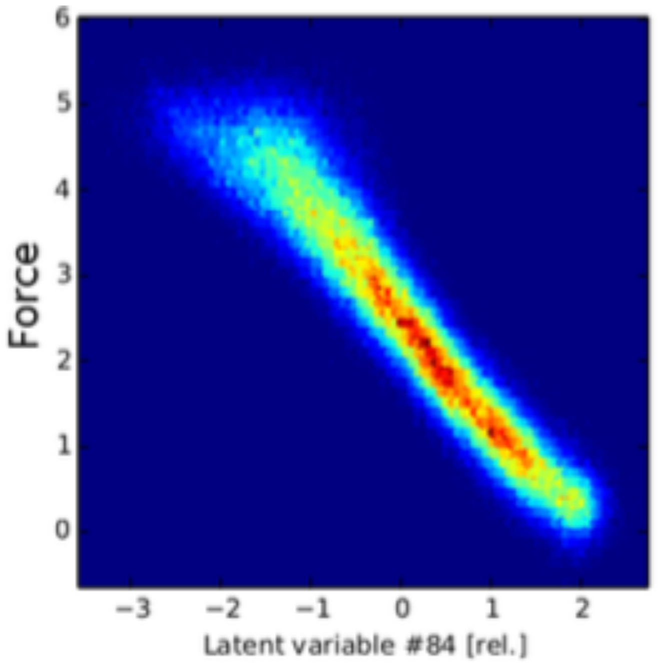
preprocessing sensor data with VAE—emerging properties

video: SynTouch, LLC



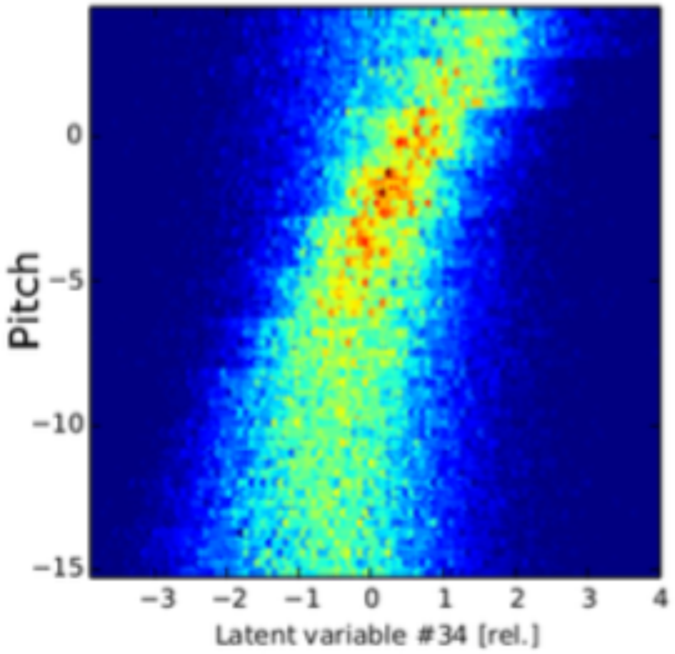


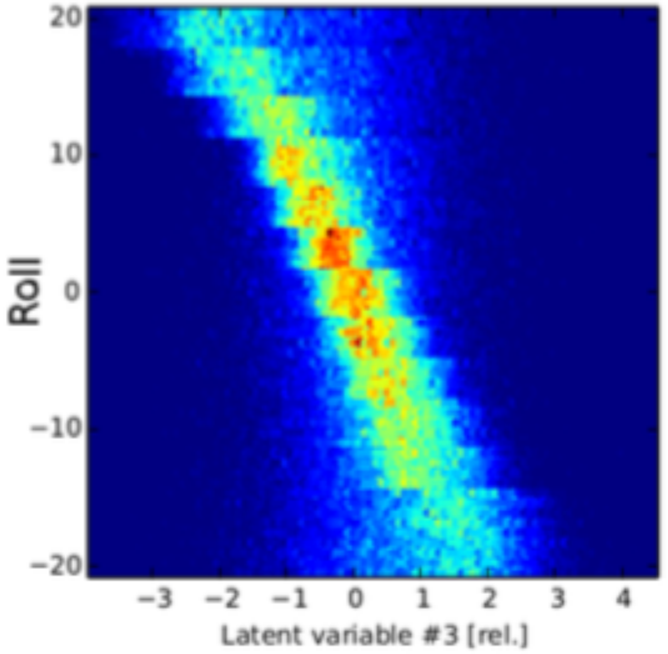
Maximilian Karl, Nutan Chen, Patrick van der Smagt (2014)



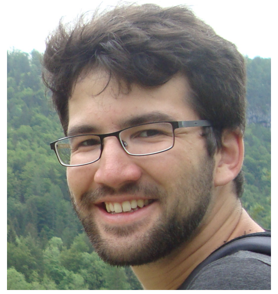
Maximilian Karl

Nutan Chen

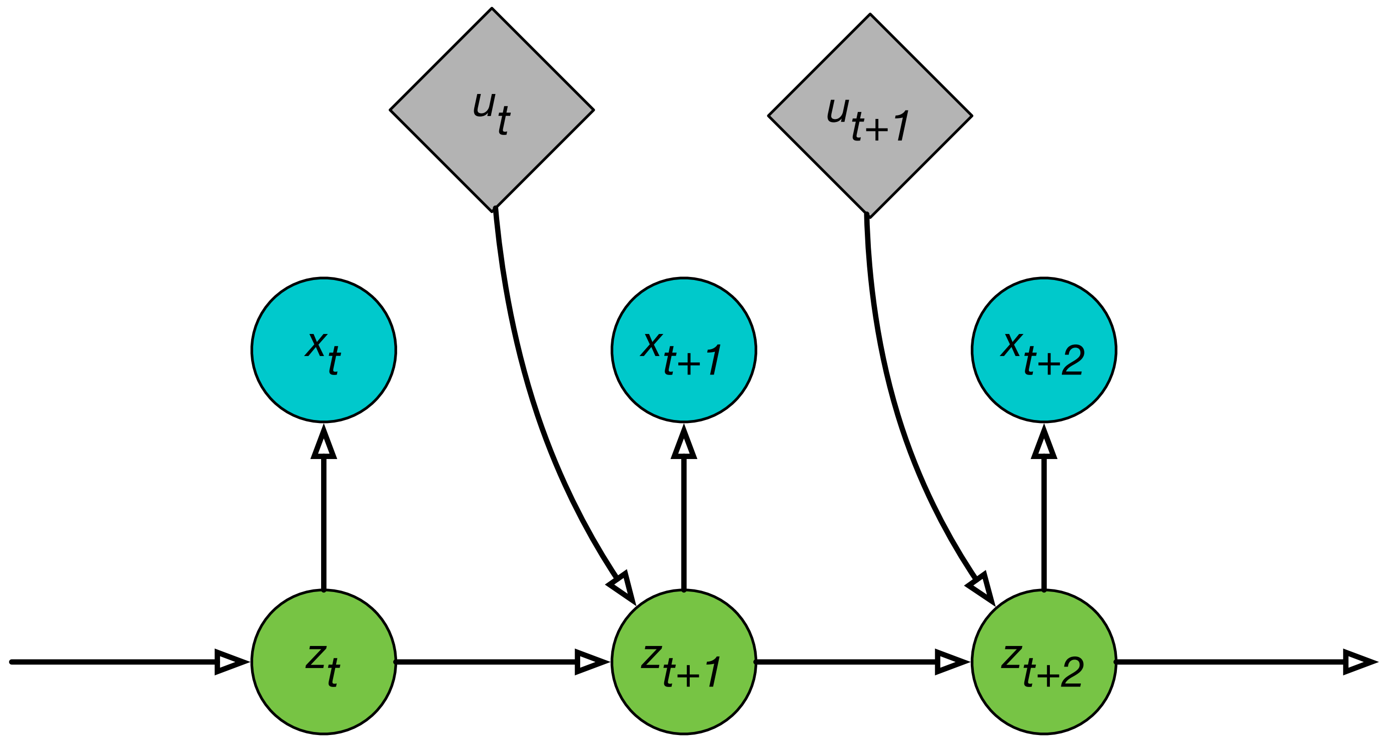




Deep Variational Bayes Filter


Maximilian Karl
Maximilian Sölch
Justin Bayer



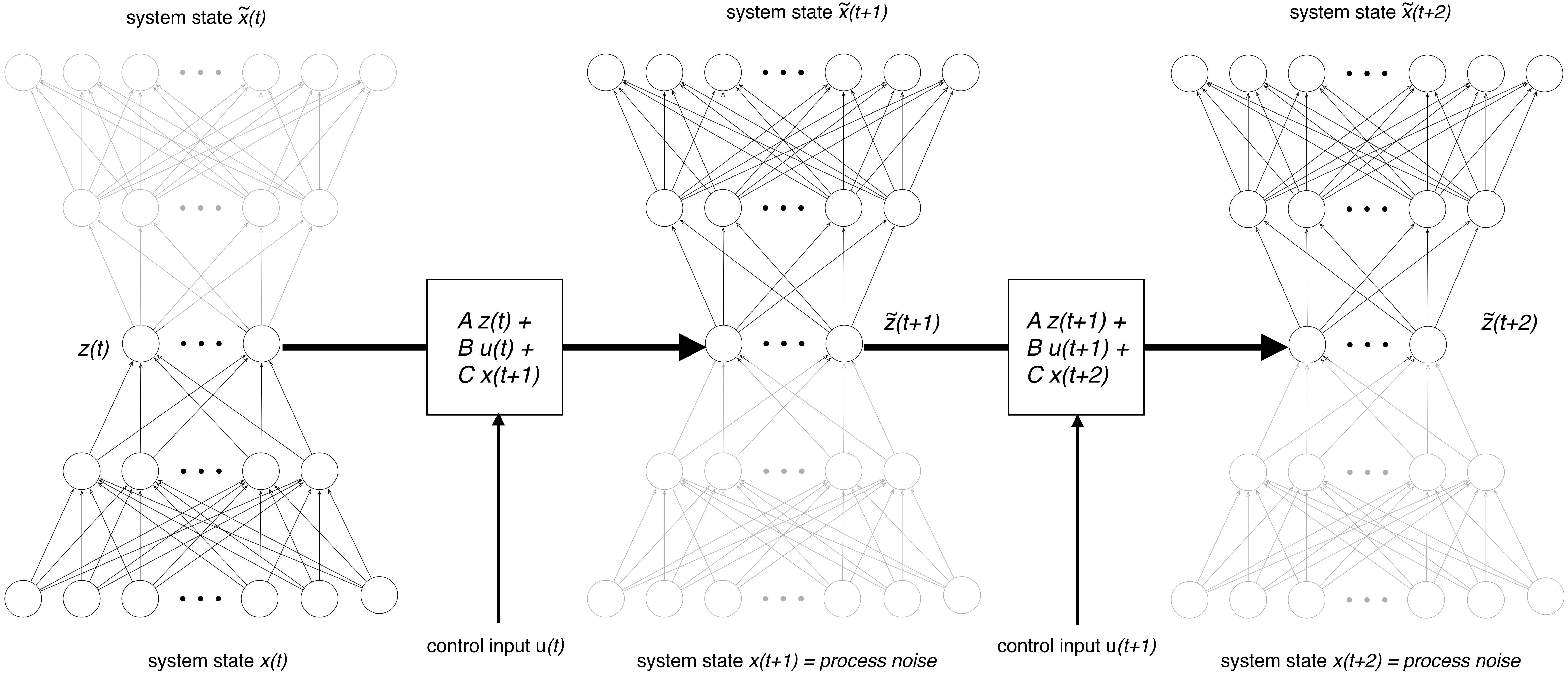
Graphical model assumes latent Markovian dynamics

i) Observations depend only on the current state,

ii) State depends only on the previous state and control signal,

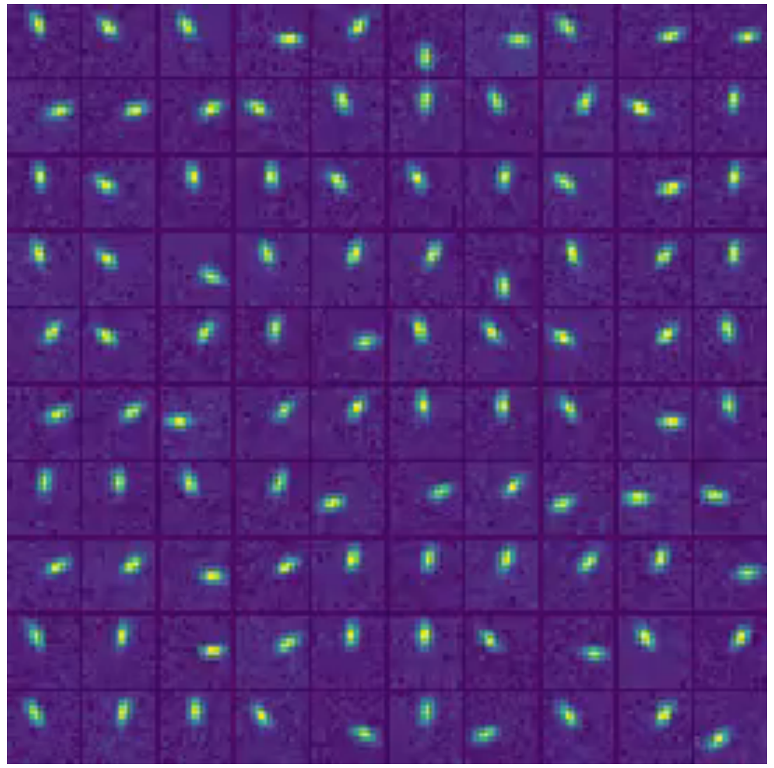
$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T} \mid \mathbf{u}_{1:T}) = \rho(\mathbf{z}_1) \prod_{t=1}^{T-1} p(\mathbf{z}_{t+1} \mid \mathbf{z}_t, \mathbf{u}_t) \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{z}_t)$$

