**heidelberg.ai**

Oct. 16th, 7:30pm: How Neuroscience can help to solve AI (2/2)

**powered by**

# Deep Active Inference

+ Introduction to Variational Autoencoders and Evolution Strategies

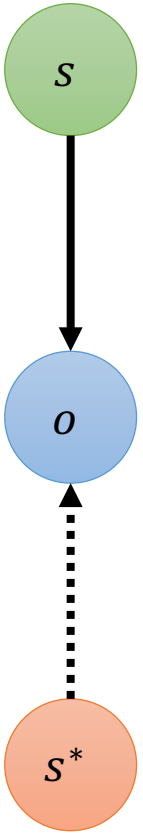
Kai Ueltzhöffer, 16.10.2017

# Outline

- Basics
  - Variational Autoencoder (VAE)
  - Evolution Strategies (ES)
  - Active Inference
- The Deep Active Inference Agent
  - A simple environment: The mountain car problem
  - Architecture of the Agent
  - Objective Function
  - Optimization
  - Results
- Discussion

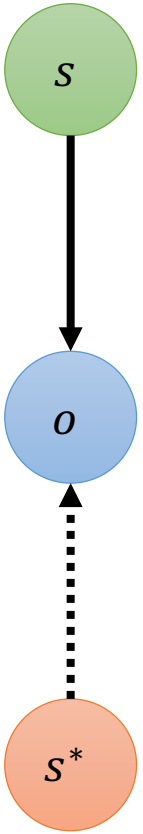
# Empirical Bayesian Inference

- Observations $o$ generated from true states of the world $s^*$ by true generative process.
- Generative model of $o$ and some hidden/latent variables $s$ (think of labels, object identities, position, …), with parameters $\theta$
- Often factorized:
  $p_\theta(o, s) = p_\theta(o|s)p_\theta(s)$



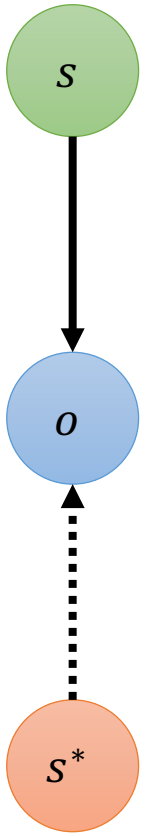
# Empirical Bayesian Inference

- Objective:
  - Maximum likelihood fit of parameters $\theta$
  - Bayesian inference on the posterior $p_\theta(s|o)$ on latent variables $s$, given observations $o$.
- Problem:
  - Exact solution $p_\theta(s|o) = \frac{p_\theta(o|s)p_\theta(s)}{p_\theta(o)}$ often computationally intractable.



# Approximative Inference Methods

- Point estimates (MAP, MLE):
  **Pro**: simple, fast
  **Con**: overfitting, no confidence intervalls
- Markov Chain Monte Carlo (MCMC):
  **Pro**: asymptotically unbiased
  **Con**: often expensive, hard to assess convergence
- Variational Inference:
  **Pro**: fast parametric approximation of posterior
  **Con**: possibly biased, representational power limited by functional family of approximation



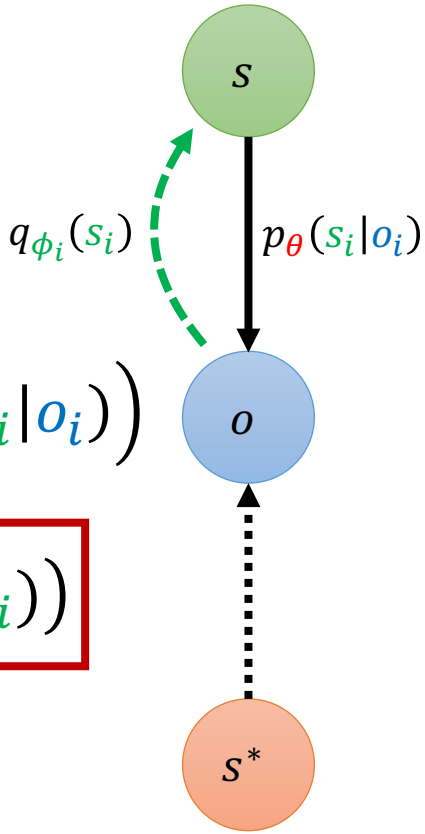
# Variational Inference

- Introduce parametric approximation $q_{\phi_i}(s_i)$ of true posterior $p_\theta(s_i|o_i)$ (e.g. diagonal Gaussian where $\phi_i = \{\mu_i, \sigma_i\}$).
- Minimize Variational Free Energy:

$$F(o, \theta, \phi) = \sum_i -\ln p_\theta(o_i) + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i)||p_\theta(s_i|o_i)\right)$$

$$= \sum_i \langle -\ln p_\theta(o_i|s_i)\rangle_{q_{\phi_i}(s_i)} + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i)||p_\theta(s_i)\right)$$

Cheap evaluation.




$$\langle f(x)\rangle_{p(x)} = \int f(x)p(x)\,\mathrm{d}x$$

$$D_{\mathrm{KL}}(q(x)||p(x)) = \langle \ln q(x) - \ln p(x)\rangle_{q(x)}$$

"Kullback-Leibler Divergence", Properties:

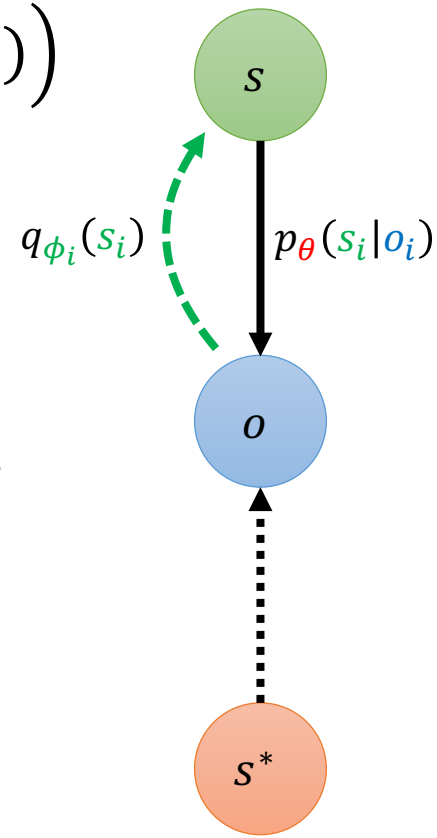
$$D_{\mathrm{KL}}(q(x)||p(x)) \geq 0$$

$$D_{\mathrm{KL}}(q(x)||p(x)) = 0 \Leftrightarrow p(x) = q(x)\ \forall\, x$$

# Variational Free Energy

$$F(o, \theta, \phi) = \sum_i -\ln p_\theta(o_i) + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i) || p_\theta(s_i|o_i)\right)$$

$$= \sum_i \langle -\ln p_\theta(o_i|s_i) \rangle_{q_{\phi_i}(s_i)} + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i) || p_\theta(s_i)\right)$$




- Upper bounds $-\ln p_\theta(o_i)$, i.e. minimization w.r.t. $\theta$ corresponds to maximum likelihood extimation of parameters.
- Minimizing w.r.t. $\phi_i$ tightens the bound by minimizing $D_{\mathrm{KL}}\left(q_{\phi_i}(s_i) || p_\theta(s_i|o_i)\right)$, i.e. driving variational densities $q_{\phi_i}(s_i)$ towards true posterior $p_\theta(s_i|o_i)$.

# Recognition Network

- Vanilla variational inference optimizes individual parameters $\phi_i$ for each observation $o_i$. This means additional, costly optimizations for each new observation $o_i$ at test time (i.e. after learning).
- Idea: Use a deep neural network to approximate the complex functional dependency of variational parameters $\phi_i$ on observations $o_i$, utilizing the flexibility of neural networks. Add the parameters of this mapping to the set of fixed parameters $\theta$.

$$q_{\phi_i}(s_i) \rightarrow q_\theta(s_i|o_i) = N(s_i; \mu_\theta(o_i), \sigma_\theta(o_i))$$

# Final VAE Objective

- Minimize Variational Free Energy

$$F(o,\theta) = \sum_i -\ln p_\theta(o_i) + D_{\mathrm{KL}}\left(q_\theta(s_i|o_i)||p_\theta(s_i|o_i)\right)$$

$$= \sum_i \langle -\ln p_\theta(o_i|s_i)\rangle_{q_\theta(s_i|o_i)} + D_{\mathrm{KL}}\left(q_\theta(s_i|o_i)||p_\theta(s_i)\right)$$

w.r.t. parameters $\theta$ to obtain generative model $p_\theta(o_i|s_i)p_\theta(s_i)$ and very fast and efficient approximation $q_\theta(s_i|o_i)$ of true posterior $p_\theta(s_i|o_i)$.

# Optimization of VAE Objective

- Gradient descent w.r.t. $\theta$:

$$\nabla_\theta F(o,\theta) = \nabla_\theta \sum_i \langle -\ln p_\theta(o_i|s_i)\rangle_{q_\theta(s_i|o_i)} + D_{\mathrm{KL}}\left(q_\theta(s_i|o_i)||p_\theta(s_i)\right)$$

$$= \sum_i \nabla_\theta \langle -\ln p_\theta(o_i|s_i)\rangle_{q_\theta(s_i|o_i)} + \boxed{\nabla_\theta D_{\mathrm{KL}}\left(q_\theta(s_i|o_i)||p_\theta(s_i)\right)}$$

Closed form for diagonal Gaussians

$$= \sum_i \nabla_\theta \langle -\ln p_\theta(o_i|s_i) + \ln q_\theta(s_i|o_i) - \ln p_\theta(s_i)\rangle_{q_\theta(s_i|o_i)}$$

Otherwise

Expectation value over $q_\theta(s_i|o_i)$ not tractable
→ Sampling based approximation

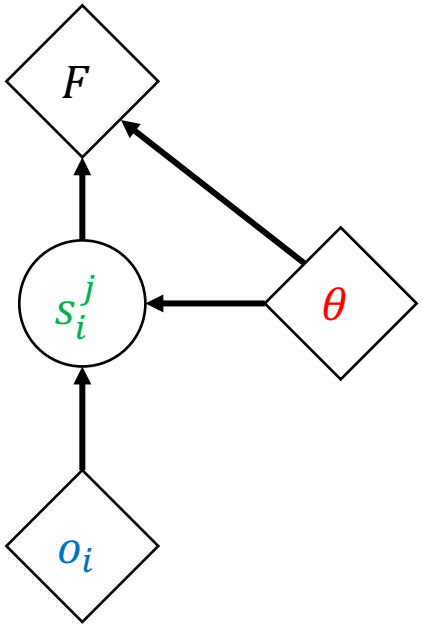
→ Problem: Backpropagation through samples

$$s_i^j \sim q_\theta(s_i|o_i)$$

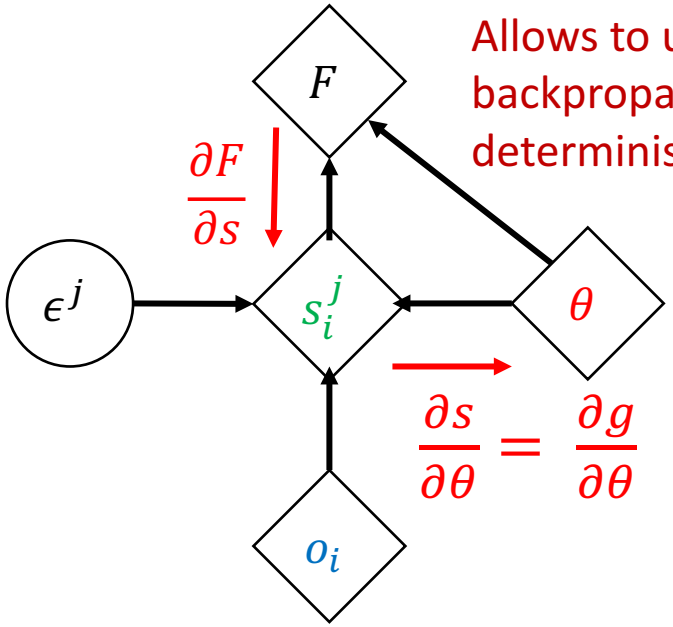
# Reparameterization Trick

- Sample $\epsilon^j, j = 1, \ldots, n_{\text{samples}}$ from $p(\epsilon) = N(\epsilon; 0,1)$
- $s_i^j = g_\theta(\epsilon^j, o_i)$, such that $s_i^j \sim q_\theta(s_i|o_i)$
- $F \approx \frac{1}{n_{\text{samples}}} \sum_{i,j} -\ln p_\theta\left(o_i|s_i^j\right) + \ln q_\theta\left(s_i^j|o_i\right) - \ln p_\theta(s_i^j)$

Original Form




Reparameterized Form




Allows to use backpropagation through deterministic function g

Introduced in Kingma and Welling, Auto-Encoding Variational Bayes, https://arxiv.org/abs/1312.6114; Rezende, Mohamed and Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models https://arxiv.org/abs/1401.4082

# Reparameterization Trick

- Works on a large range of distributions, e.g. any location-scale family (Normal, Laplace, Logistic, …)
- Needs a good gradient-based optimizer to work well (e.g. ADAM)
- Allows variational inference by gradient descent on deep generative models.

# Relation to Autoencoders

- Learned likelihood $p_\theta(o_i|s_i)$ → "Decoder"
- Approximate posterior $q_\theta(s_i|o_i)$ → "Encoder"

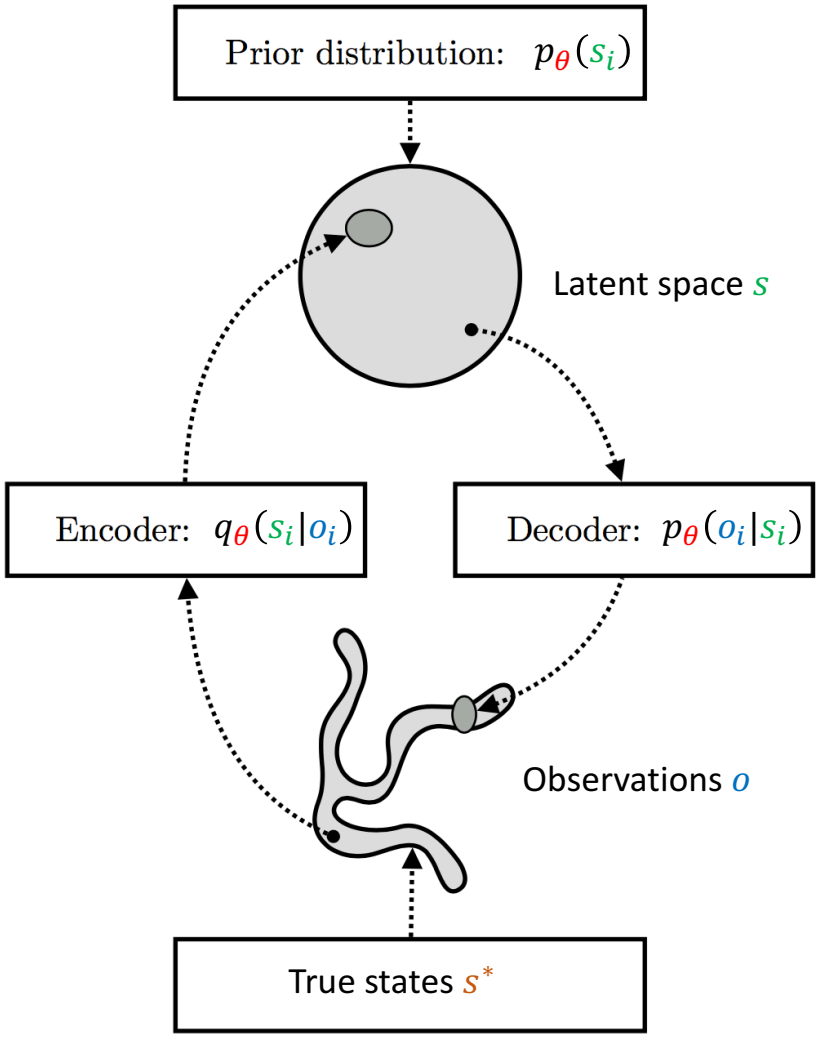



Illustration modified from: D.P. Kingma, "Variational Inference and Deep Learning: A New Synthesis", PhD Thesis, http://dpkingma.com