Deep Variational Bayes Filtering:
filtering in latent space of a variational autoencoder

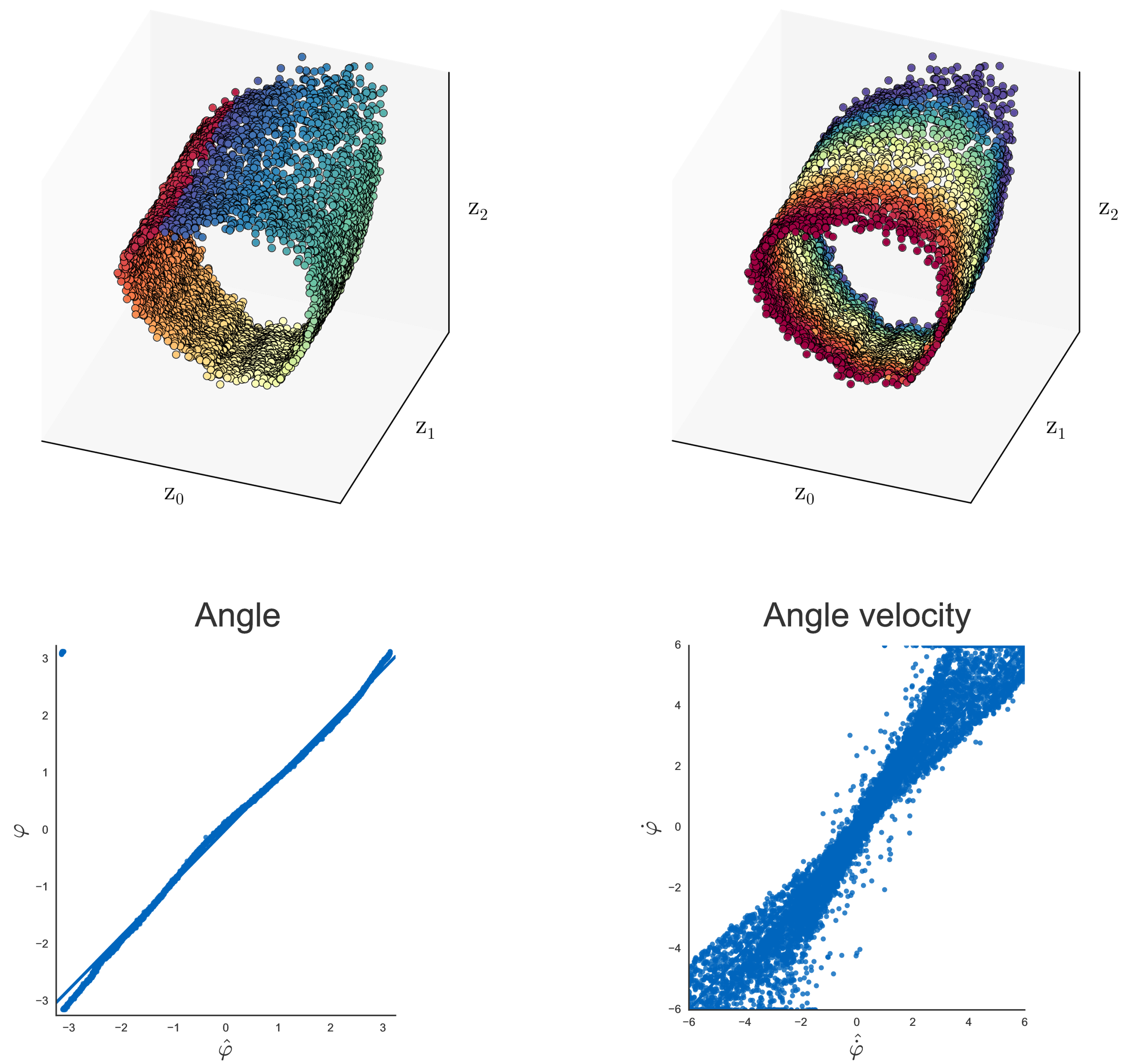


Karl & Soelch & Bayer & van der Smagt, ICLR 2017

Deep Variational Bayes Filter: example



$$ml^2\ddot{\varphi}(t) = -\mu\dot{\varphi}(t) + mgl\sin\varphi(t) + u(t)$$



transition model: z(t+1) = A z(t) + B u(t) + C x(t+1)

Formula E use case with Audi Motorsport

Philip
Becker

Audi Motorsport is interested in **optimal energy strategies**.
Knowing future battery temperature is key.

Approach:

Learn simulator of battery temperature given race conditions and control commands.

Use simulator to choose strategy that has best temperature for final performance.

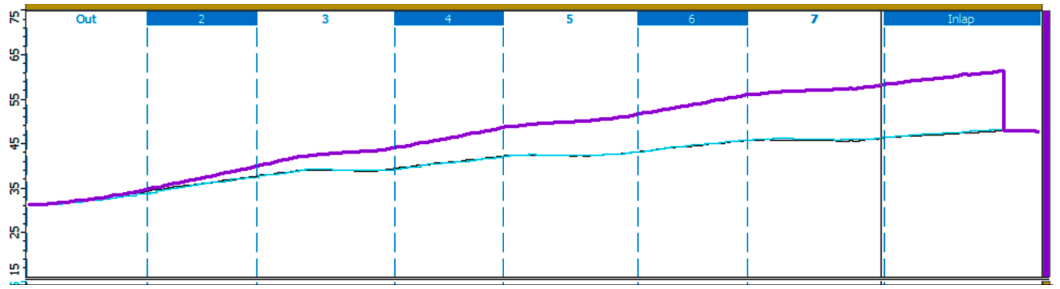
Project…

… initiated end of August,
… started a week later,
… deployed to hardware during test in November 2017,
… tested on car during race early December 2017.

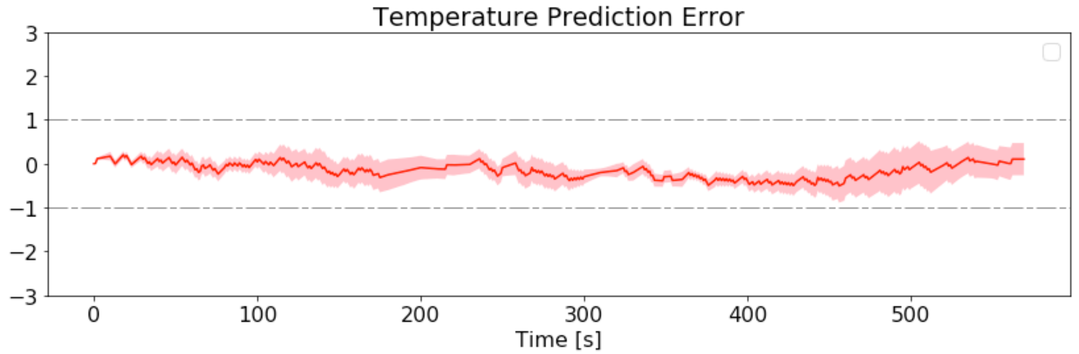
Results:

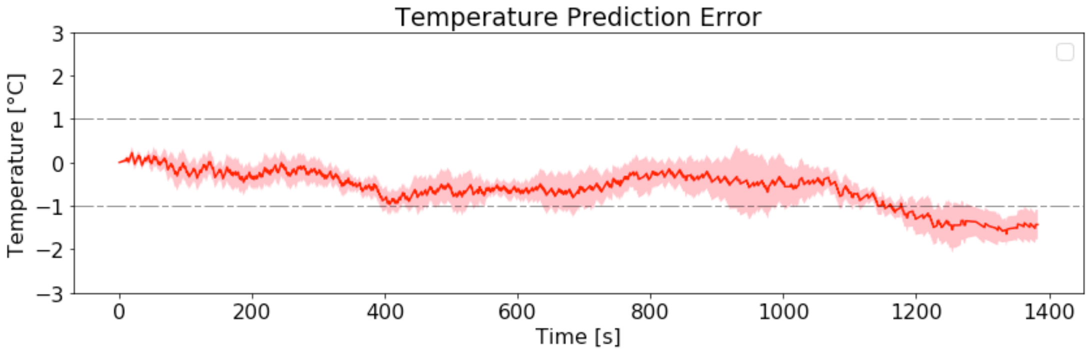
error < ±1 degree in 50% of the races

baseline



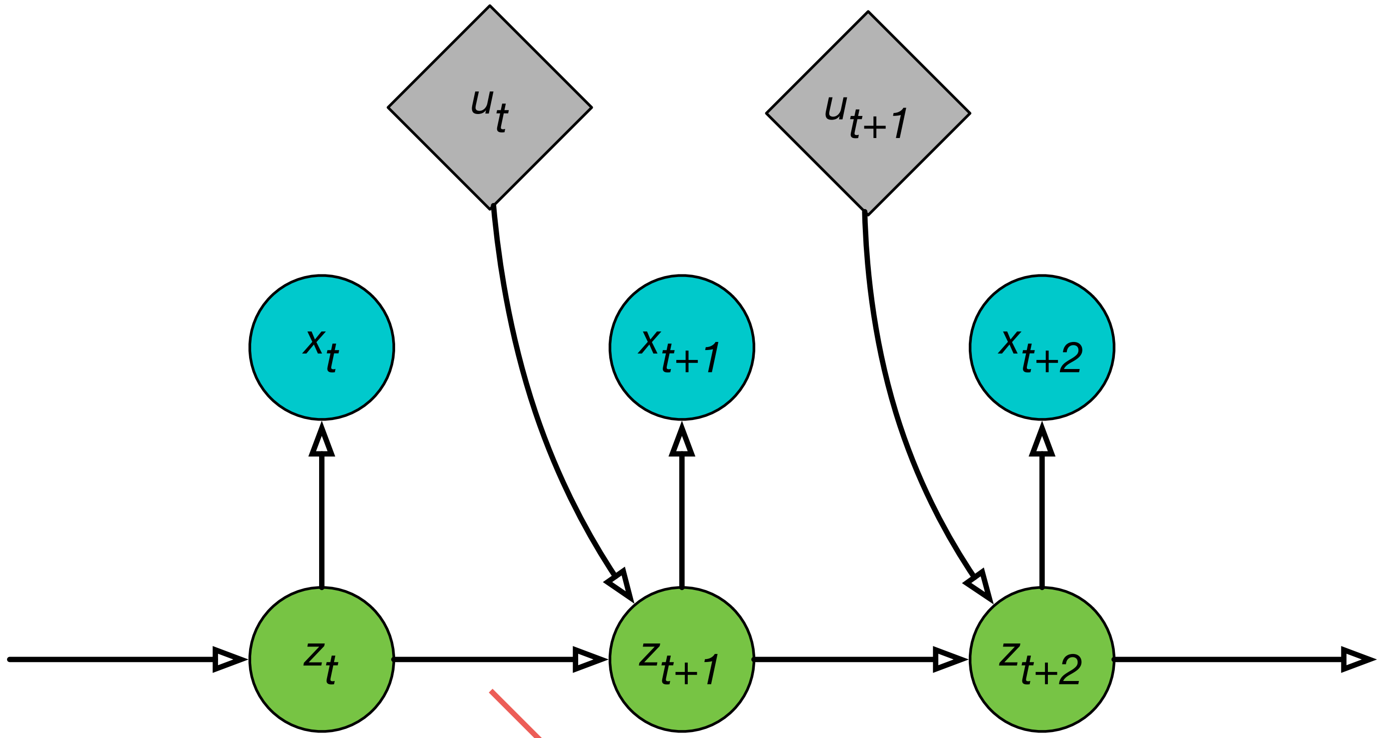
our method





Deep Variational Bayes Filter with DMP

Nutan Chen   Maximilian Karl

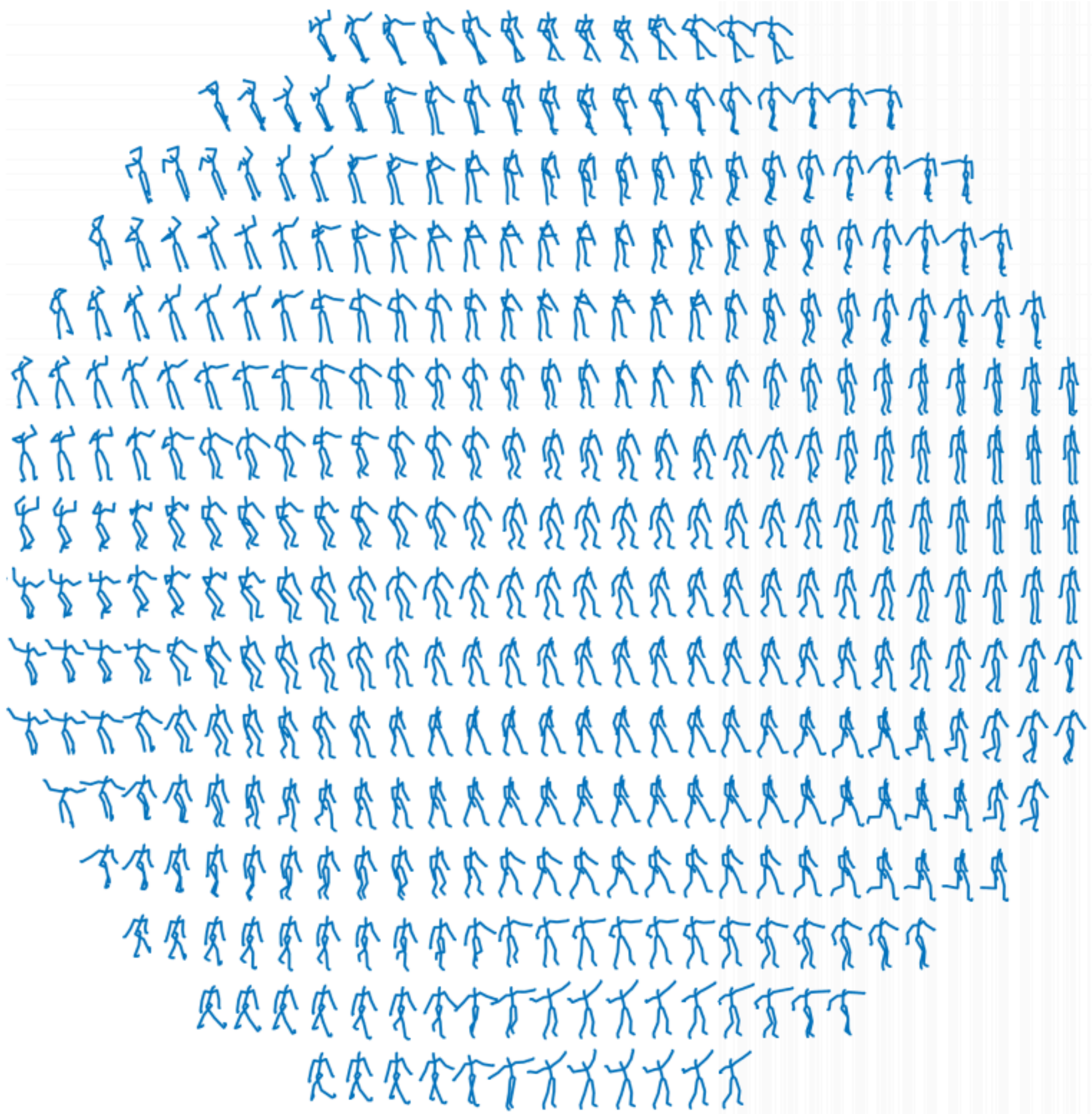


transition model:

$$\tau \ddot{\mathbf{z}}_{t+1} = \alpha(\beta_z(\mathbf{z}^{\text{goal}} - \mathbf{z}_t) - \dot{\mathbf{z}}_t) + \mathbf{f}_t + \epsilon$$

Deep Variational Bayes Filtering:

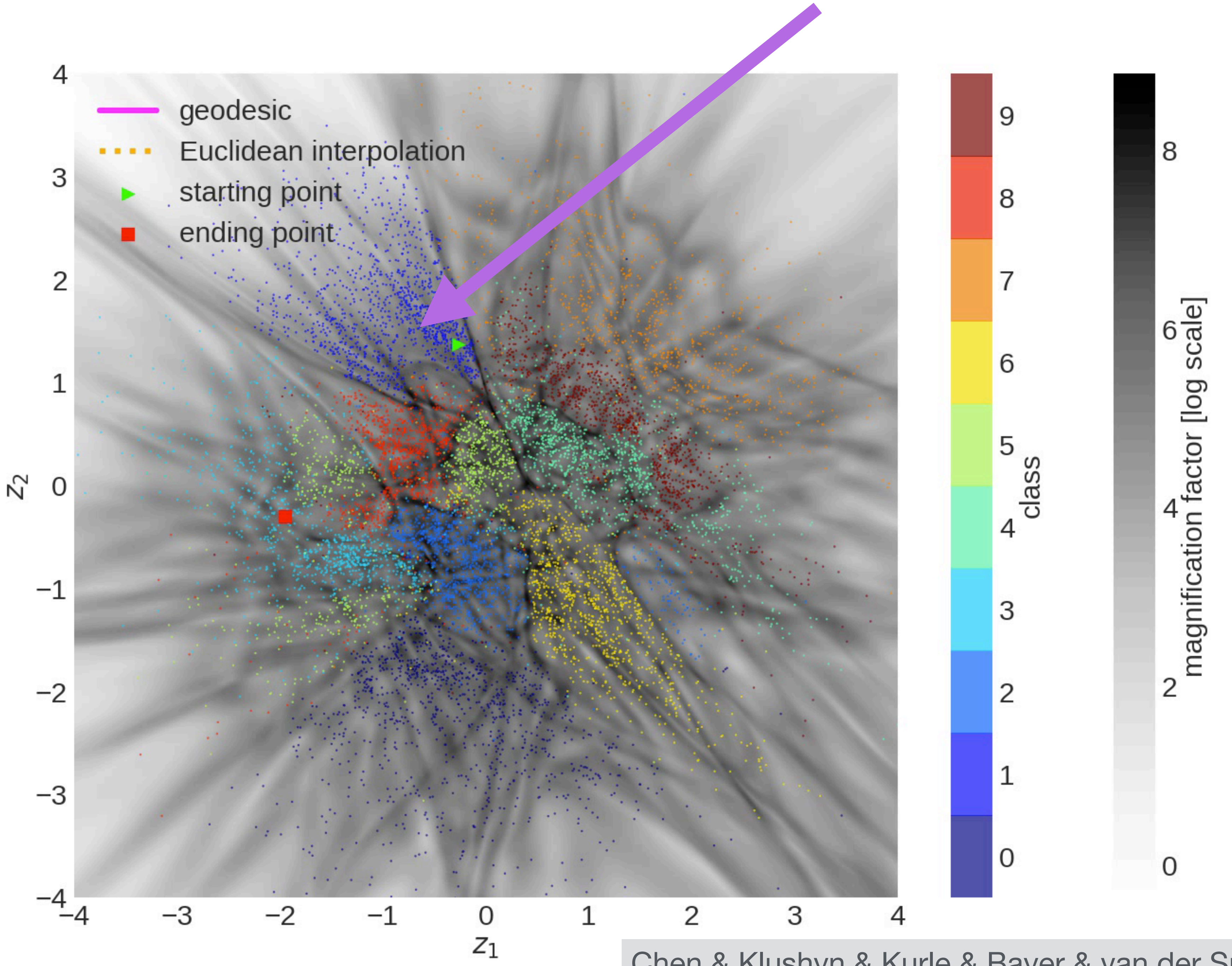
DMPs in latent space of a variational autoencoder



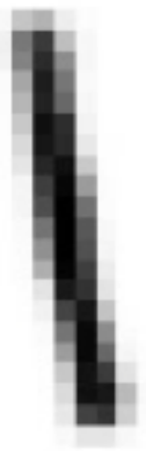
unsupervised

latent space *z(t)* is not straight!

latent space sampling #2: this is the optimal "shortest" path

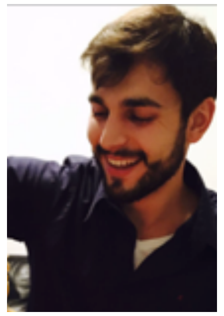


Chen & Klushyn & Kurle & Bayer & van der Smagt, 2018



how does Geodesics work?


Nutan Chen


Alexej Klushyn


Richard Kurle

length of γ

decoder NN

curve in $\mathbf{z}$

$$L(\gamma) := \int_0^1 \left\| \frac{\partial f(\gamma(t))}{\partial t} \right\| \mathrm{d}t = \int_0^1 \left\| \frac{\partial f(\gamma(t))}{\partial \gamma(t)} \frac{\partial \gamma(t)}{\partial t} \right\| \mathrm{d}t = \int_0^1 \left\| \mathbf{J} \frac{\partial \gamma(t)}{\partial t} \right\| \mathrm{d}t,$$

$$= \int_0^1 \sqrt{\langle \gamma'(t), \gamma'(t) \rangle_{\gamma(t)}} \mathrm{d}t = \int_0^1 \sqrt{\gamma'(t)^T \, \mathbf{G} \, \gamma'(t)} \mathrm{d}t$$