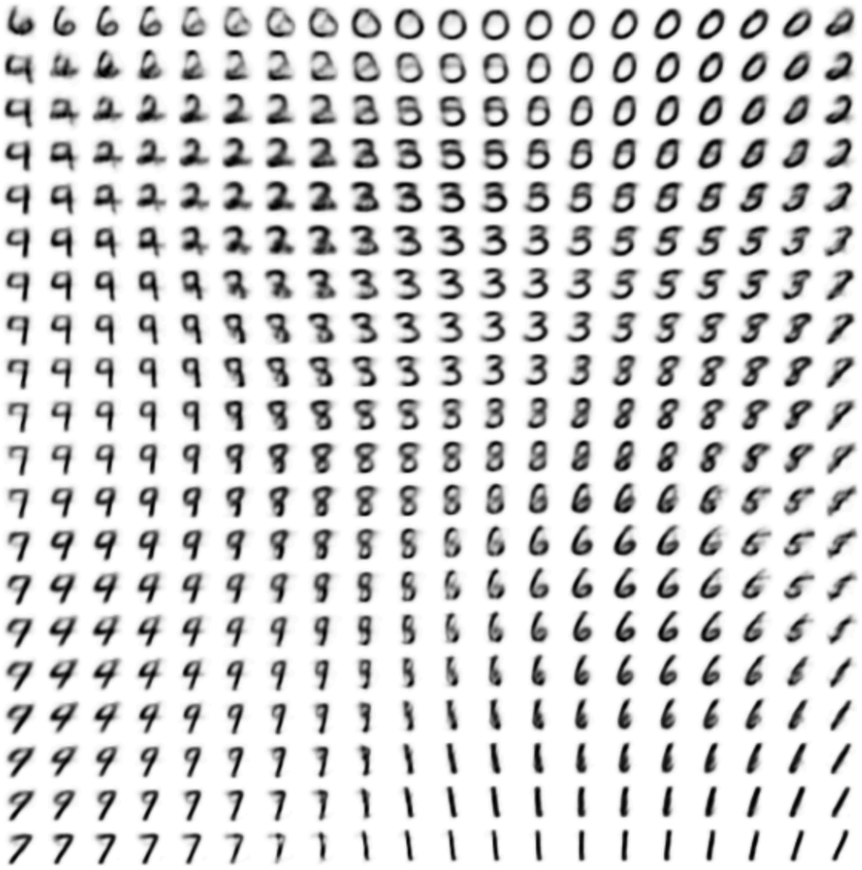
# Examples

- Frey Faces & MNIST

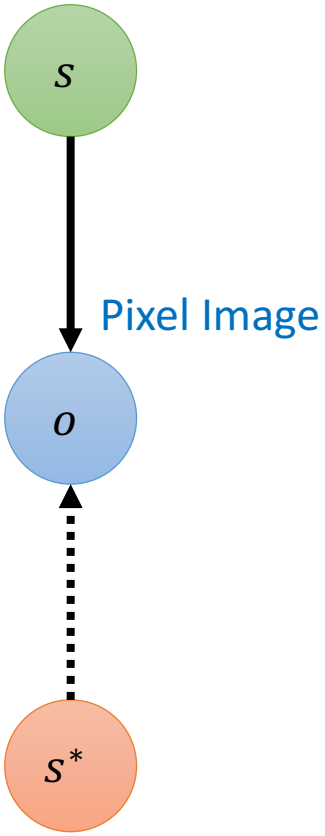
Face Identity & Mood

Number & Handwriting Style

(a) Learned Frey Face manifold



(b) Learned MNIST manifold




Decoding from 2D-Latent Space

Figures from: Kingma and Welling, Auto-Encoding Variational Bayes, https://arxiv.org/abs/1312.6114

# Examples

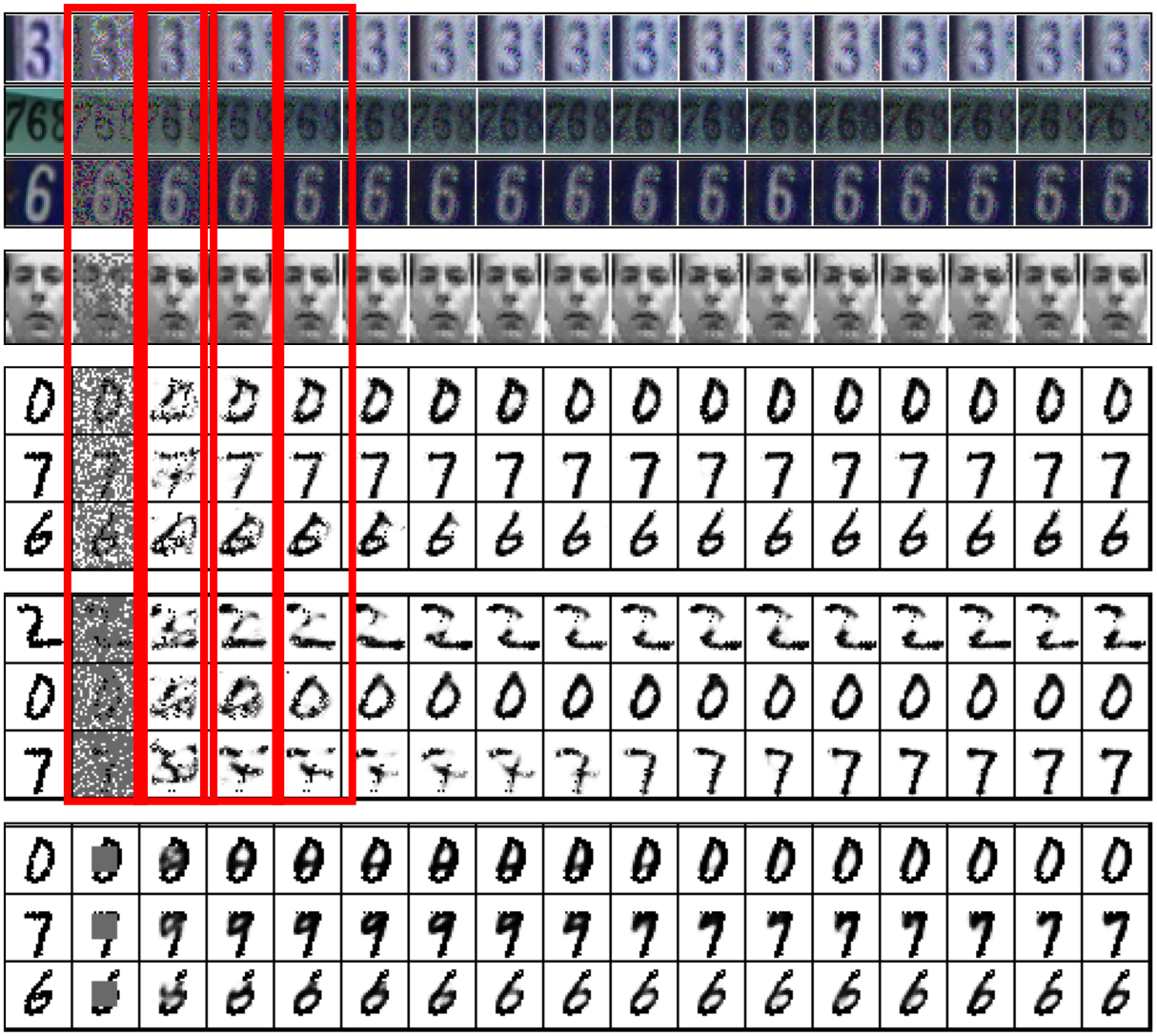
Figure from: Rezende, Mohamed and Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models https://arxiv.org/abs/1401.4082

- SVHN & Frey Faces & MNIST

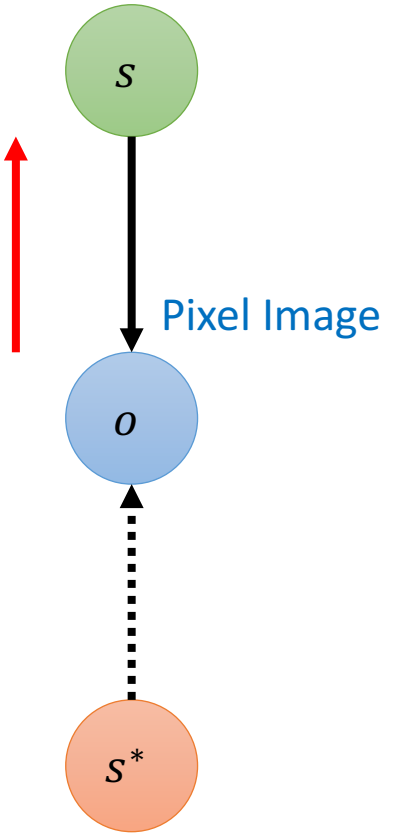
House Number

Face Identity & Mood

Number & Handwriting Style



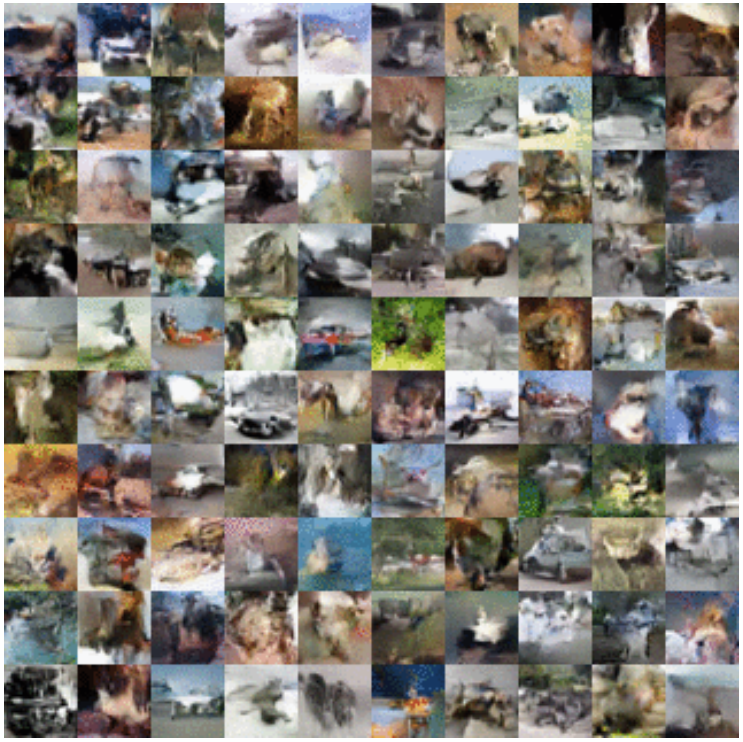
Imputation of Missing Data by **Constrained Sampling from Generative Model**



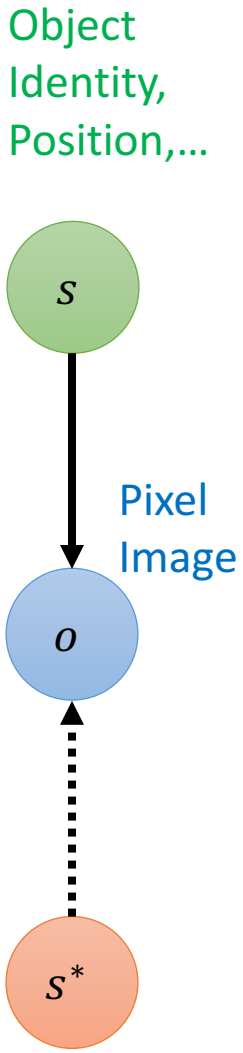

Idea: Project back and forth between data space and latent space, while keeping the known pixels fixed. If initialized close enough to true solution, the samples for the unknown pixels will converge to samples from the constrained probability distribution $p(o_{\text{unknown}}|o_{\text{known}})$

# Examples

CIFAR-10 using DRAW



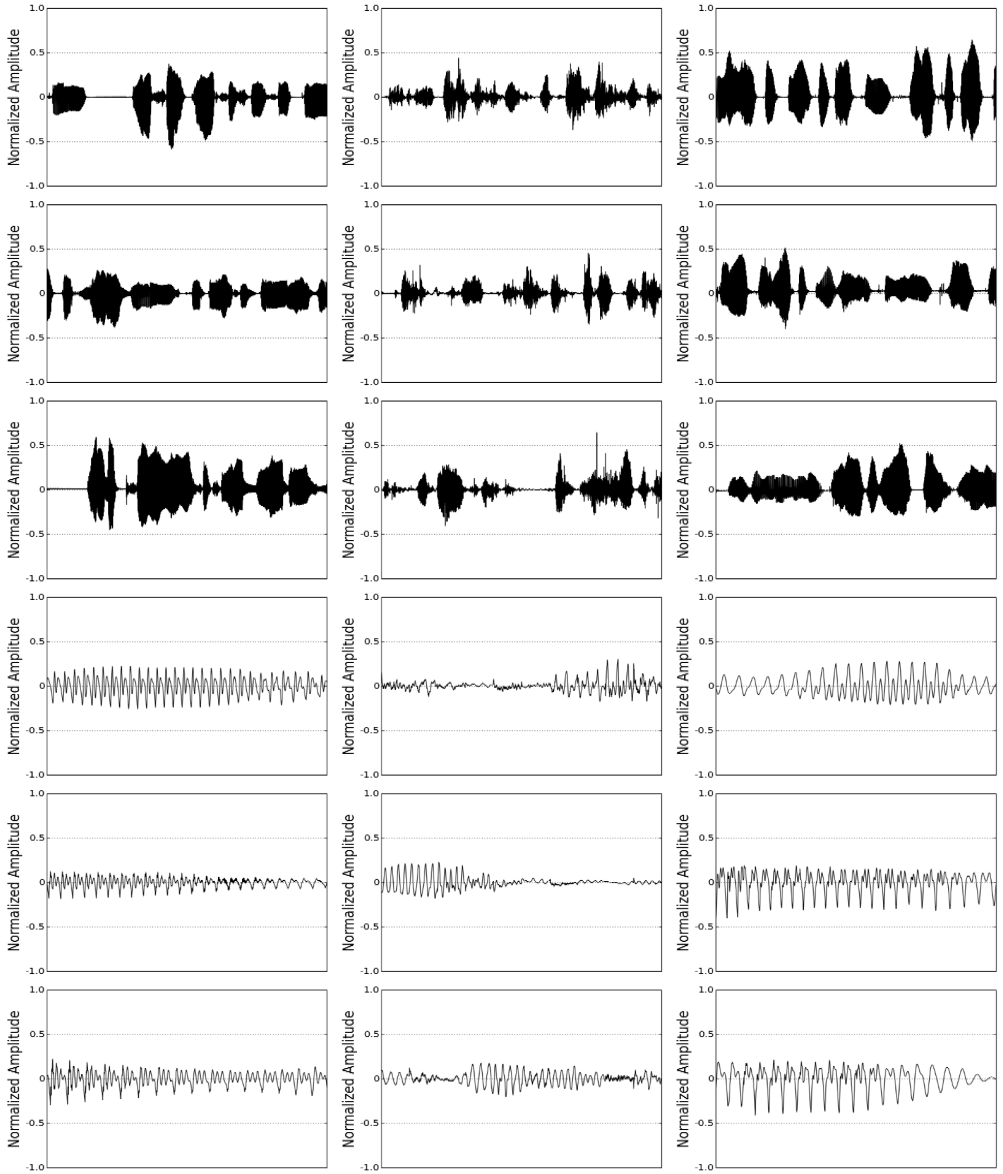
CIFAR-10 using VAE with diagonal Gaussian densities and **normalizing flows (namely Inverse Autoregressive Flows)**