$= \mathbf{J}^T \mathbf{J}$

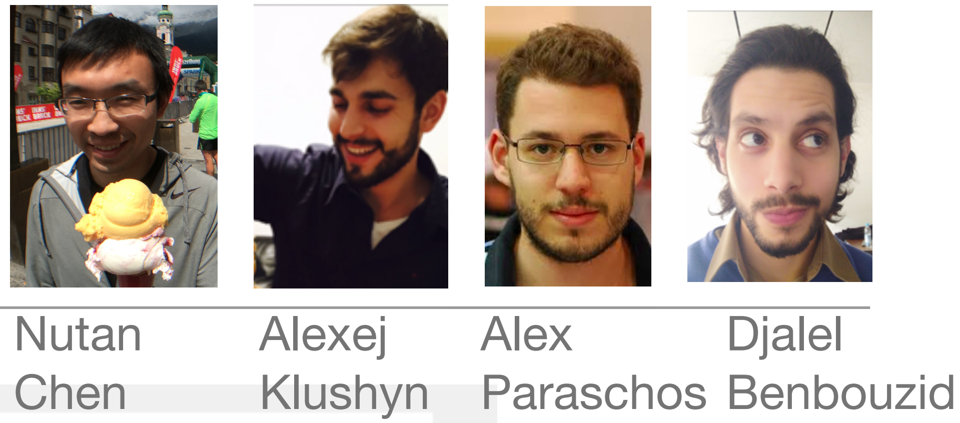
neural network

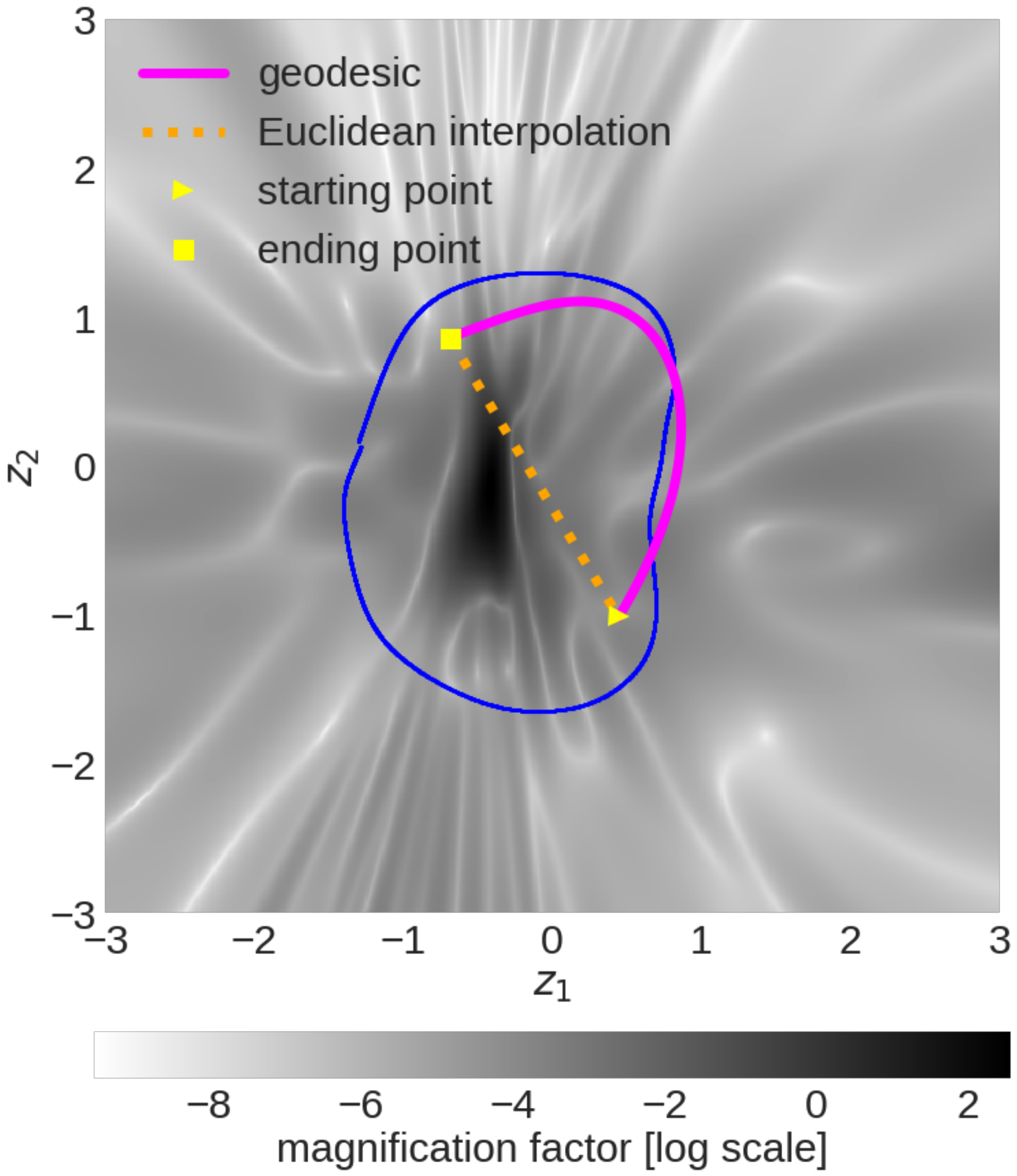
$\mathbb{R} \to \mathbb{R}^{N_z}$

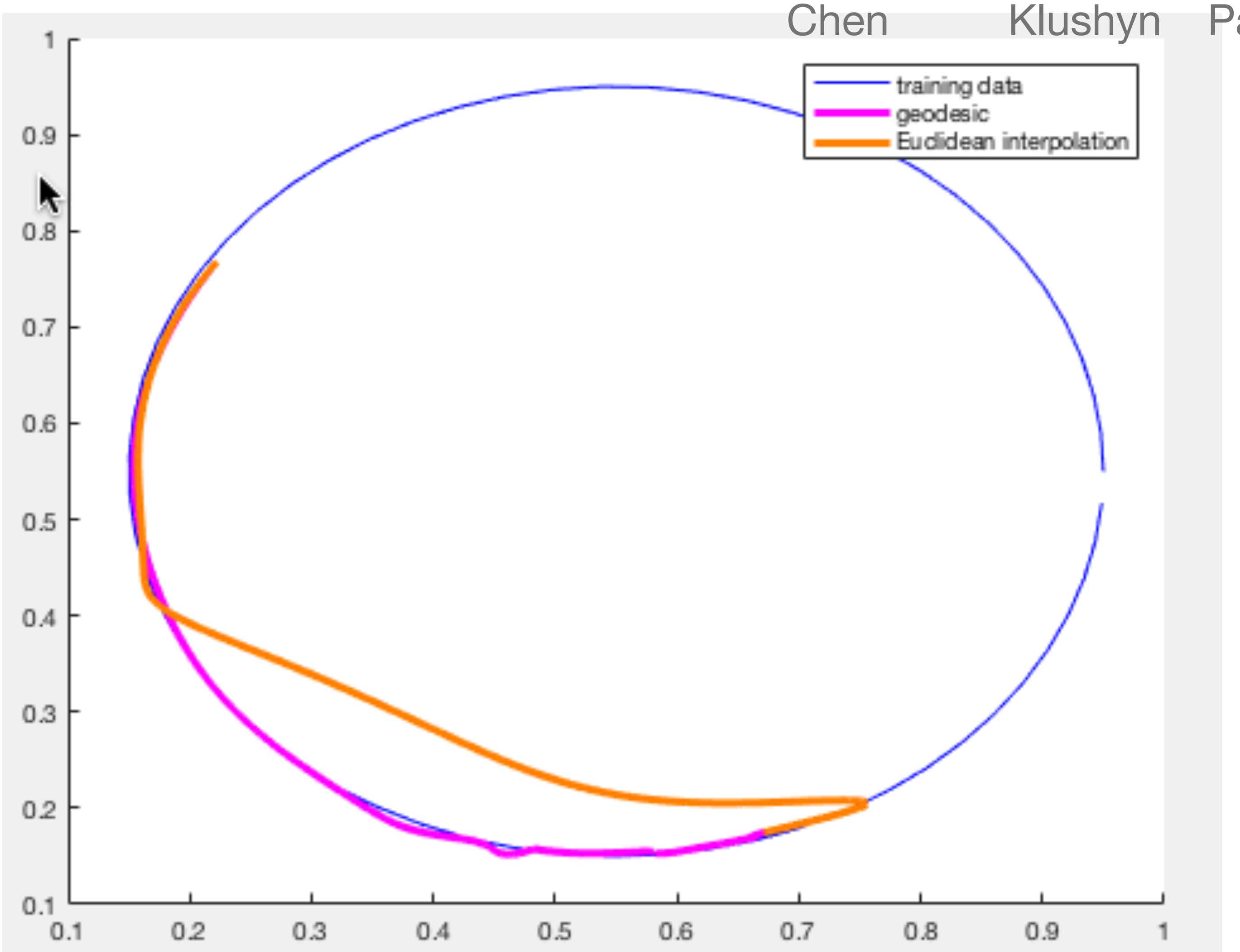
$$\min_{\omega} L(g_\omega(t))$$
$$s.t. \; g_\omega(0) = \mathbf{z}_0, \; g_\omega(1) = \mathbf{z}_1.$$

$$MF := \sqrt{\det \mathbf{G}}$$

...on a 6-DoF robot arm...







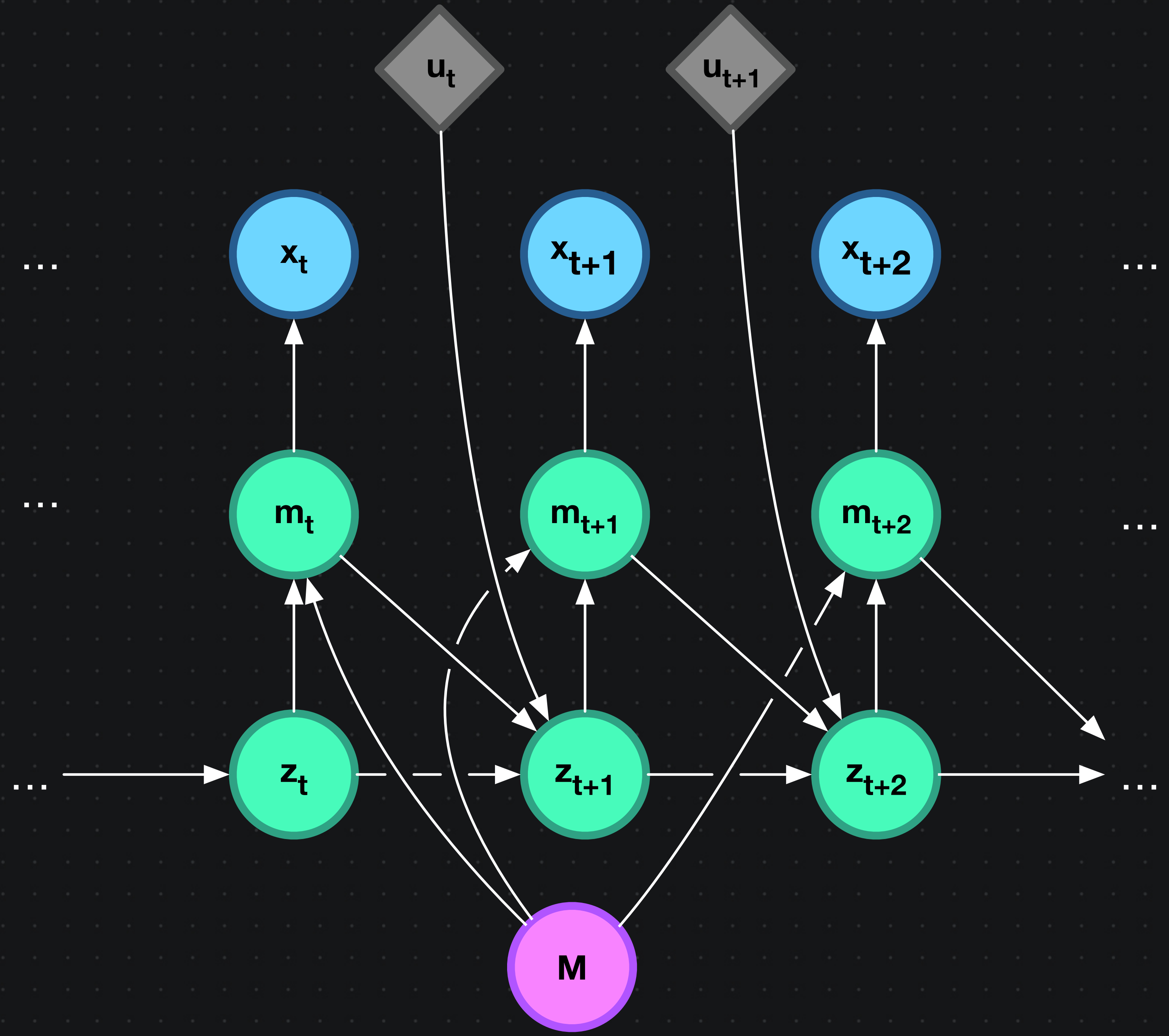
Active Learning based on Data Uncertainty and Model Sensitivity

Nutan Chen, Alexej Klushyn, Alexandros Paraschos, Djalel Benbouzid, Patrick van der Smagt

AI research, Data:Lab, Volkswagen Group

Deep Variational Bayes Filter with a map

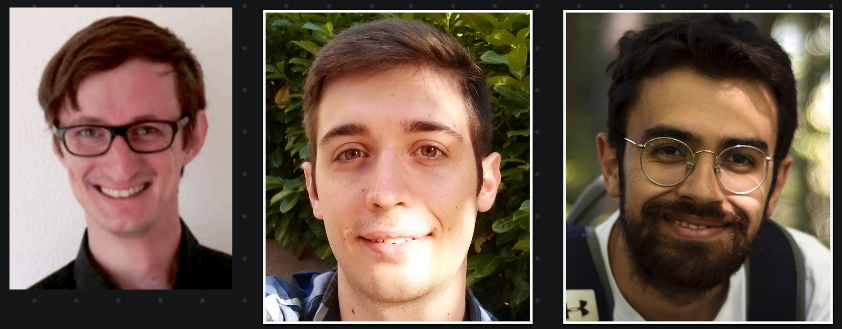
ARGMAX.ai
VOLKSWAGEN GROUP AI RESEARCH



Graphical model assumes global Map

iii) Observations are extracted from map through attention model based on current location,

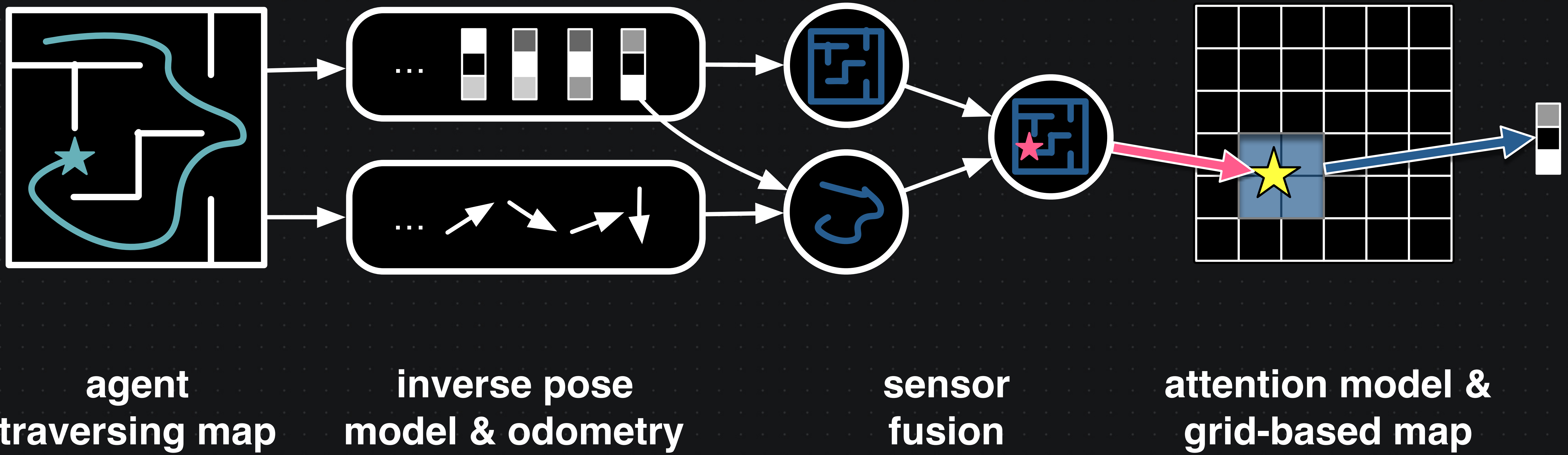
iv) Latent state is identified with location.