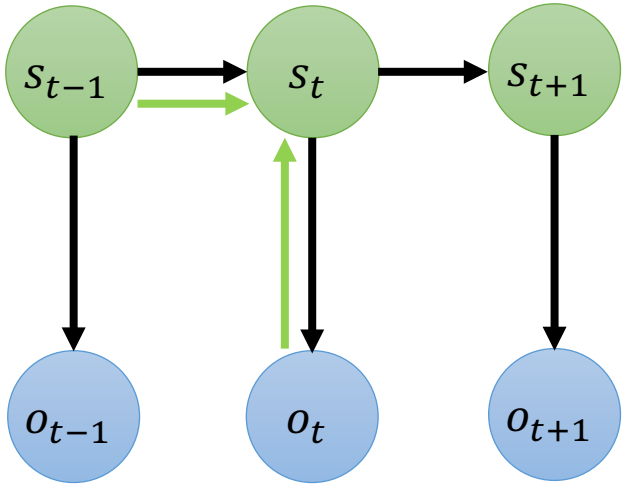

Images from: https://blog.openai.com/generative-models/
Paper: Kingma et al., Improving Variational Inference with Inverse Autoregressive Flow, https://arxiv.org/abs/1606.04934,
Code: https://github.com/openai/iaf
c.f. http://www.inference.vc/choice-of-recognition-models-in-vaes-a-regularisation-view/

# Models of Time-Series

From: Chung et al., Recurrent Latent Variable Model for Sequential Data, https://arxiv.org/abs/1506.02216




Hidden state-space model



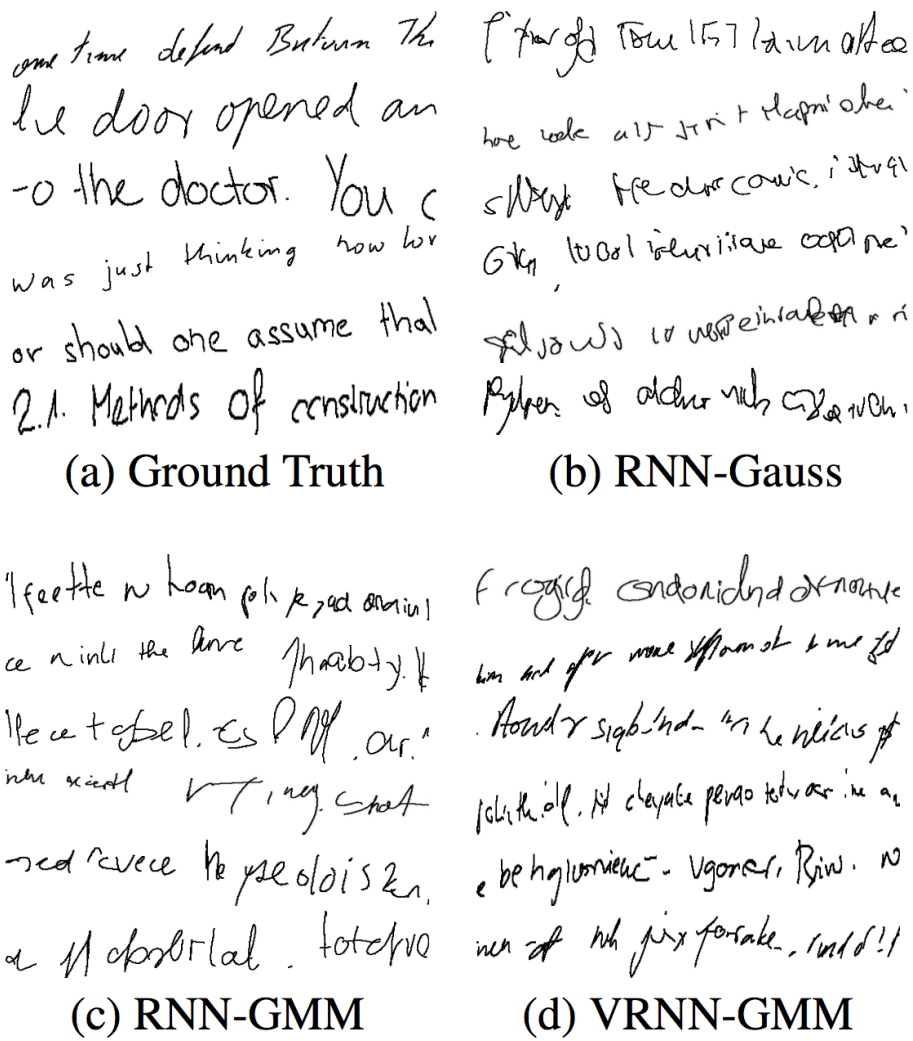

Raw audio waveform of female speach

- Training data: 300h of audiobooks by single, female speaker

- Samples

# Models of Time-Series

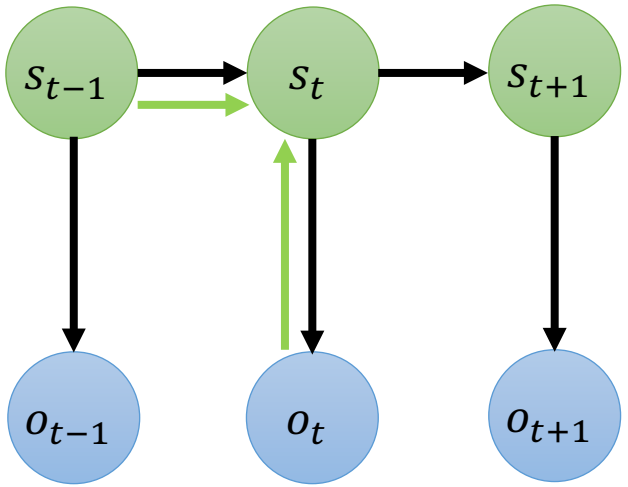
From: Chung et al., Recurrent Latent Variable Model for Sequential Data, https://arxiv.org/abs/1506.02216



(a) Ground Truth (b) RNN-Gauss

(c) RNN-GMM (d) VRNN-GMM

Hidden state-space model




X-Y Position and Up/Down State of Pen on Paper

# Summary: Variational Autoencoder

- **Allows approximate Bayesian inference on deep generative models using gradient descent.**
- In practice, diagonal Gaussian forms are assumed for the likelihood $p_\theta(o_i|s_i)$ , prior $p_\theta(s_i)$ and variational density $q_\theta(s_i|o_i)$ .
- Using normalizing flows to nonlinearly transform the variational density, these restrictions can be ameliorated for $q_\theta(s_i|o_i)$ .
- Learned generative models allow (constrained) sampling to generate full observations or impute missing data.
- The resulting approximate posterior allows to get sensible confidence bounds on inferred latent variables.

# Evolution Strategies

- Decades old (Rechenberg & Eigen, 1973).
- Black-box optimization technique for non-differentiable objective functions.
- Using this optimization technique, simple direct policy search was recently shown to yield comparable performance to state-of-the art reinforcement learning algorithms in MuJoCo Control Tasks and Atari-Games

# Setup

Objective:

- Minimize objective function $F(\theta)$ w.r.t. parameters $\theta$

Problem:

- Partial derivatives $\frac{\partial F(\theta)}{\partial \theta}$ are not accessible

# Idea

Instead of looking for a single, optimal parameter vector $\theta$, introduce **population density** on parameters $p_\psi(\theta) = N(\theta; \mu^\psi, \sigma^\psi)$ with sufficient statistics $\psi = \{\mu^\psi, \sigma^\psi\}$ and optimize the expectation value

$$\eta(\psi) = \langle F(\theta) \rangle_{p_\psi(\theta)}$$

w.r.t. $\psi$.

It is easy to show that the gradients w.r.t. the sufficient statistics $\psi$ of the population density are

$$\nabla_\psi \eta(\psi) = \left\langle F(\theta) \nabla \ln p_\psi(\theta) \right\rangle_{p_\psi(\theta)}$$

By (again) using a diagonal Gaussian for $p_\psi(\theta)$, i.e. $\psi = \{\mu^\psi; \sigma^\psi\}$, one can now estimate the gradients w.r.t. the means $\mu^\psi$ as

$$\nabla_{\mu^\psi} \eta(\psi) = \left\langle F(\theta) \frac{1}{(\sigma^\psi)^2} (\theta - \mu^\psi) \right\rangle_{p_\psi(\theta)}$$

"Smoothed finite-differences approximation"

And w.r.t the standard-deviations $\sigma^\psi$ as

$$\nabla_{\sigma^\psi} \eta(\psi) = \left\langle F(\theta) \frac{(\theta - \mu^\psi)^2 - (\sigma^\psi)^2}{(\sigma^\psi)^3} \right\rangle_{p_\psi(\theta)}$$

# Sampling based approximation

These gradients can easily be approximated using **samples from the population density** to approximate the expectation value.

Using $n_{\text{samples}}$ samples $\theta_i \sim p_\psi(\theta)$:

$$\nabla_{\mu^\psi} \eta(\psi) \approx \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} F(\theta_i) \frac{1}{(\sigma^\psi)^2} (\theta_i - \mu^\psi)$$

$$\nabla_{\sigma^\psi} \eta(\psi) \approx \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} F(\theta_i) \frac{(\theta_i - \mu^\psi)^2 - (\sigma^\psi)^2}{(\sigma^\psi)^3}$$

The sampling can trivially and very efficiently be parallelized, as only the parameter samples $\theta_i$ and the resulting values $F(\theta_i)$ have to be communicated between the main process and the compute processes.

# Intuition

$\eta\left(\mu_1^{\psi}, \mu_2^{\psi}\right)$, for **fixed values of** $\sigma_1^{\psi}, \sigma_2^{\psi}$.

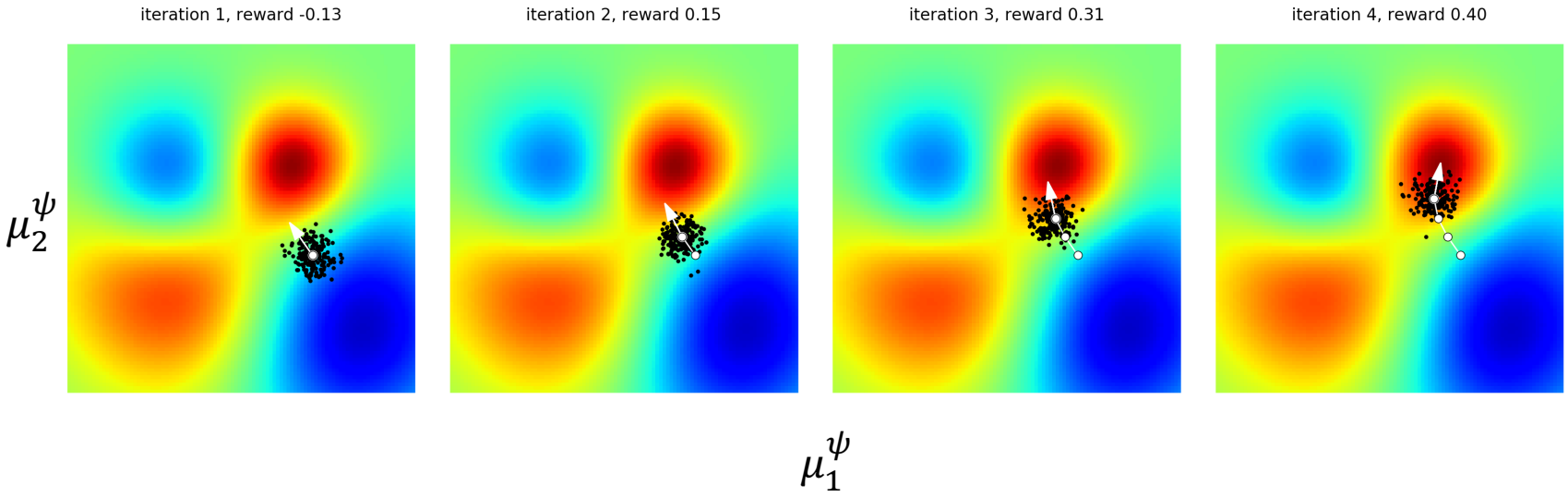
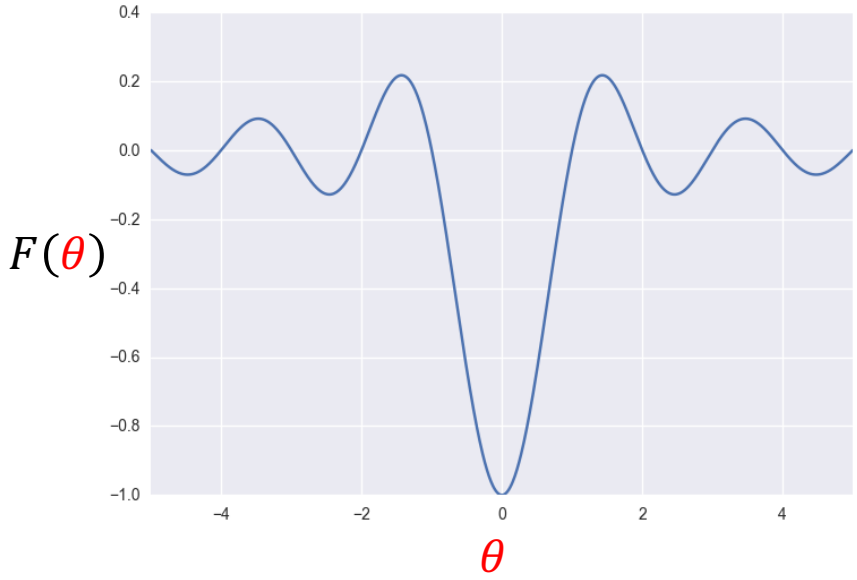
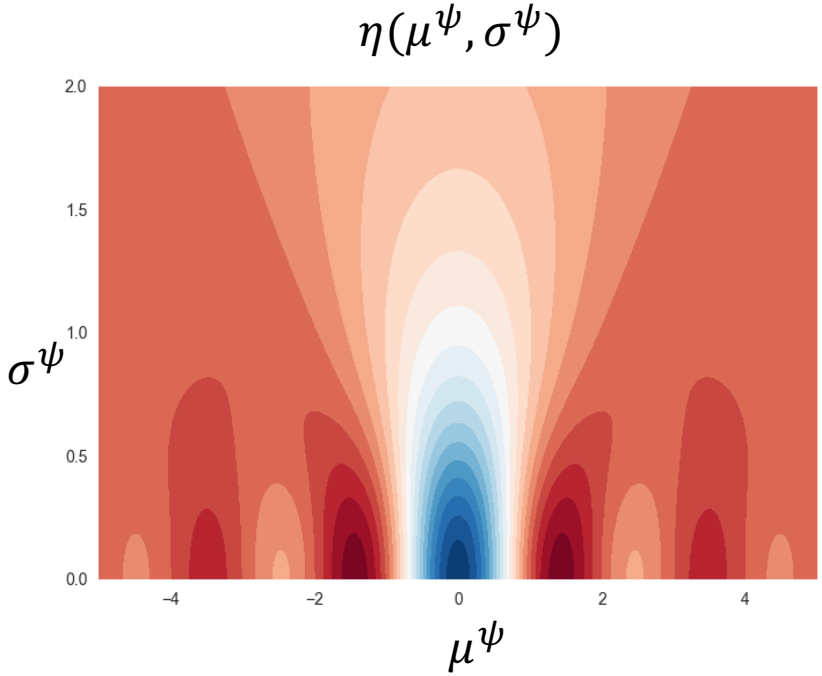


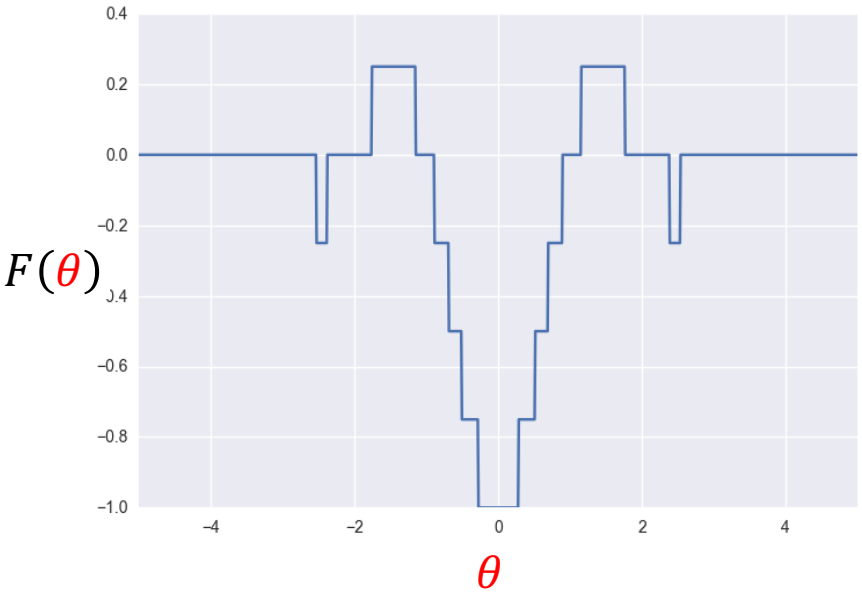

Illustration from: https://blog.openai.com/evolution-strategies/

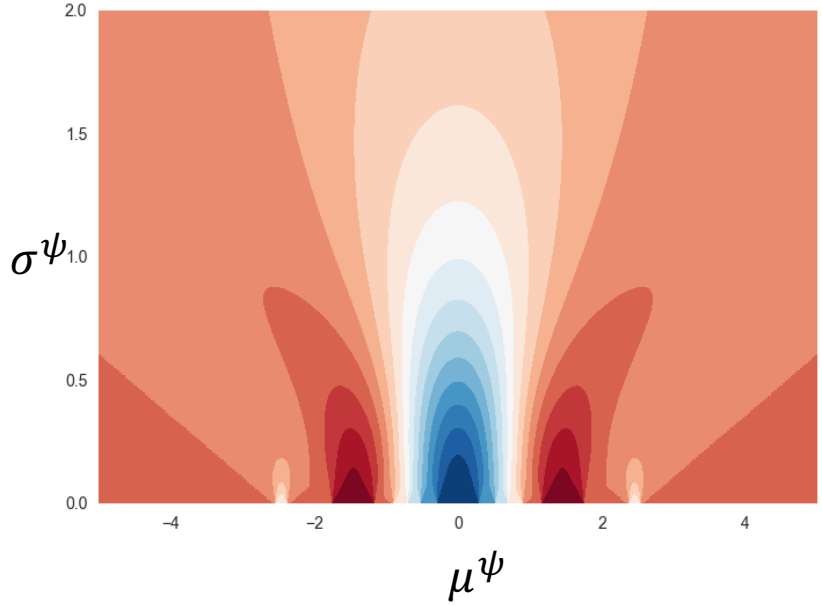
Images from: http://www.inference.vc/evolution-strategies-variational-optimisation-and-natural-es-2/

# Intuition

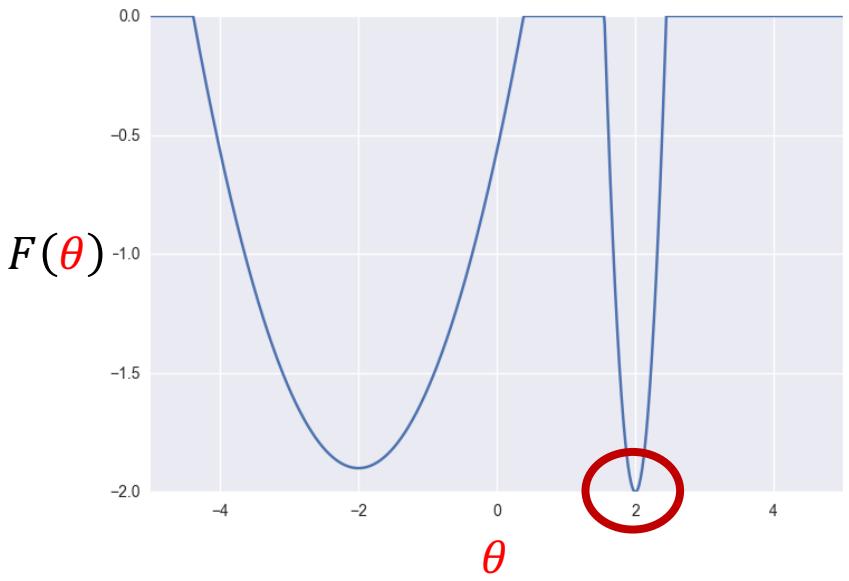


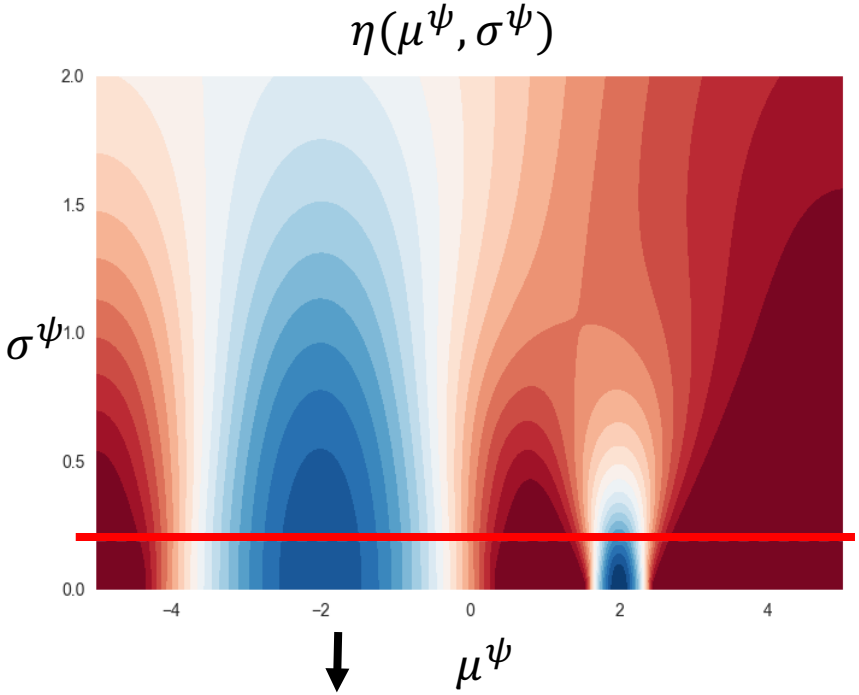



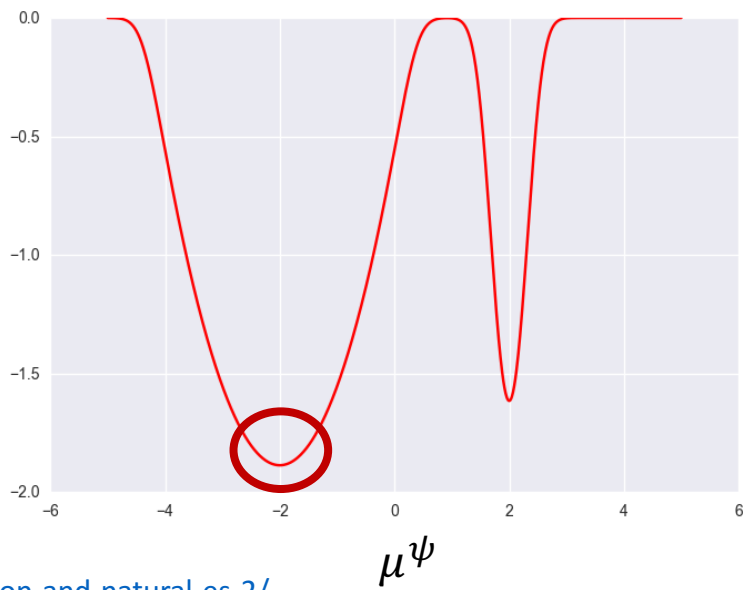



# Caveats



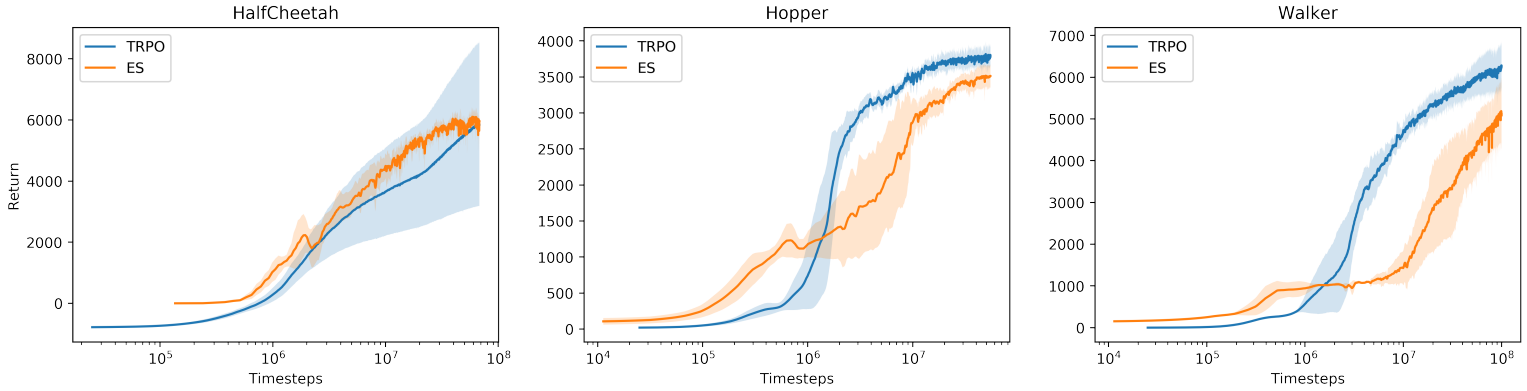

Always optimize means and standard deviations! Fixed values for the standard deviation might lead to spurious minima.

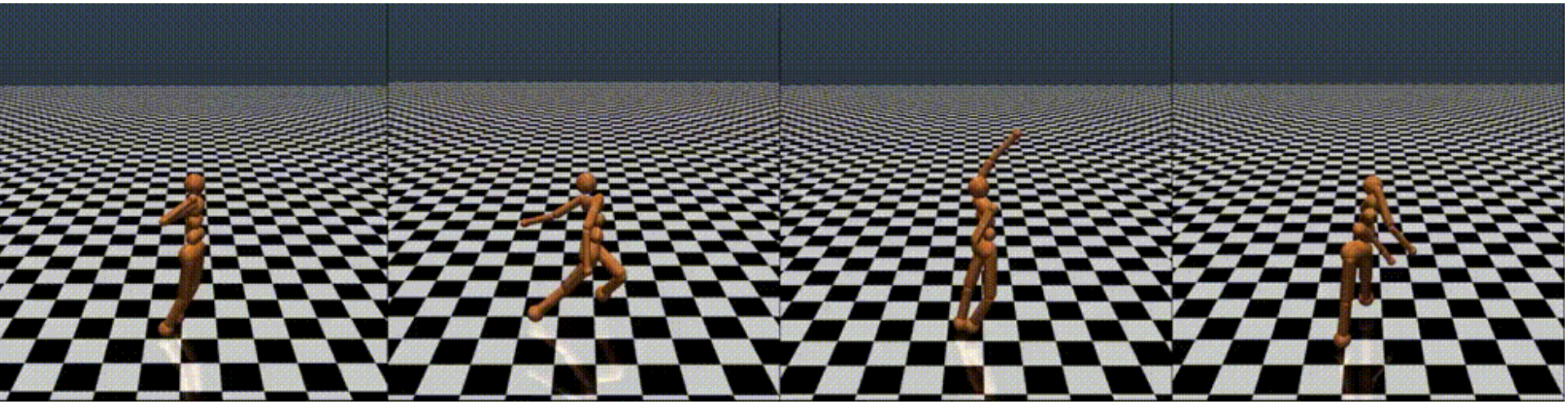
Images from: http://www.inference.vc/evolution-strategies-variational-optimisation-and-natural-es-2/

# Examples

$\theta \rightarrow$ Parameters of control policy

$F(\theta) \rightarrow$ Distance Walked



Images from: https://blog.openai.com/evolution-strategies/

Paper: Salimans et al., Evolution Strategies as a Scalable Alternative to Reinforcement Learning, https://arxiv.org/abs/1703.03864,

Code: https://github.com/openai/evolution-strategies-starter

# Examples

$\theta$ → Parameters of control policy

$F(\theta)$ → Score

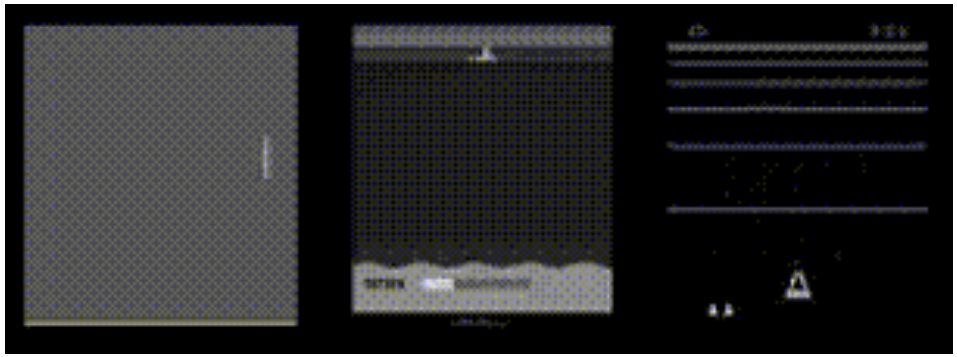


Illustration from: https://blog.openai.com/evolution-strategies/
Table from: Salimans et al., Evolution Strategies as a Scalable Alternative to Reinforcement Learning, https://arxiv.org/abs/1703.03864,
Code: https://github.com/openai/evolution-strategies-starter