Justin Bayer
Atanas Mirchev
Baris Kayalibay

End-to-End SLAM

Our approach is data-driven: deep neural networks, attention models and variational inference.

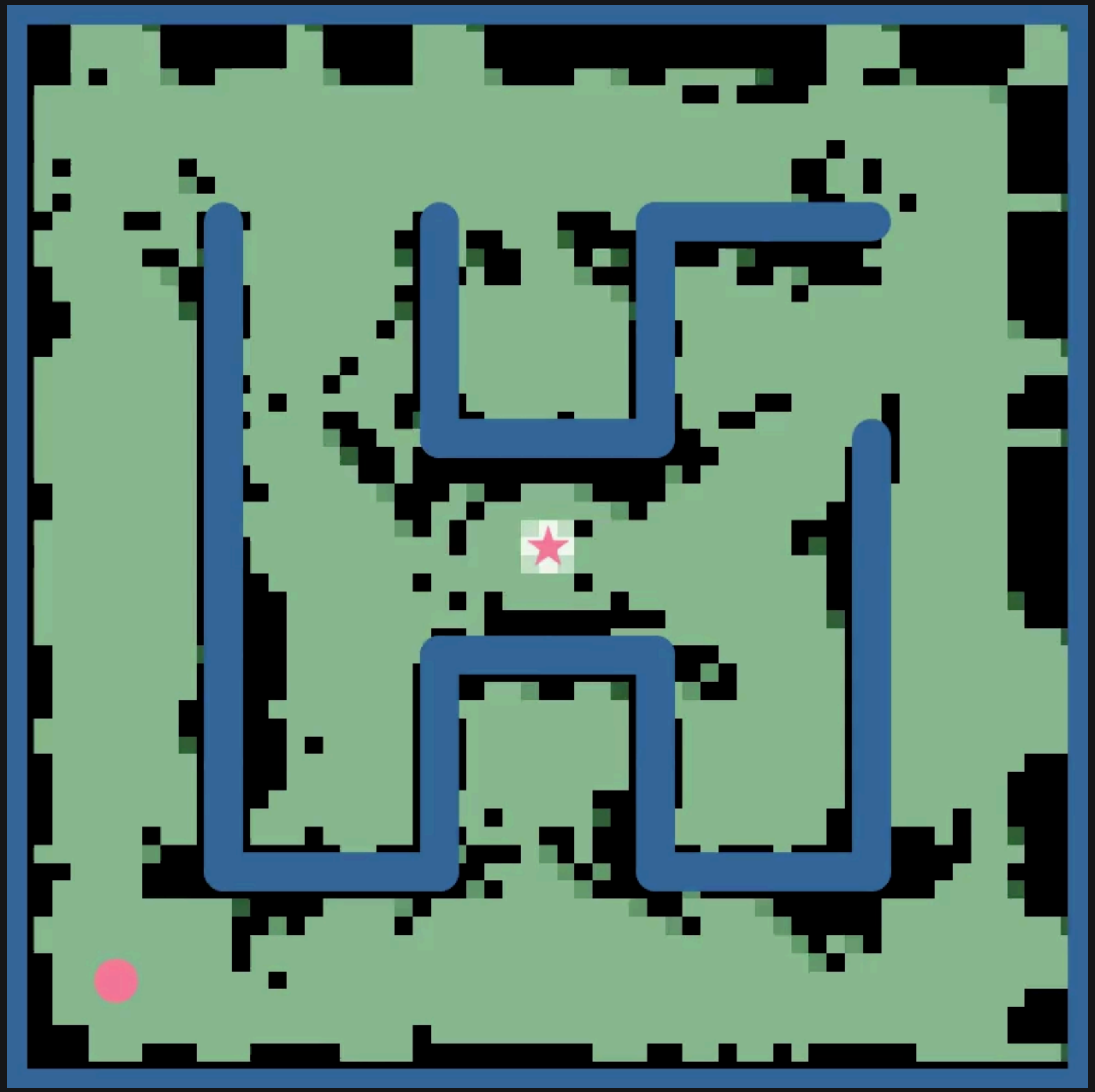


optimal control of a learnt model

mapping, localisation and planning—all in the same model.

Navigation via **optimal control:** The cost at the goal is 0 and -1 everywhere else.

Optimisation is performed in a **learned model** and executed only after planning has finished.

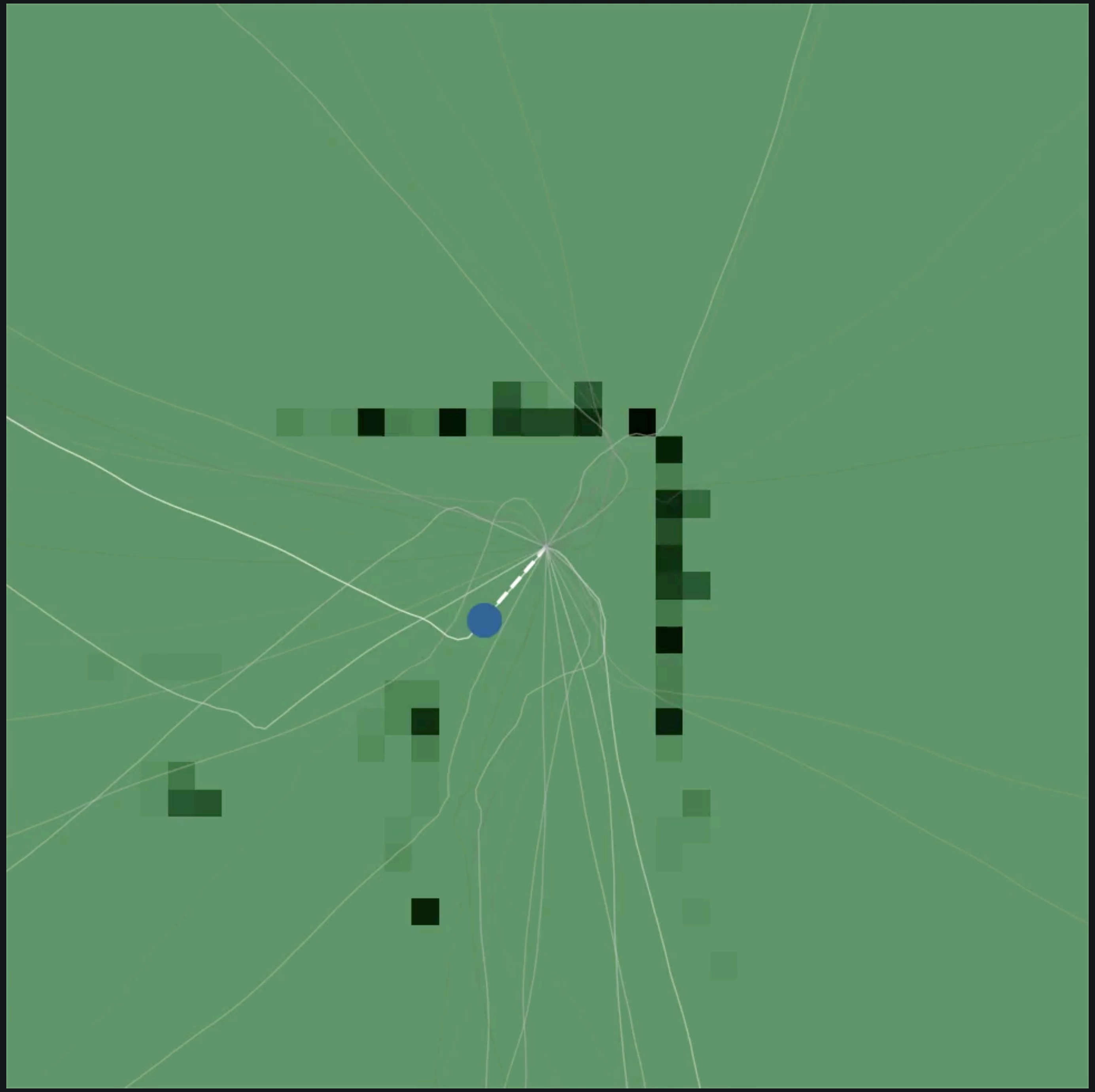


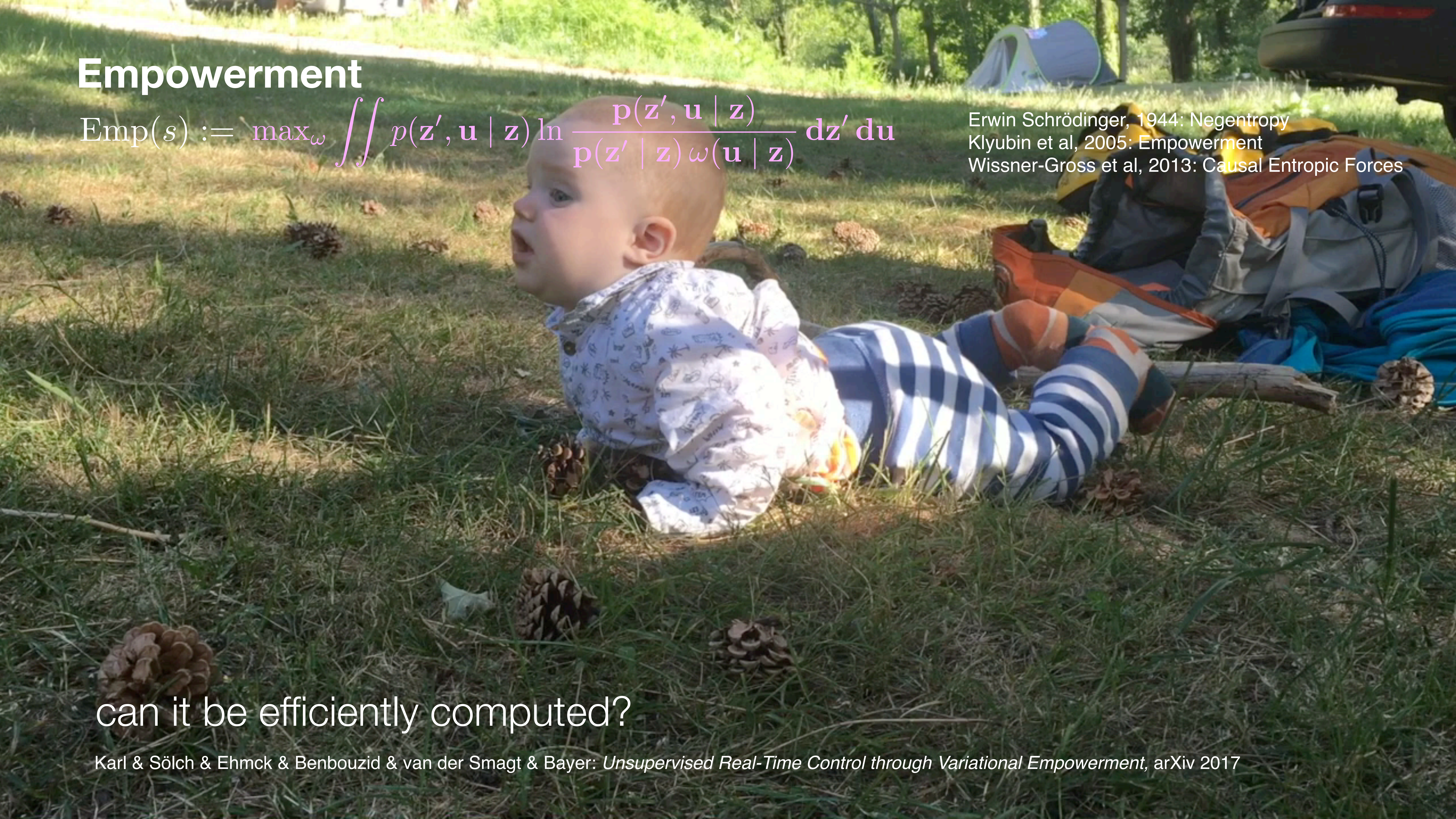
exploration — maximise expected surprise

The Bayesian nature of the model allows a **principled quantification of uncertainty.**

We can estimate how good the model knows certain regions of its environment.

Optimal control drives the agent into unexplored regions.





Empowerment

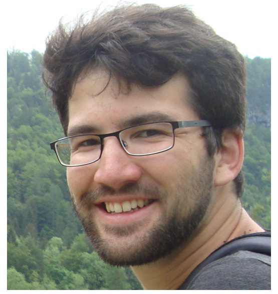
$$\mathrm{Emp}(s) := \max_\omega \iint p(\mathbf{z}', \mathbf{u} \mid \mathbf{z}) \ln \frac{\mathbf{p}(\mathbf{z}', \mathbf{u} \mid \mathbf{z})}{\mathbf{p}(\mathbf{z}' \mid \mathbf{z})\, \omega(\mathbf{u} \mid \mathbf{z})} \, \mathbf{dz}' \, \mathbf{du}$$

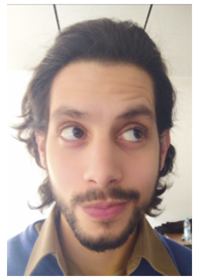
Erwin Schrödinger, 1944: Negentropy
Klyubin et al, 2005: Empowerment
Wissner-Gross et al, 2013: Causal Entropic Forces

can it be efficiently computed?