| Game | DQN | A3C FF, 1 day | HyperNEAT | ES FF, 1 hour | A2C FF |
|---|---|---|---|---|---|
| Amidar | 133.4 | 283.9 | 184.4 | 112.0 | **548.2** |
| Assault | 3332.3 | **3746.1** | 912.6 | 1673.9 | 2026.6 |
| Asterix | 124.5 | **6723.0** | 2340.0 | 1440.0 | 3779.7 |
| Asteroids | 697.1 | **3009.4** | 1694.0 | 1562.0 | 1733.4 |
| Atlantis | 76108.0 | 772392.0 | 61260.0 | 1267410.0 | **2872644.8** |
| Bank Heist | 176.3 | **946.0** | 214.0 | 225.0 | 724.1 |
| Battle Zone | 17560.0 | 11340.0 | **36200.0** | 16600.0 | 8406.2 |
| Beam Rider | 8672.4 | **13235.9** | 1412.8 | 744.0 | 4438.9 |
| Berzerk | | **1433.4** | 1394.0 | 686.0 | 720.6 |
| Bowling | 41.2 | 36.2 | **135.8** | 30.0 | 28.9 |
| Boxing | 25.8 | 33.7 | 16.4 | 49.8 | **95.8** |
| Breakout | 303.9 | **551.6** | 2.8 | 9.5 | 368.5 |
| Centipede | 3773.1 | 3306.5 | **25275.2** | 7783.9 | 2773.3 |
| Chopper Command | 3046.0 | **4669.0** | 3960.0 | 3710.0 | 1700.0 |
| Crazy Climber | 50992.0 | **101624.0** | 0.0 | 26430.0 | 100034.4 |
| Demon Attack | 12835.2 | **84997.5** | 14620.0 | 1166.5 | 23657.7 |
| Double Dunk | **21.6** | 0.1 | 2.0 | 0.2 | 3.2 |
| Enduro | **475.6** | 82.2 | 93.6 | 95.0 | 0.0 |
| Fishing Derby | 2.3 | 13.6 | **49.8** | 49.0 | 33.9 |
| Freeway | 25.8 | 0.1 | 29.0 | **31.0** | 0.0 |
| Frostbite | 157.4 | 180.1 | **2260.0** | 370.0 | 266.6 |
| Gopher | 2731.8 | **8442.8** | 364.0 | 582.0 | 6266.2 |
| Gravitar | 216.5 | 269.5 | 370.0 | **805.0** | 256.2 |
| Ice Hockey | 3.8 | 4.7 | **10.6** | 4.1 | 4.9 |
| Kangaroo | 2696.0 | 106.0 | 800.0 | **11200.0** | 1357.6 |
| Krull | 3864.0 | 8066.6 | **12601.4** | 8647.2 | 6411.5 |
| Montezuma's Revenge | 50.0 | **53.0** | 0.0 | 0.0 | 0.0 |
| Name This Game | 5439.9 | 5614.0 | **6742.0** | 4503.0 | 5532.8 |
| Phoenix | | **28181.8** | 1762.0 | 4041.0 | 14104.7 |
| Pit Fall | | **123.0** | 0.0 | 0.0 | 8.2 |
| Pong | 16.2 | 11.4 | 17.4 | **21.0** | 20.8 |
| Private Eye | 298.2 | 194.4 | **10747.4** | 100.0 | 100.0 |
| Q*Bert | 4589.8 | 13752.3 | 695.0 | 147.5 | **15758.6** |
| River Raid | 4065.3 | **10001.2** | 2616.0 | 5009.0 | 9856.9 |
| Road Runner | 9264.0 | 31769.0 | 3220.0 | 16590.0 | **33846.9** |
| Robotank | **58.5** | 2.3 | 43.8 | 11.9 | 2.2 |
| Seaquest | **2793.9** | 2300.2 | 716.0 | 1390.0 | 1763.7 |
| Skiing | | 13700.0 | 7983.6 | **15442.5** | 15245.8 |
| Solaris | | 1884.8 | 160.0 | 2090.0 | **2265.0** |
| Space Invaders | 1449.7 | **2214.7** | 1251.0 | 678.5 | 951.9 |
| Star Gunner | 34081.0 | **64393.0** | 2720.0 | 1470.0 | 40065.6 |
| Tennis | 2.3 | 10.2 | 0.0 | 4.5 | **11.2** |
| Time Pilot | 5640.0 | 5825.0 | **7340.0** | 4970.0 | 4637.5 |
| Tutankham | 32.4 | 26.1 | 23.6 | 130.3 | **194.3** |
| Up and Down | 3311.3 | 54525.4 | 43734.0 | 67974.0 | **75785.9** |
| Venture | 54.0 | 19.0 | 0.0 | **760.0** | 0.0 |
| Video Pinball | 20228.1 | **185852.6** | 0.0 | 22834.8 | 46470.1 |
| Wizard of Wor | 246.0 | **5278.0** | 3360.0 | 3480.0 | 1587.5 |
| Yars Revenge | | 7270.8 | **24096.4** | 16401.7 | 8963.5 |
| Zaxxon | 831.0 | 2659.0 | 3000.0 | **6380.0** | 5.6 |

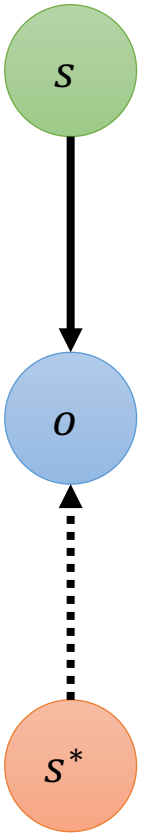
Table 2: Final results obtained using Evolution Strategies on Atari 2600 games (feedforward CNN policy, deterministic policy evaluation, averaged over 10 re-runs with up to 30 random initial no-ops), and compared to results for DQN and A3C from Mnih et al. [2016] and HyperNEAT from Hausknecht et al. [2014]. A2C is our synchronous variant of A3C, and its reported scores are obtained with 320M training frames with the same evaluation setup as for the ES results. All methods were trained on raw pixel input.

# Summary: Evolution Strategies

- Black-box optimizer for non-differentiable objective functions (e.g. the reward as function of the parameters of a policy)
- No need for backpropagation, i.e. explicit partial derivatives w.r.t. parameters.
- Parallelizes extremely well
- Allows exploration of new policies directly on parameter space. → No need to artificially force random exploration within individual policies. → Works also for fully deterministic policies.
- Can handle credit assignment over long timescales, as full games/episodes are evaluated to compute gradient estimates. → Works well in environments with long time horizons and/or sparse rewards.

# Active Inference

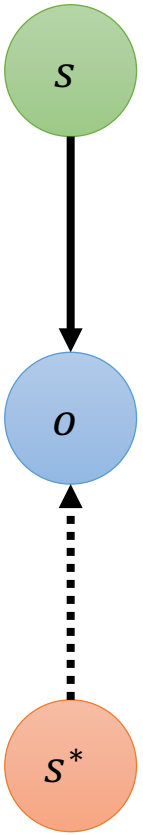
- Normative theory of brain function
- Motivated by a homeostatic argument:
  - To survive in a changing, noisy environment, a system has to keep certain vital parameters within strict bounds (think of body temperature, pH, $O_2$, $K^+$, $Na^+$, …).
  - To do this, it has to keep the entropy of its distribution on state space low, otherwise diffusive processes would at a certain point push one of these variables out of the viable range, leading to a phase transition (death).



# Active Inference

- The entropy of the true states $s^*$ can be bounded by the entropy of the sensory states $o$, given the sensory system has a sufficient mapping from vitally important states to sensory inputs (globus caroticus, hypothalamus, macula densa, …).

$$H(s^*) \leq H(o) + C$$



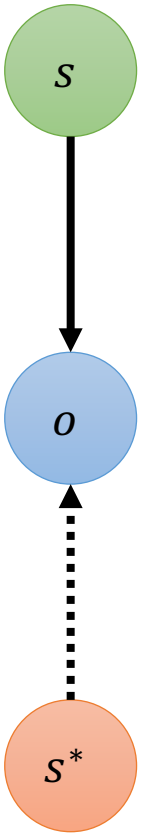
# Active Inference

- The entropy of the sensory states can be expressed, assuming ergodicity, as:

$$H(o) = \int -\ln p(o)\, p(o) \mathrm{d}o$$

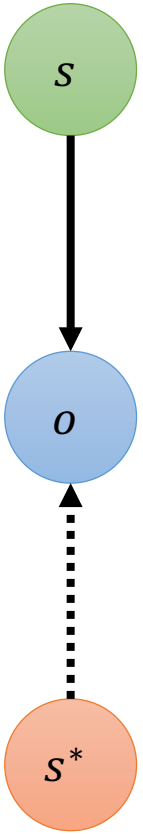
$$= \frac{1}{T} \int -\ln p(o) \mathrm{d}t$$

→ Can be minimized by an agent by minimizing the time integral over surprise $-\ln p(o)$ or by minimizing surprise at every instant (c.f. Euler-Lagrange formalism)



# Active Inference

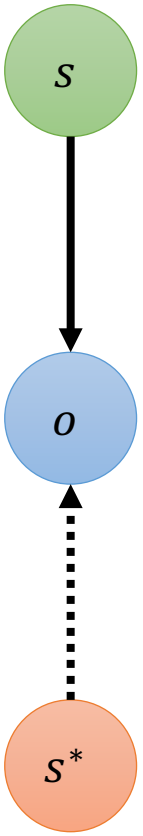
- Surprise directly is not accessible to the agent.
- However, if the agent possesses a generative model of its environment $p_\theta(o, s) = p_\theta(o|s)p_\theta(s)$ it could calculate surprise by marginalizing over hidden states → Computationally infeasible!



# Active Inference

- However, an agent could use a variational approximation to upper-bound surprise

$$F(o,\theta,\phi) = \sum_i -\ln p_\theta(o_i) + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i)||p_\theta(s_i|o_i)\right)$$
$$= \sum_i \langle -\ln p_\theta(o_i|s_i)\rangle_{q_{\phi_i}(s_i)} + D_{\mathrm{KL}}\left(q_{\phi_i}(s_i)||p_\theta(s_i)\right)$$

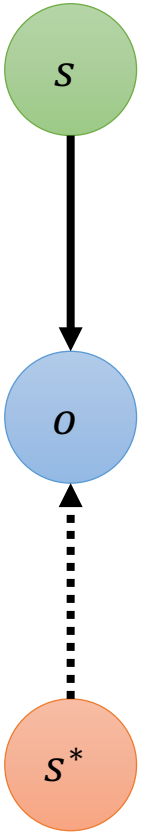



- The sufficient statistics $\phi_i$ , representing the approximate posterior over hidden variables, given observations, change on a fast timescale and could be represented using neural activity.
- The parameters $\theta$ of the generative model change slowly with the accumulation of new observations, and could be represented in terms of synaptic connectivity.

# Active Inference

- In contrast to a Variational Autoencoder, an agent can also act on the world, via active states $a$, describing the states of its muscles, motors, actuators, …
- The agent‘s observations $o$ are now a – complex – function of the agents active states $o(a)$
- Now the agent can change the states $a$ of its actuators to minimize a Free Energy bound on surprise (“Action”), which is made tight by optimizing the sufficient statistics $\phi_i$ of the variational density $q_{\phi_i}(s_i)$ (“Perception”) and the parameters $\theta$ of its generative model $p_\theta(o, s) = p_\theta(o|s)p_\theta(s)$ (“Learning”)
- Hypothesis: The dynamics of neural activity, synaptic plasticity and motor activity are given by:

$$(a, \theta, \phi) = \underset{(a^*, \theta^*, \phi^*)}{\operatorname{argmin}} F(o(a^*), \theta, \phi)$$



# Active Inference

- Con:
  - Very abstract.
  - Uses a loooong list of non-trivial assumptions.
  - No hard evidence so far.
  - Literature up to now has not shown that this functional can actually lead to intelligent behavior and learning in situations where the agent does now know a lot about its environment.
- Pro:
  - Integrated account of action, perception, behavior.
  - Derived from homeostatic (i.e. evolutionary) functional.
  - Explains – on an abstract level – basic features of neuroanatomy, electrophysiology, behavior.
  - Some preliminary evidence by behavioral and functional MRI experiments (far away from cell-physiology).
- To Dos:
  - Design experiments and test concrete models against data.
  - Demonstrate that Active Inference allows an agent to reach concrete goals in a not-entirely-trivial environment.

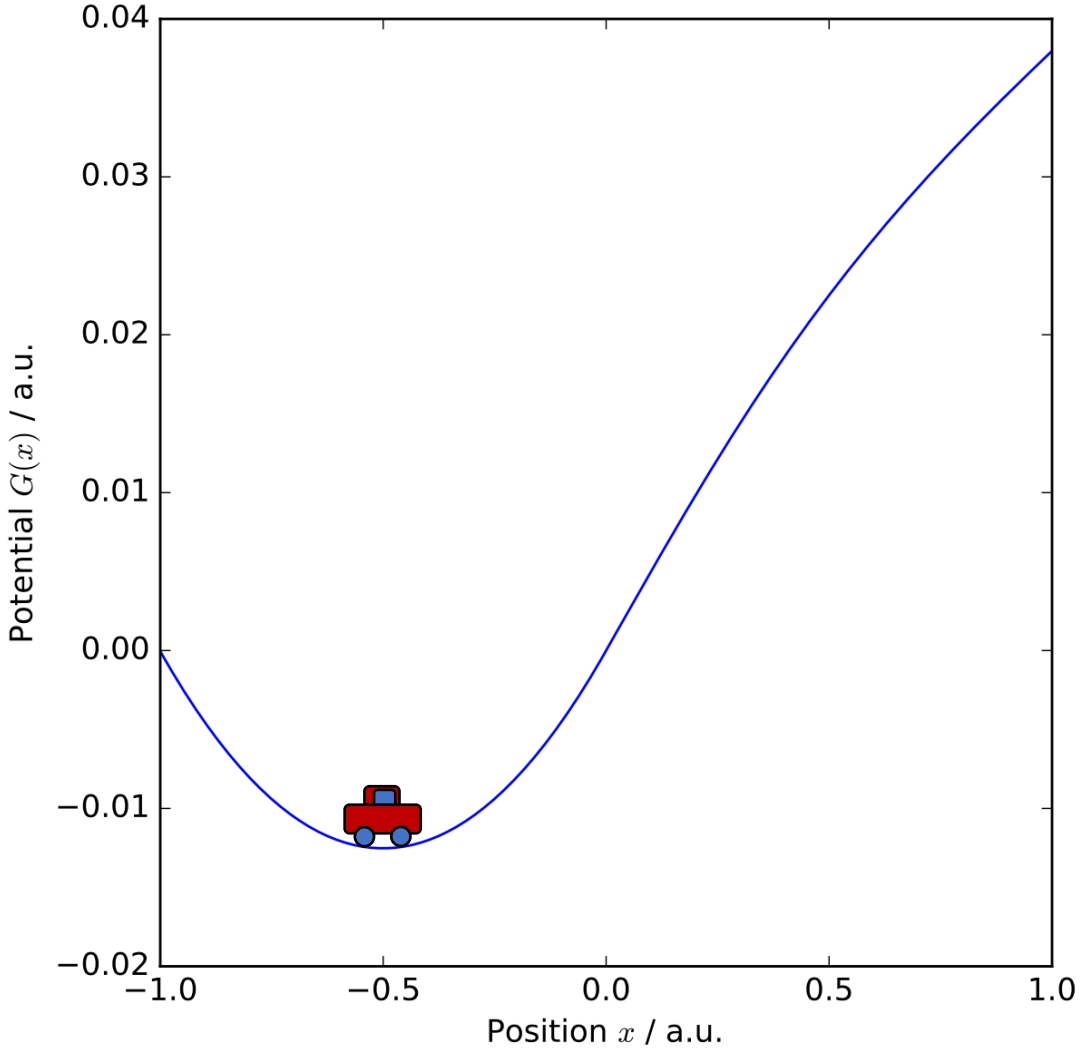
**First steps ahead now**.

# Deep Active Inference

- Implementation of Active Inference using methods form deep learning (mostly VAE and ES).
- Should scale well to larger environments.
- Up to now only example: Mountain car world.

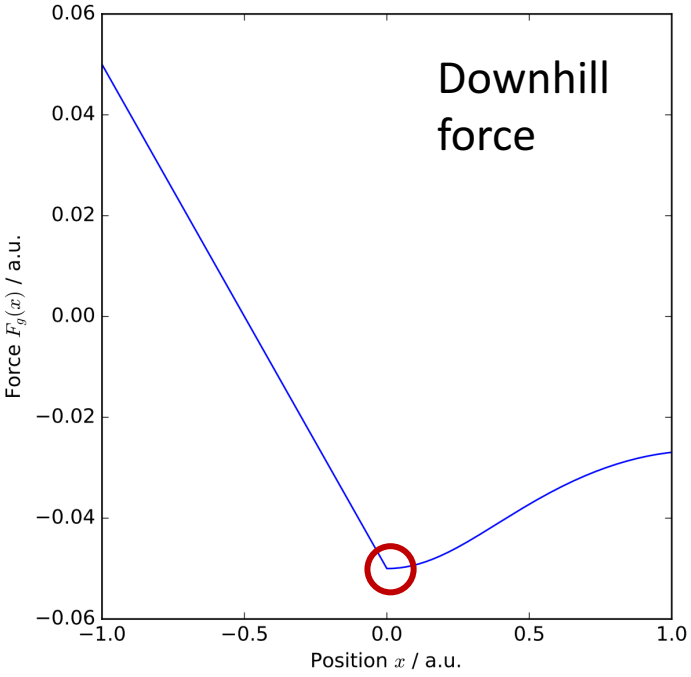
# Mountain Car Problem

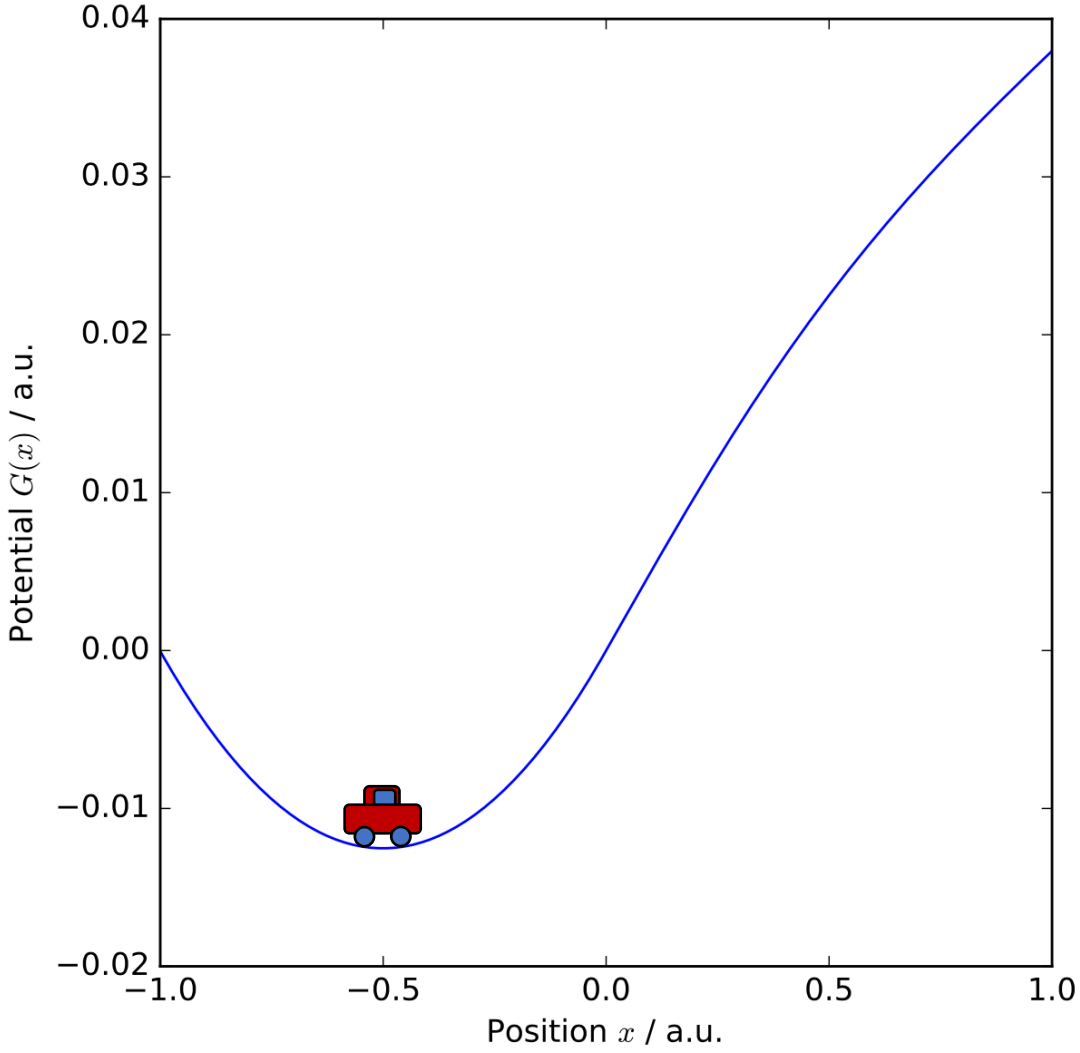
- Agent is a small car on a hilly landscape
- It starts at the bottom of the valley at $x = -0.5$
- Wants to reach position $x = 1.0$



# Mountain Car Problem

- Problem:
  Due to limited power output, $F_{\max}^{\text{motor}} = 0.03$, its motor can not overcome the steep part of the slope at $x = 0$ directly.



# Montain Car Problem

- True state $s^* = (a_t, x_t, v_t)$
- Initial state $(a_0, x_0, v_0) = (0, -0.5, 0)$
- Discreet time dynamics:

$$F_g(x) = -\frac{\partial G(x)}{\partial x} \quad \text{Gravity}$$
$$F_a(a) = 0.03 \tanh a \quad \text{Motor Output}$$
$$F_f(v) = -0.25v \quad \text{Friction}$$

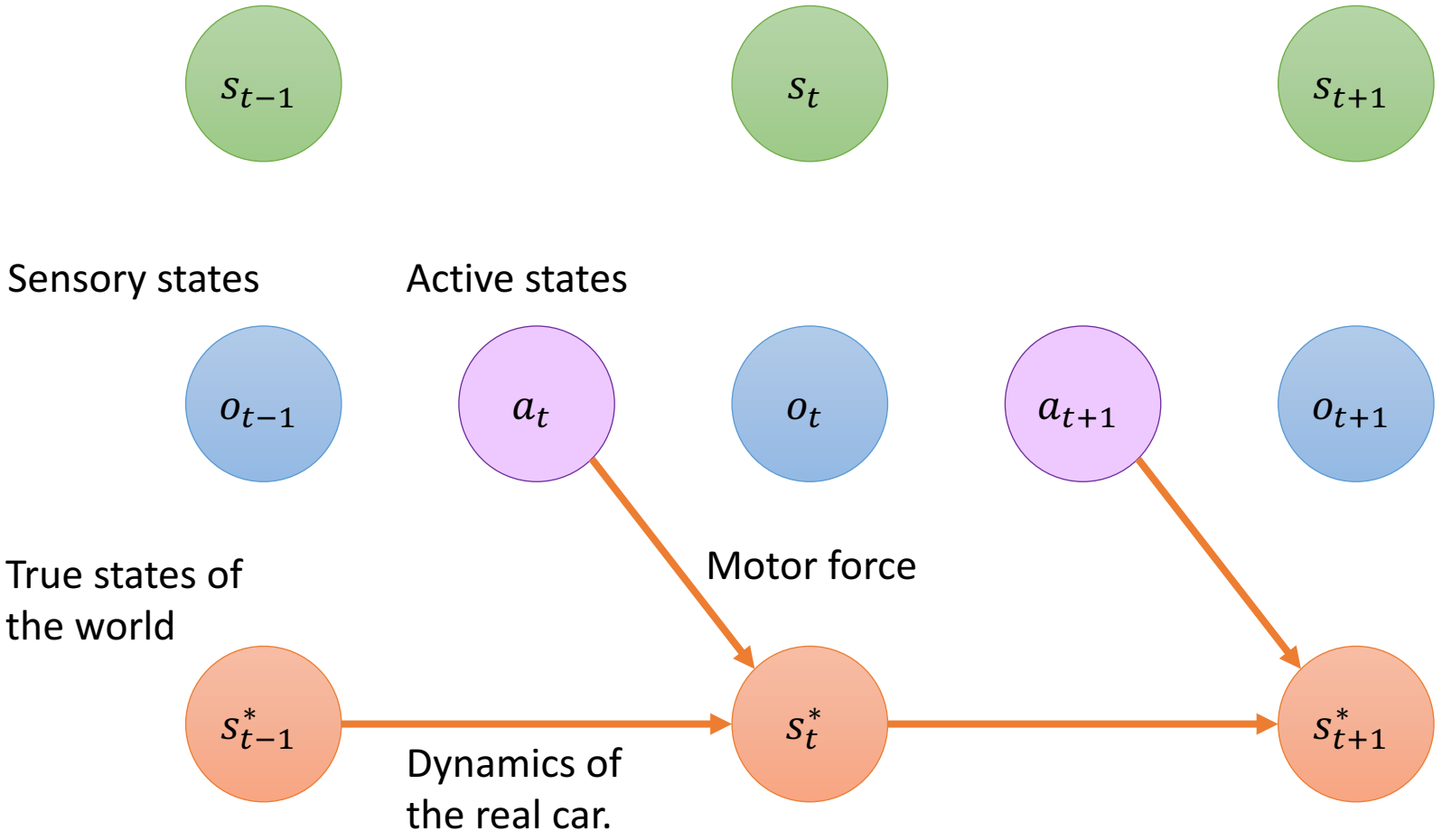
$$v_{t+1} = v_t + F_g(x_t) + F_a(a_{t+1}) + F_f(v_t)$$
$$x_{t+1} = x_t + v_{t+1}$$

- Shorthand:

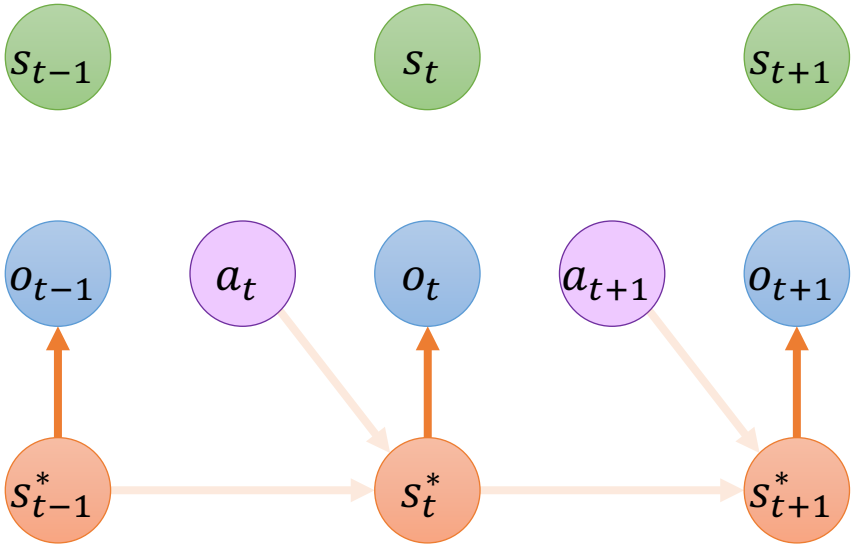
$$(x_{t+1}, v_{t+1}) = R(x_t, v_t, a_{t+1})$$

# Architecture of the Agent

Latent variables: Hidden states in the agent's generative model



# Architecture of the Agent




Sensory system:

Noisy sense of position

$$p(o_{x,t}|x_t) = N(o_{x,t}; x_t; 0.01)$$

Irrelevant sensory channel

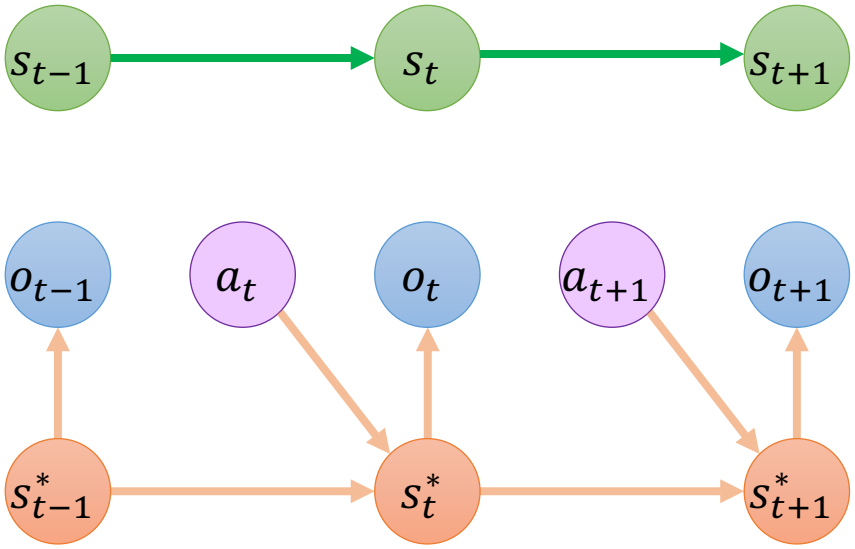
$$o_{h,t} = \exp\left(-\frac{(x_t - 1.0)^2}{2 \cdot 0.3^2}\right)$$

Proprioceptive sensory channel

$$o_{a,t} = a_t$$

Note: No direct sense of velocity or momentum!

# Agent's Generative Model




Priors:

$$p(s_{t+1}|s_t)$$

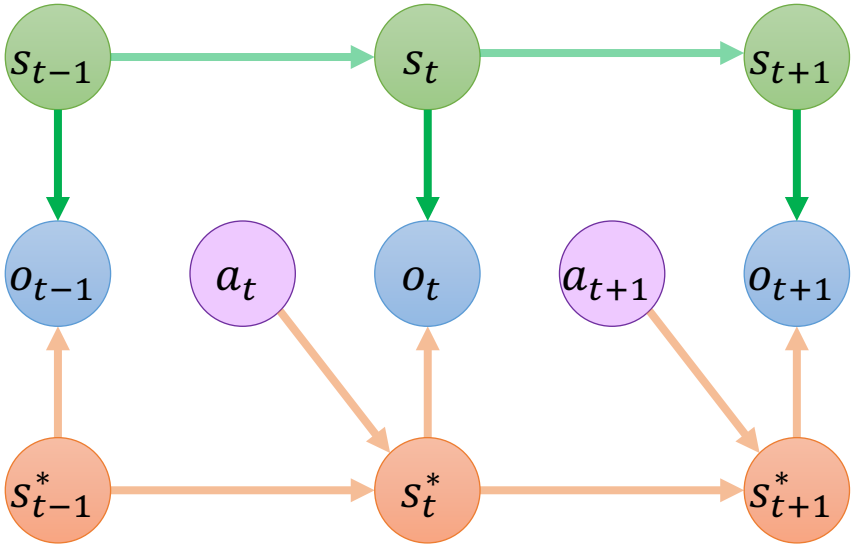
$$= N(s_{t+1}|\mu^t_\theta(s_t), \sigma^t_\theta(s_t))$$

where $s_0 = (0, \dots, 0)$

and $\dim s = 10$

Implemented using a fully connected network with tanh-nonlinearity for the means and softplus nonlinearity for the standard-deviations.