Karl & Sölch & Ehmck & Benbouzid & van der Smagt & Bayer: *Unsupervised Real-Time Control through Variational Empowerment,* arXiv 2017

how is efficient empowerment computed?




Maximilian Karl | Maximilian Sölch | Philip Becker | Djalel Benbouzid | Justin Bayer

empowerment is the channel capacity between action $\mathbf{u}_t$ and the following state $\mathbf{z}_{t+1}$

$$\mathcal{E}(\mathbf{z}) = \max_{\omega} \mathcal{I}(\mathbf{z}', \mathbf{u} \mid \mathbf{z})$$

$$:= \mathrm{KL}(p(\mathbf{z}', \mathbf{u} \mid \mathbf{z}) \,||\, p(\mathbf{z}' \mid \mathbf{z})\, \omega(\mathbf{u} \mid \mathbf{z}))$$

$$= \iint p(\mathbf{z}', \mathbf{u} \mid \mathbf{z}) \ln \frac{p(\mathbf{z}', \mathbf{u} \mid \mathbf{z})}{p(\mathbf{z}' \mid \mathbf{z})\, \omega(\mathbf{u} \mid \mathbf{z})} \,\mathrm{d}\mathbf{z}'\mathrm{d}\mathbf{u}$$

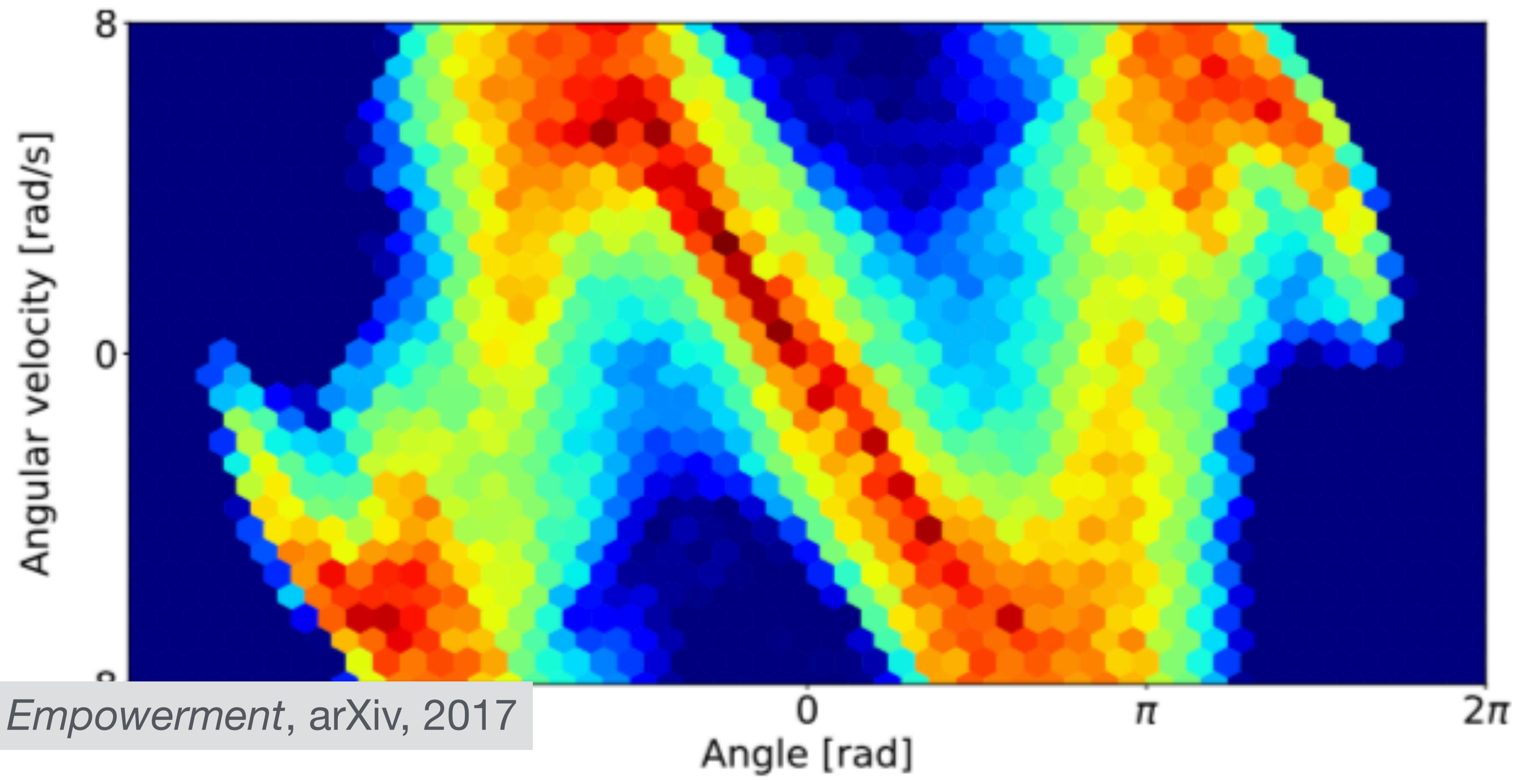
looking for the best state where each action has a meaningful consequence

intractable
(computed for all actions)

approximate with lower bound $\mathcal{I}(\mathbf{z}', \mathbf{u} \mid \mathbf{z}) \geq \iint p(\mathbf{z}', \mathbf{u} \mid \mathbf{z}) \ln \frac{q(\mathbf{u} \mid \mathbf{z}', \mathbf{z})}{\omega(\mathbf{u} \mid \mathbf{z})} \,\mathrm{d}\mathbf{z}'\mathrm{d}\mathbf{u} \; =$

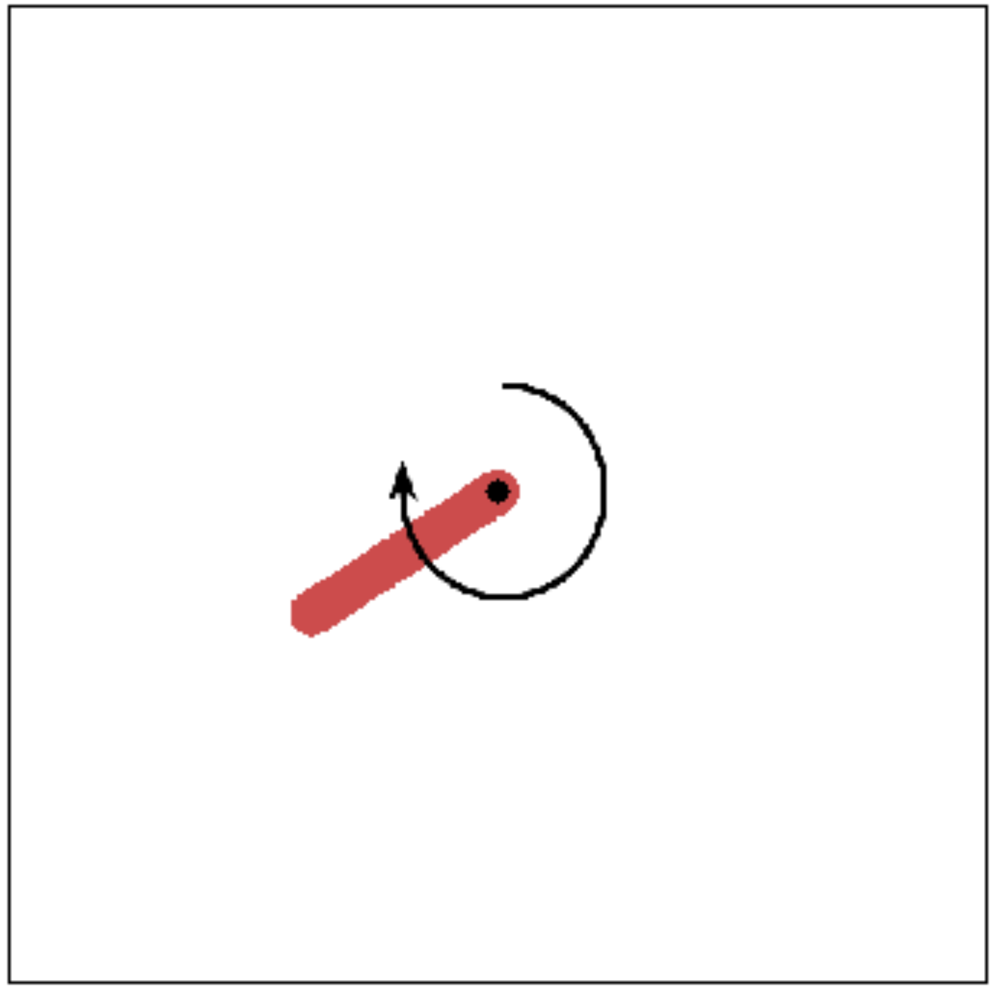
$$\mathcal{I} - \hat{\mathcal{I}} = \mathbb{E}_{\mathbf{z}' \sim p(\mathbf{z}' \mid \mathbf{z})}[\mathrm{KL}(p(\mathbf{u} \mid \mathbf{z}', \mathbf{z}) \,||\, ($$

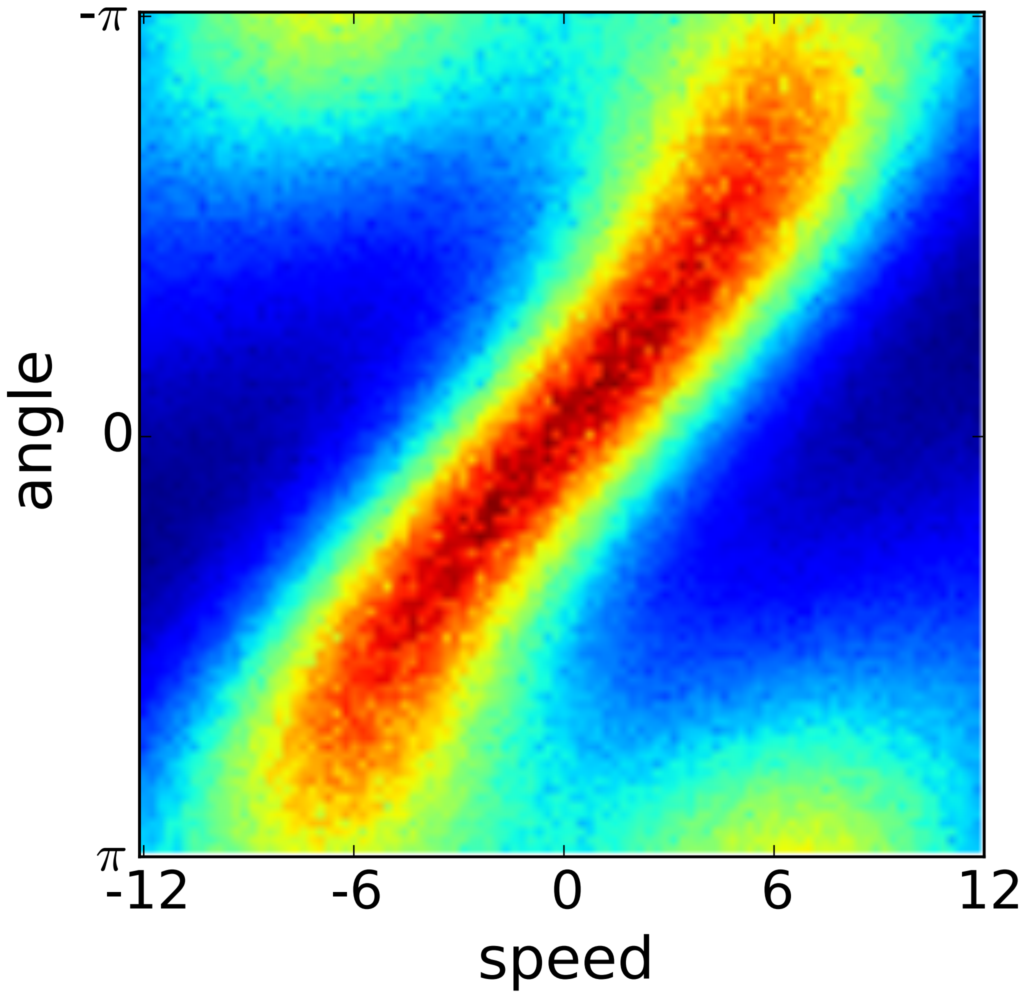
plan



Karl & ... & van der Smagt & Bayer: *Unsupervised Real-Time Control through Variational Empowerment*, arXiv, 2017