# Agent's Generative Model




Likelihoods:

Noisy sense of position

$$p(o_{x,t}|s_t) = N(o_{x,t}; \mu_\theta^{o_x}(s_t), \sigma_\theta^{o_x}(s_t))$$

Irrelevant sensory channel

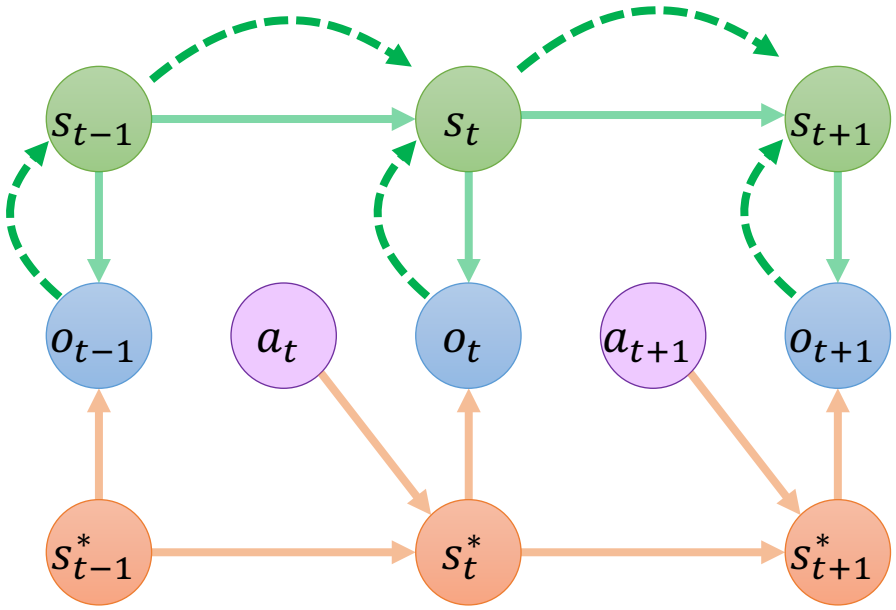
$$p(o_{h,t}|s_t) = N(o_{h,t}; \mu_\theta^{o_h}(s_t), \sigma_\theta^{o_h}(s_t))$$

Proprioceptive sensory channel

$$p(o_{a,t}|s_t) = N(o_{a,t}; \mu_\theta^{o_a}(s_t), \sigma_\theta^{o_a}(s_t))$$

Implemented by deep feed-forward network with 3 hidden tanh-layers and 10 hidden units per layer. Means are calculated from last hidden layer by a linear layer, standard-deviations by a softplus layer.

# Recognition density



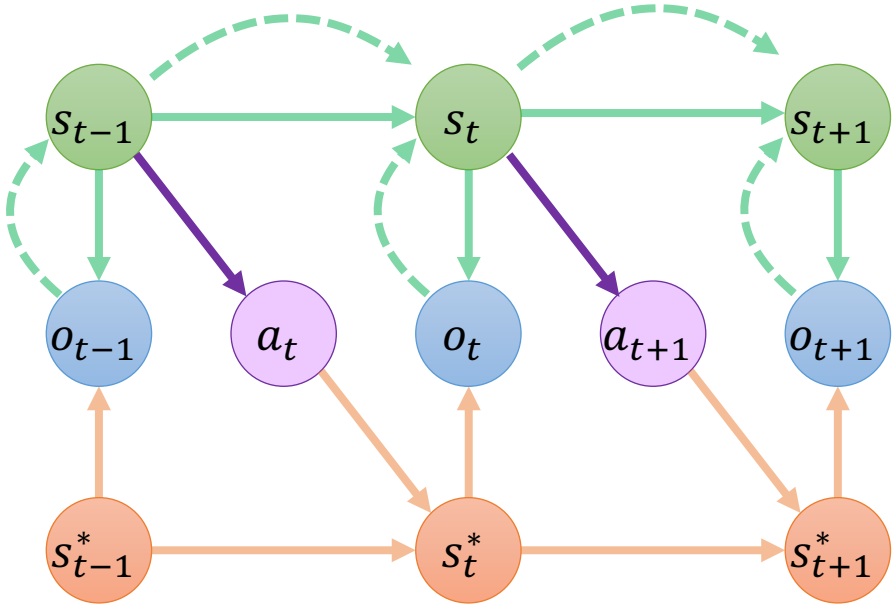

Analogous to Variational Autoencoder:

Instead of optimizing sufficient statistics $\phi_t$ of $q_{\phi_t}(s_t)$ at every time step, use fixed approximation by neural networks:

$$q_\theta\left(s_t \middle| s_{t-1}, o_{x,t}, o_{h,t}, o_{a,t}\right) =$$
$$N\left(s_t; \mu^q_\theta\left(s_{t-1}, o_{x,t}, o_{h,t}, o_{a,t}\right), \sigma^q_\theta\left(s_{t-1}, o_{x,t}, o_{h,t}, o_{a,t}\right)\right)$$

Implemented by deep feed-forward network with 2 hidden tanh-layers and 10 hidden units per layer.
Means are calculated from last hidden layer by a linear layer, standard-deviations by a softplus layer.

# Action




Analogous to Recognition Density:

Instead of optimizing sufficient statistics of action states at every time step, use fixed approximation by neural networks:

$$p_\theta(a_{t+1}|s_t) = N(a_{t+1}; \mu^a_\theta(s_t), \sigma^a_\theta(s_t))$$

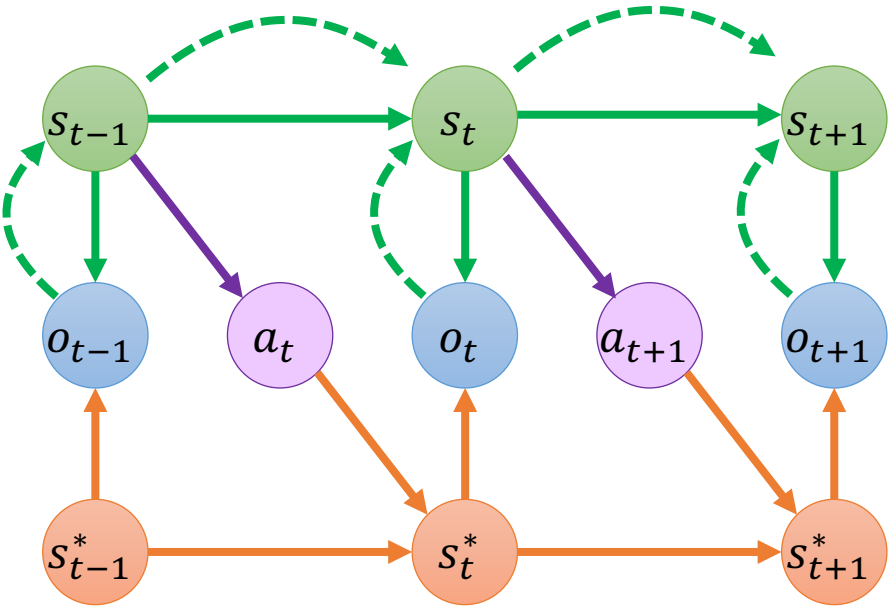
Implemented by deep feed-forward network with 1 hidden tanh-layer (10 hidden units).
Means are calculated from last hidden layer by a linear layer, standard-deviations by a softplus layer.

# Objective Function – Variational Free Energy

$$F(o,\theta) = \sum_{t=1}^{T}[$$
$$< -\ln p_\theta(o_{x,t}|s_t) \; -\ln p_\theta(o_{h,t}|s_t) - \ln p_\theta\left(o_{a,t}\middle|s_t\right) >_{q_\theta(s_t|s_{t-1},o_{x,t},o_{h,t},o_{a,t})}$$
$$+D_{KL}\left(q_\theta\left(s_t\middle|s_{t-1},o_{x,t},o_{h,t},o_{a,t}\right)||p_\theta(s_t|s_{t-1})\right)$$
$$]$$

- We use the closed form of $D_{KL}$ for diagonal Gaussians
- To evaluate this function, we can propagate individual processes, with ***a single sample*** per density.

# Evaluation



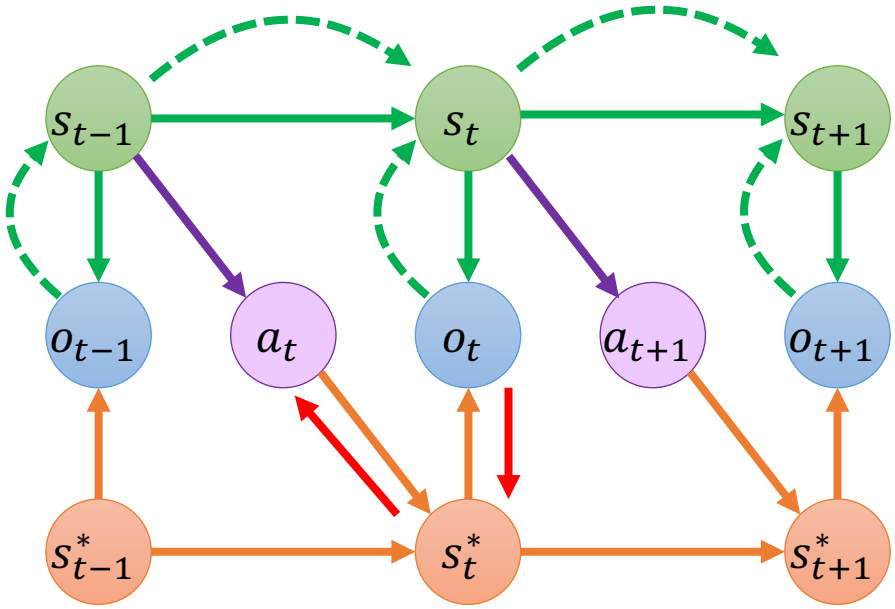

Initialize the free energy estimate with $F = 0$
Initialize $n_F$ agents with $\hat{\mathbf{s}}_0 = (0, ..., 0)$ and $(x_0, v_0) = (-0.5, 0.0)$.
**for each** Agent **do**
  **for** $t = 1, ..., T$ **do**
    Sample an action $\hat{a}_t$ from $p_\theta(a_t|\hat{\mathbf{s}}_{t-1})$
    Propagate the environment using $(x_t, v_t) = \mathbf{R}(x_{t-1}, v_{t-1}, \hat{a}_t)$
    Draw single observations $\hat{o}_{x,t}$ from $p_e(o_{x,t}|x_t)$
    Set observation $\hat{o}_{h,t} = \exp(-\frac{(x_t-1.0)^2}{2 \cdot 0.3^2})$
    Set observation $\hat{o}_{a,t} = \hat{a}_t$
    Draw a single sample $\hat{\mathbf{s}}_t$ from $q_\theta(\mathbf{s}_t|\hat{\mathbf{s}}_{t-1}, \hat{o}_{x,t}, \hat{o}_{h,t}, \hat{o}_{a,t})$
    Calculate $l = -\sum_{i \in \{x,h,a\}} \ln p_\theta(\hat{o}_{i,t}|\hat{\mathbf{s}}_t)$
    Calculate $d = D_{\mathrm{KL}}(q_\theta(\mathbf{s}_t|\hat{\mathbf{s}}_{t-1}, \hat{o}_{x,t}, \hat{o}_{h,t}, \hat{o}_{a,t})||p_\theta(\mathbf{s}_t|\hat{\mathbf{s}}_{t-1}))$
    Increment free energy $F = F + \frac{d+l}{n_p}$
    Carry $\hat{\mathbf{s}}_t$ and $(x_t, v_t)$ over to the next time step.
  **end for**
**end for**
**return** F

$$F(o,\theta) = \sum_{t=1}^{T}[$$
$$< -\ln p_\theta(o_{x,t}|s_t) \; -\ln p_\theta(o_{h,t}|s_t) - \ln p_\theta\left(o_{a,t}\middle|s_t\right) >_{q_\theta(s_t|s_{t-1},o_{x,t},o_{h,t},o_{a,t})}$$
$$+D_{KL}\left(q_\theta\left(s_t\middle|s_{t-1}, o_{x,t}, o_{h,t}, o_{a,t}\right)||p_\theta\left(s_t|s_{t-1}\right)\right)$$
$$]$$

# Optimization

Problem:

We can not directly calculate gradients w.r.t. the parameters of the action function, as we would have to backpropagate through the unknown dynamics of the environment.




Agent does not know true environmental dynamics or their partial derivatives w.r.t. action.

Solution: That's why we talked about **Evolution Strategies** before!

# How to set Goals?

Problem:

If we propagate the agent as is, it will find a comfortable stable state (likely its starting position) and settle there.

Solution:

Instill homeostatic “drive” in terms of very concrete prior about the agents position at the end of a simulation run.

First, represent the relevant quantity explicitly in terms of a hidden variable:

$$q_\theta\left(s_{1,t}\middle|s_{t-1}, o_{x,t}, o_{h,t}, o_{a,t}\right) = N\left(s_{1,t}; 0.1\mathrm{o_{x,t}}, 0.001\right)$$

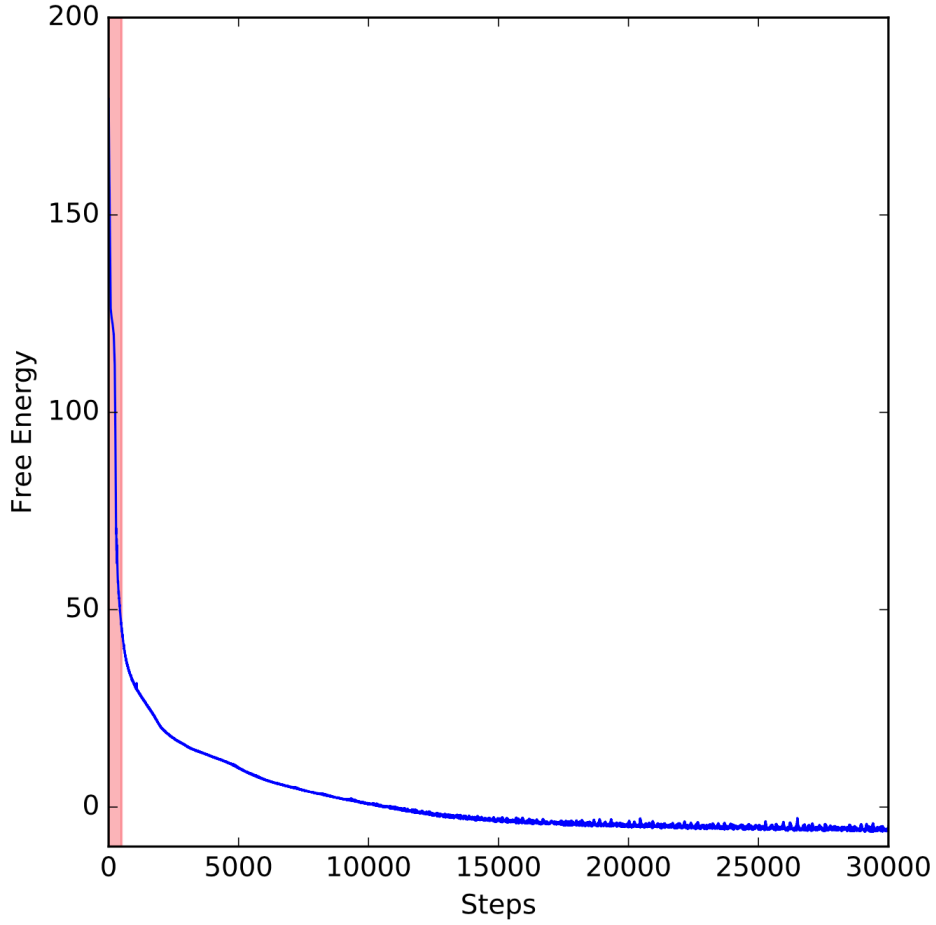
Second, define very explicit expectations of ”where to be” at the end of an episode (30 time steps):