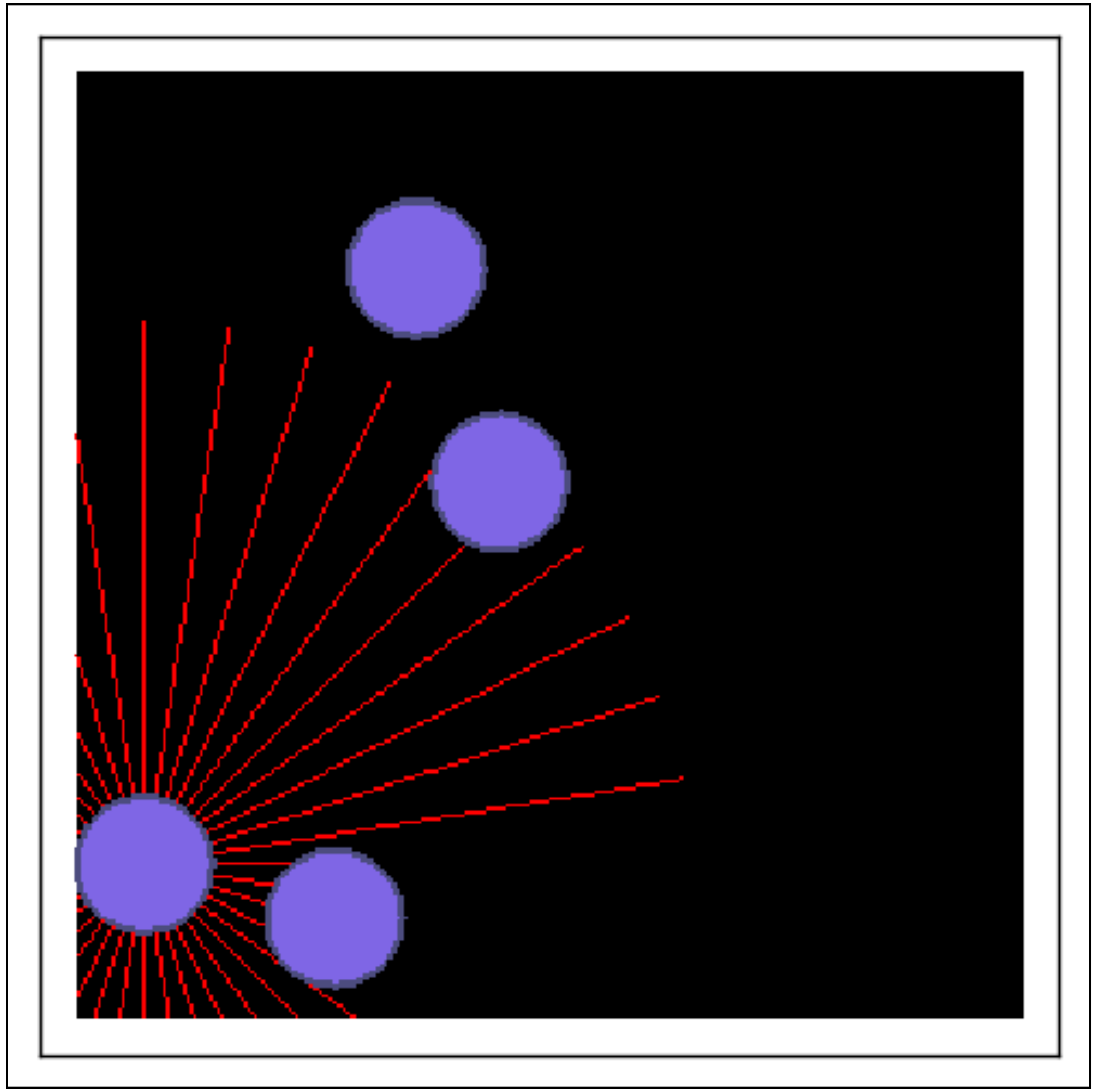
empowerment on a pendulum

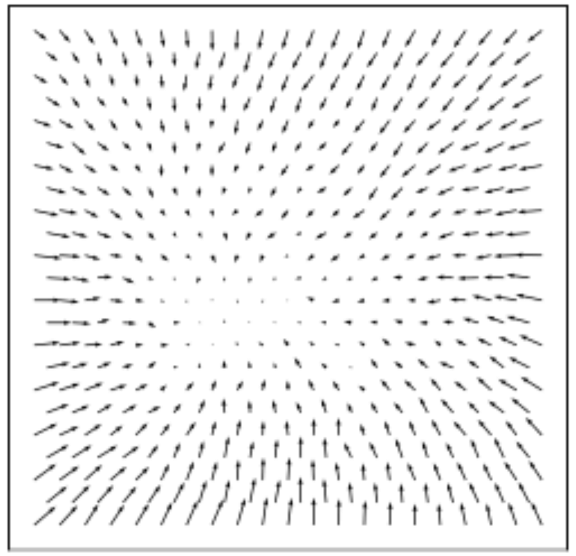




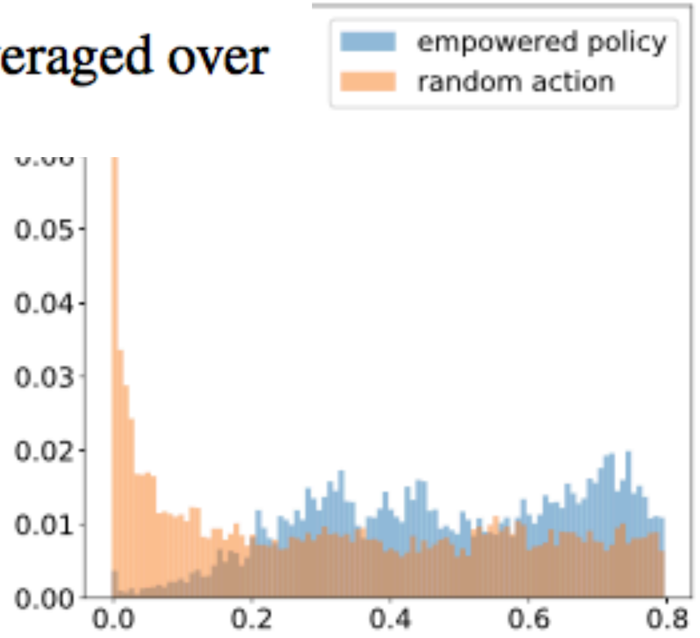
unsupervised

independent balls with 40-dimensional lidar sensors

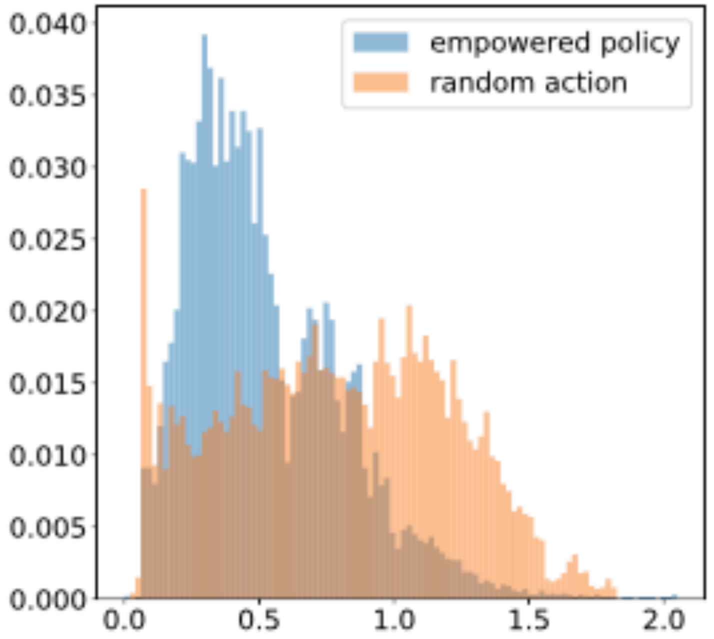




(b) Policy averaged over all balls.



(c) Distance between ball and closest wall.

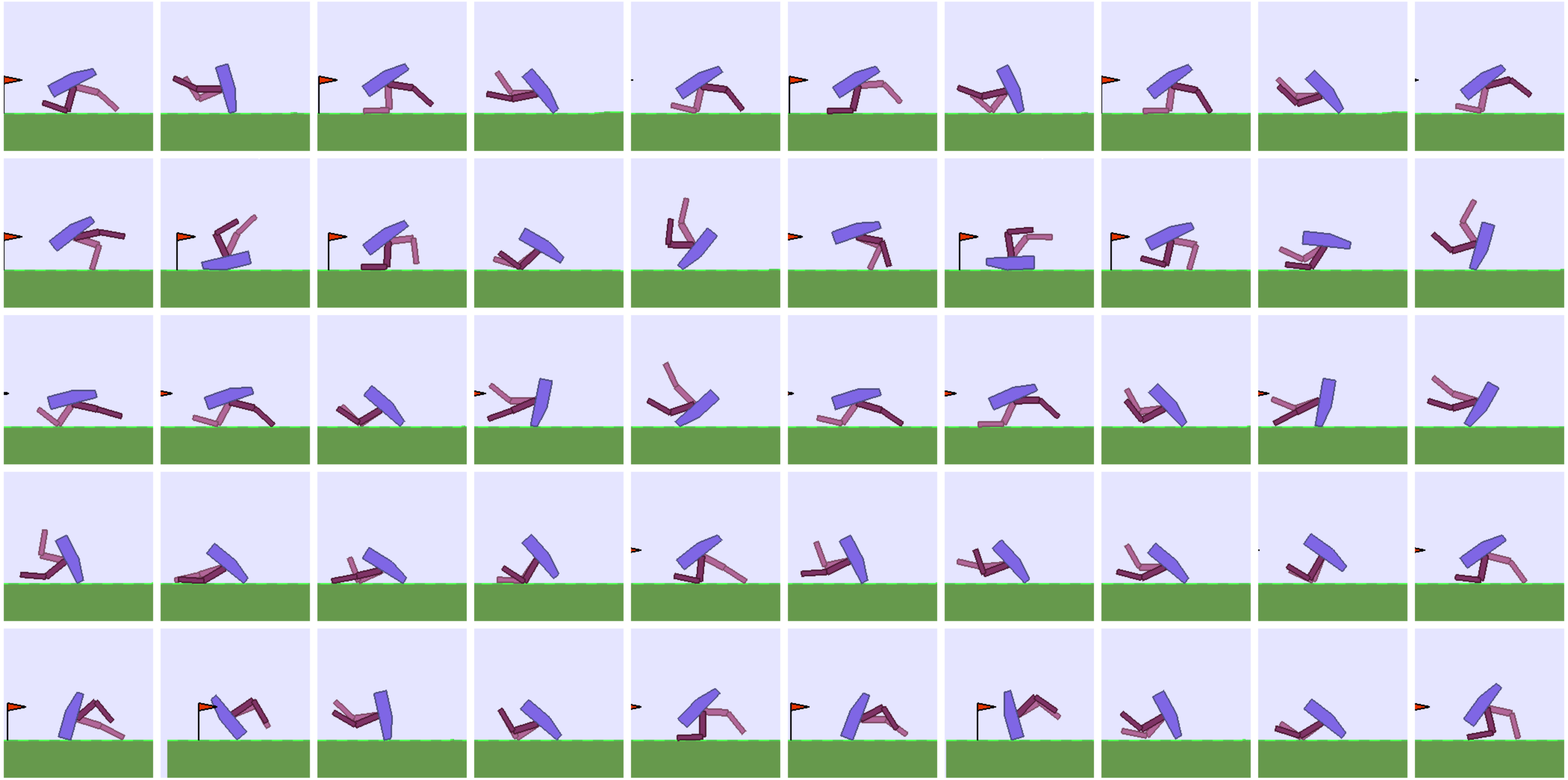


(d) Distance between two balls.

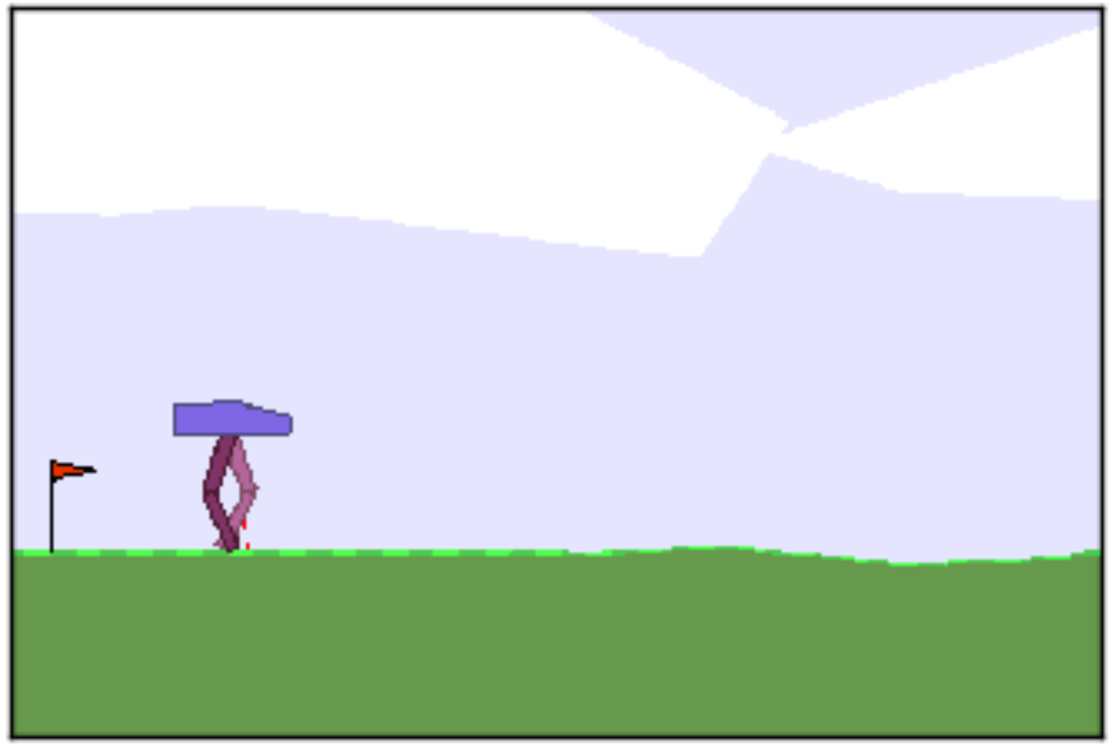
Karl & ... & van der Smagt & Bayer: *Unsupervised Real-Time Control through Variational Empowerment*, arXiv, 2017

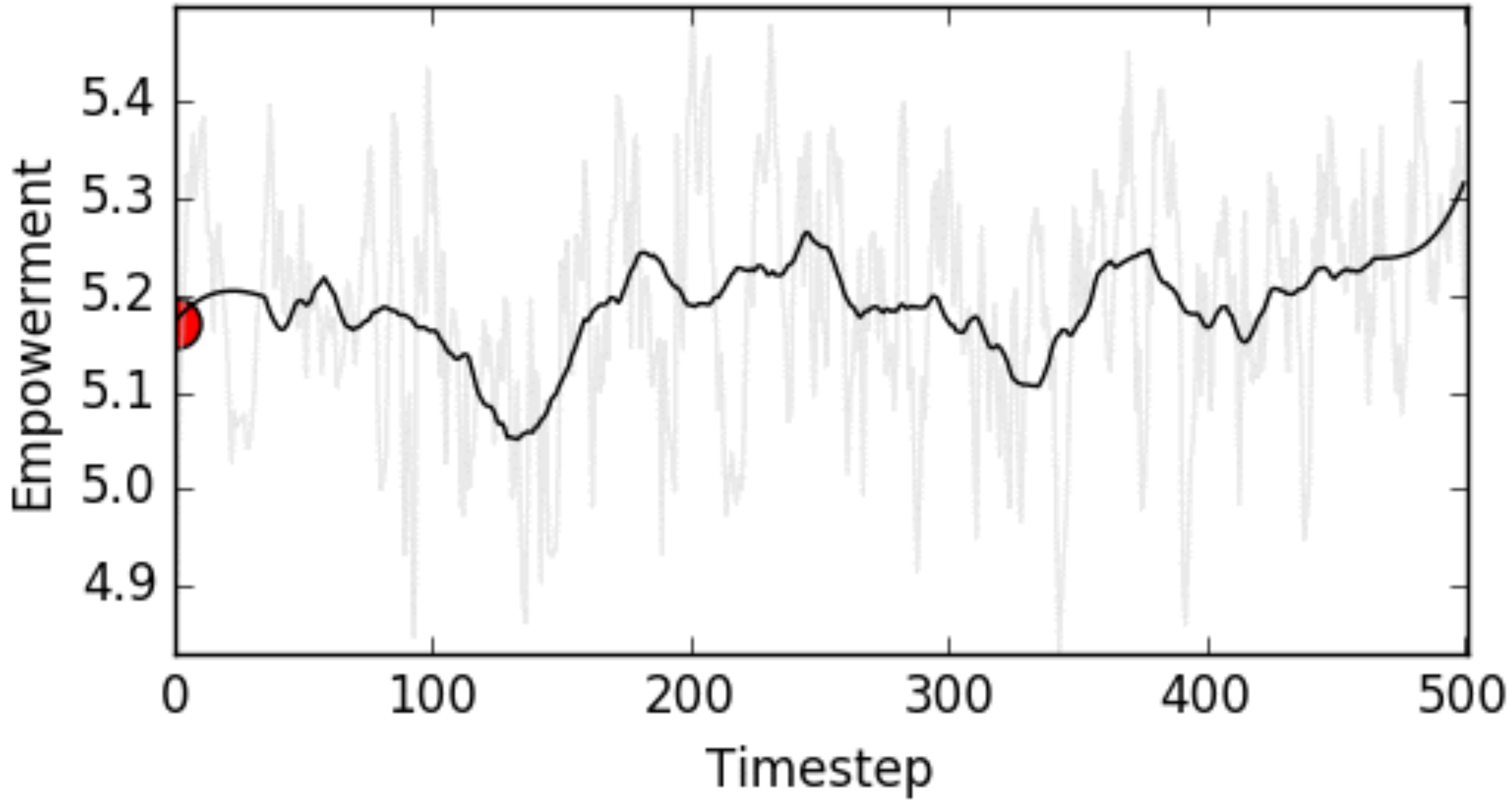
control through DVBF:

exploration

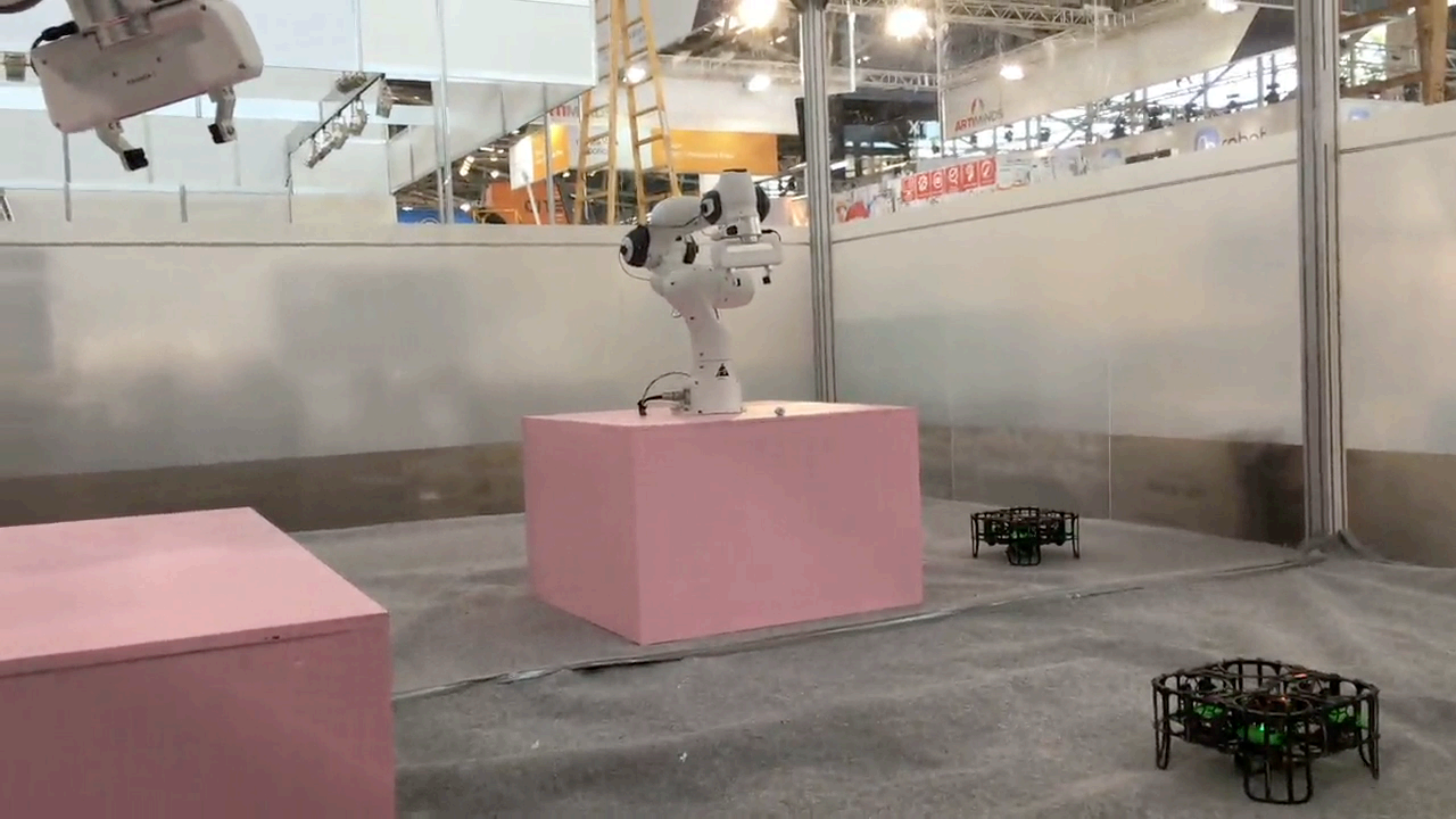


control through DVBF:
after unsupervised learning with Empowerment

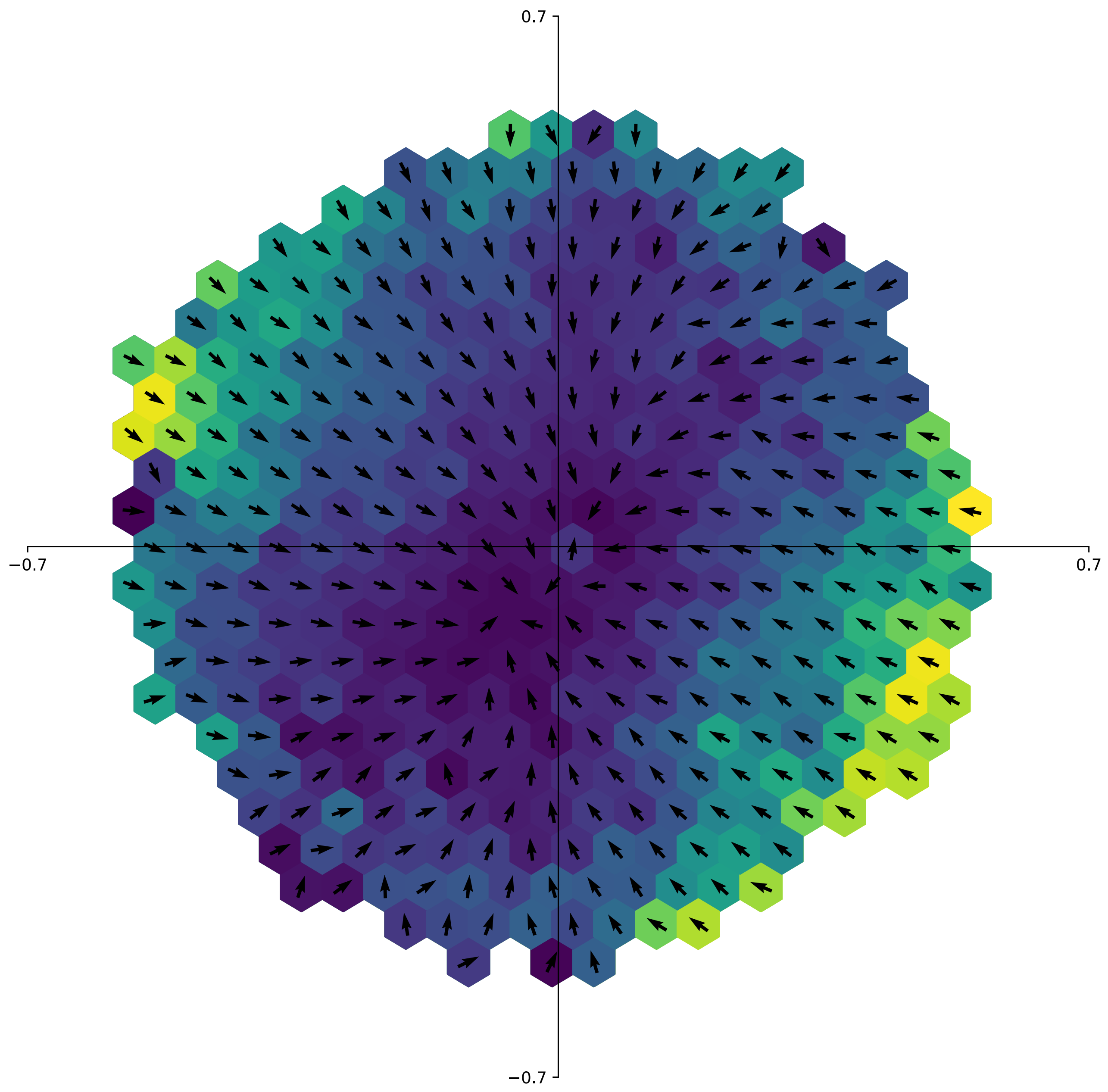




Karl & ... & van der Smagt & Bayer: *Unsupervised Real-Time Control through Variational Empowerment*, arXiv, 2017


ARTIMINDS

actions in lidar space



empowerment