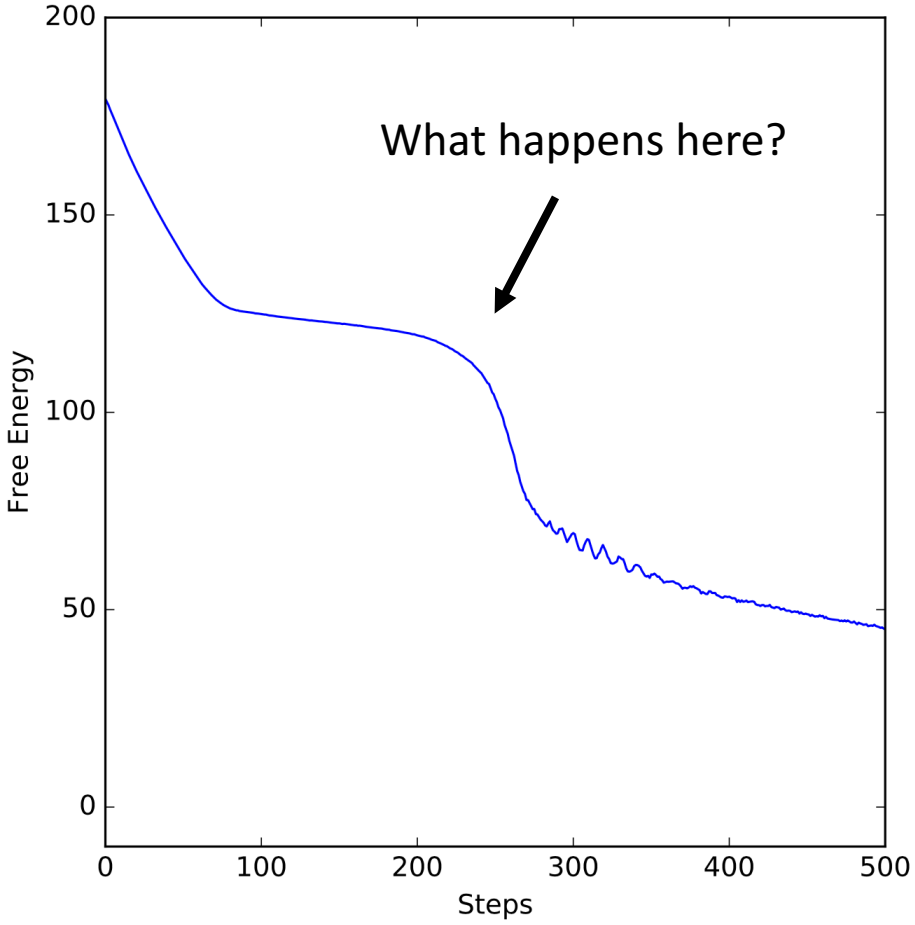
$$p_\theta\left(s_{1,t}\middle|s_{t-1}\right) = N\left(s_{1,t}; 0.1, 0.001\right), \mathrm{t} > 20$$

# Nitty gritty

- Using ADAM with standard parameters.
- Using 10000 samples from the population density.
- Using only one process per parameter-sample to. approximate variational free energy.
- Using desktop computer with NVIDIA Titan (2013) one optimization step takes about 0.4s, taking up about 300 MB of GPU memory.

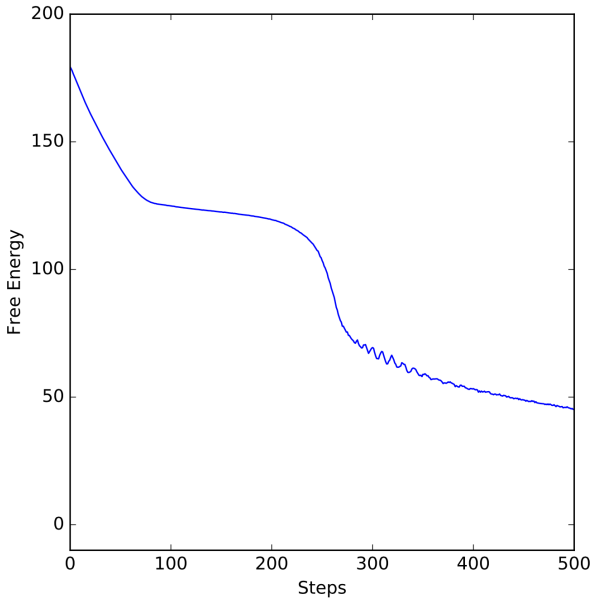
# Results - Convergence

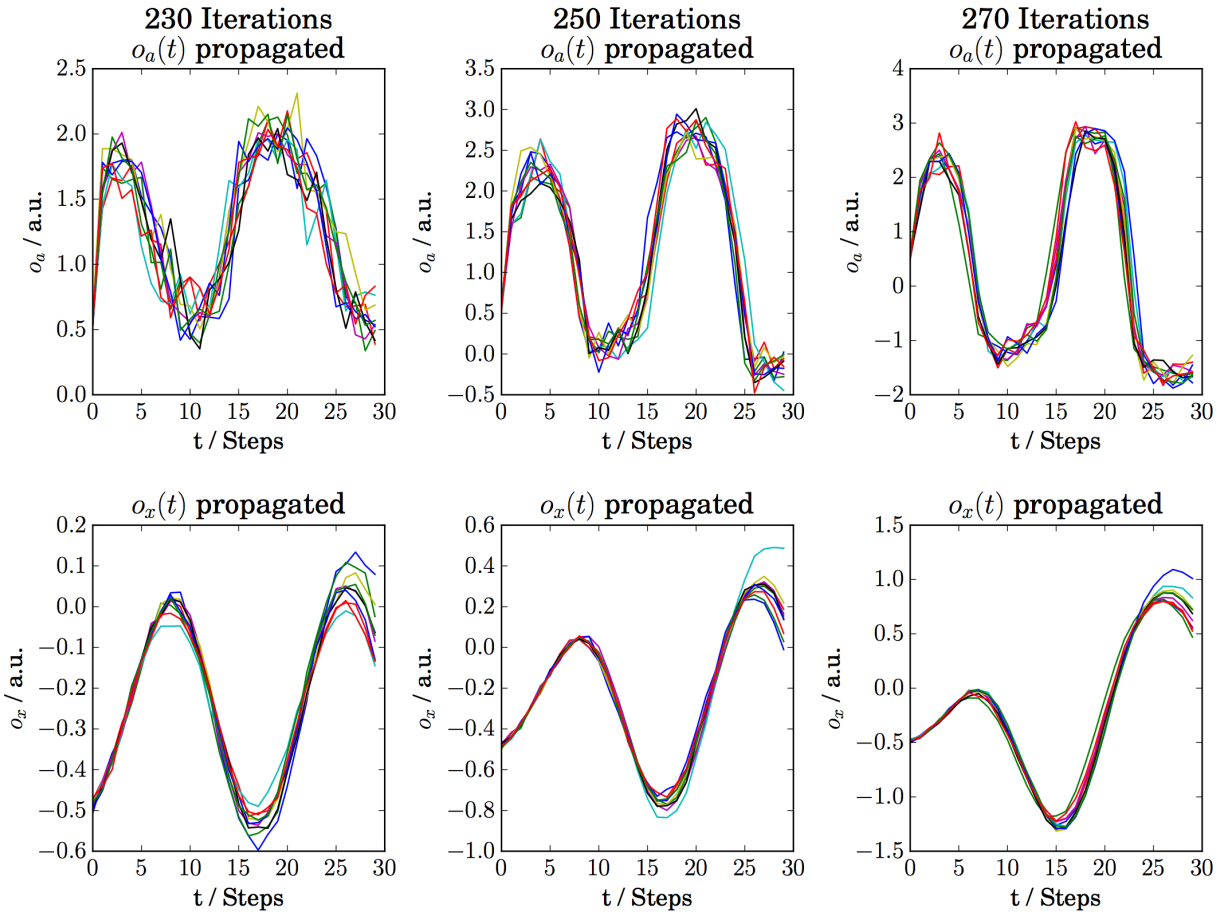


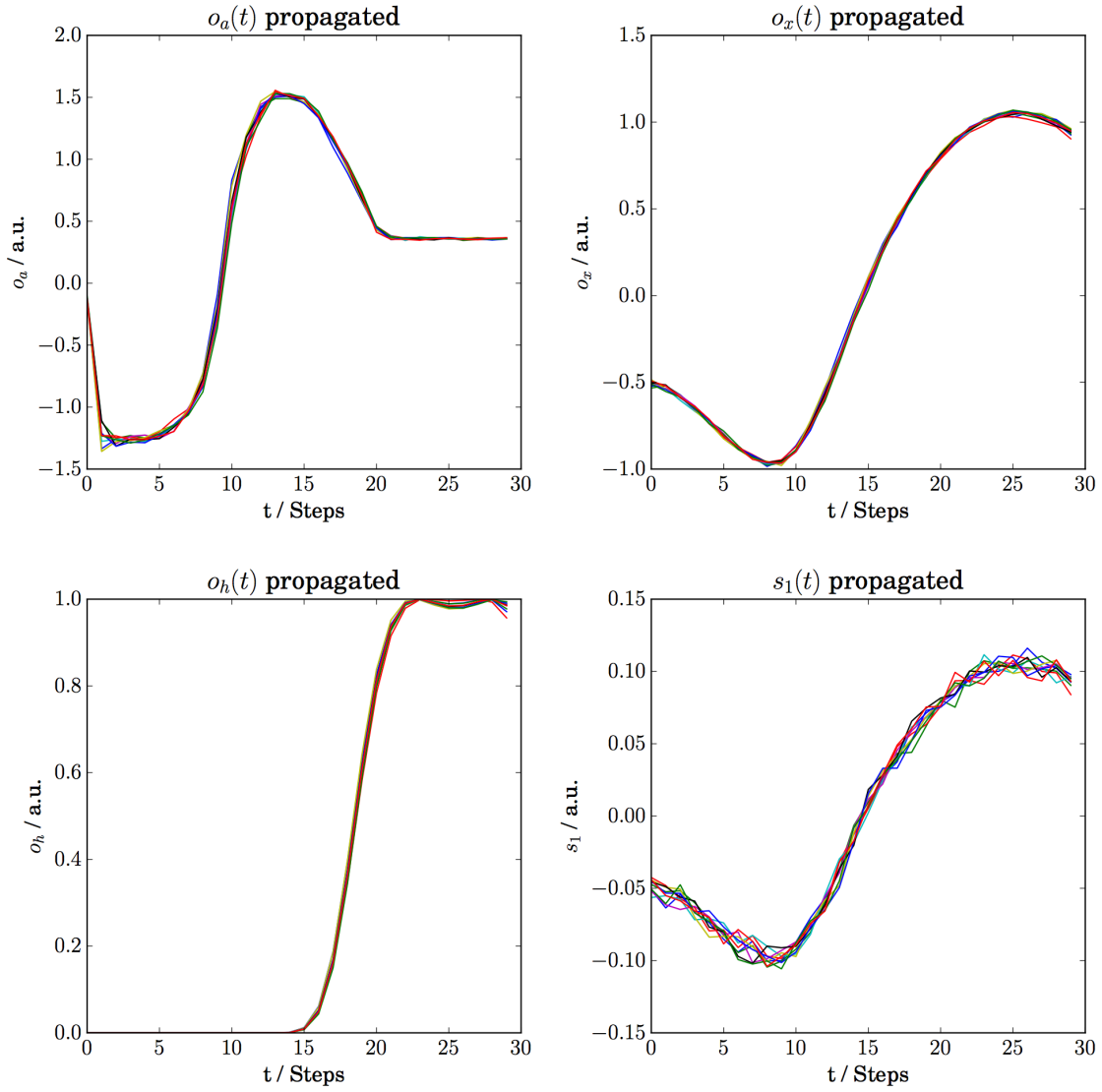
# Results

Agent learns to first move in opposite direction!

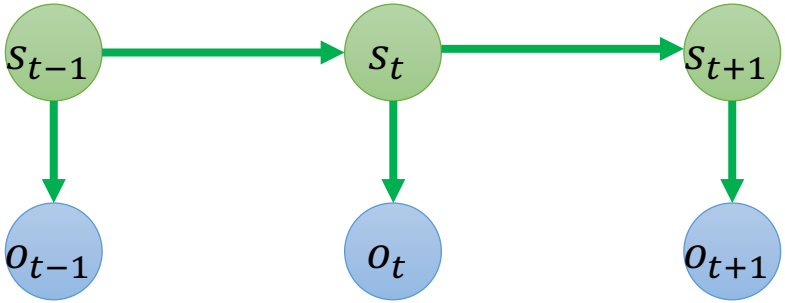


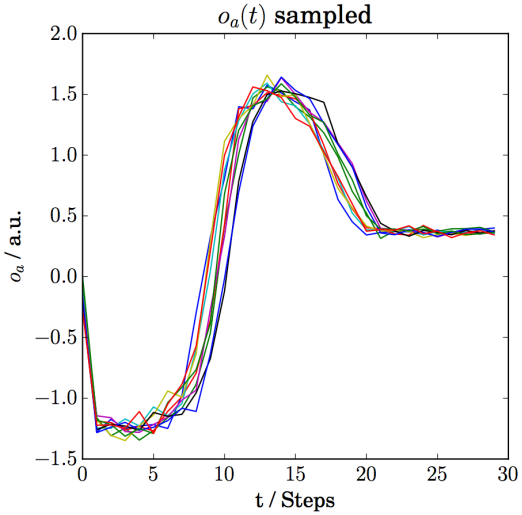
# Results – After convergence

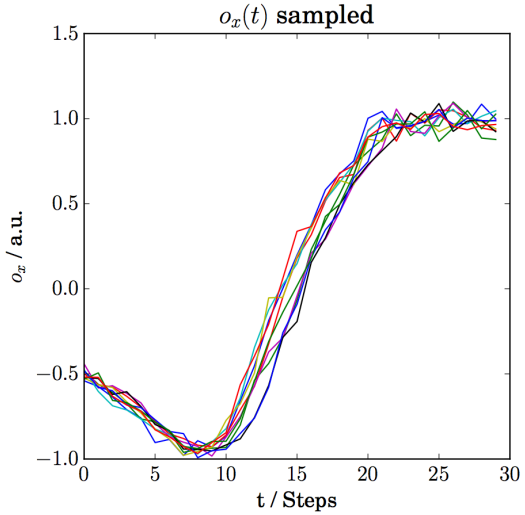


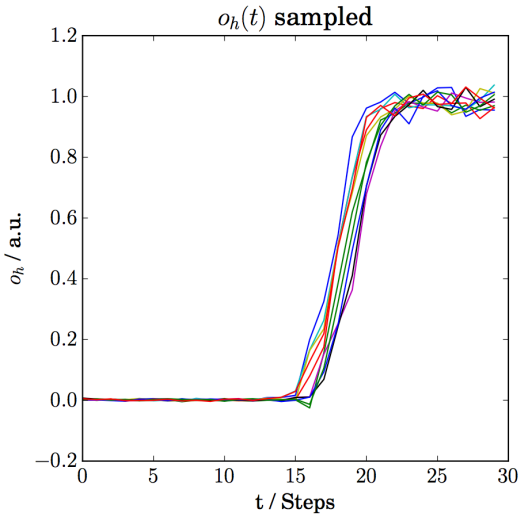
# Results – Sampling from Generative Model

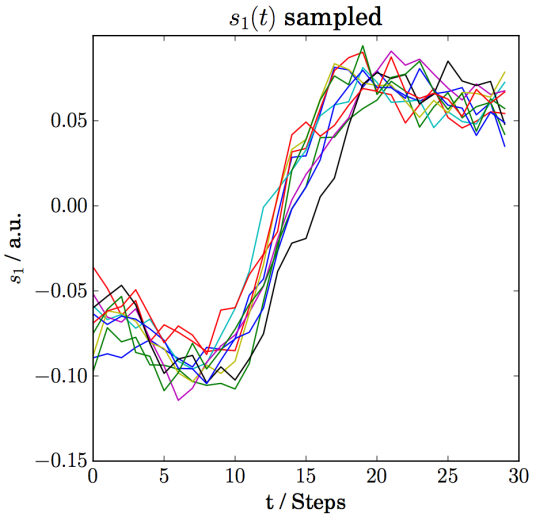



- No interaction with the environment!
- Only one sample per density per process.

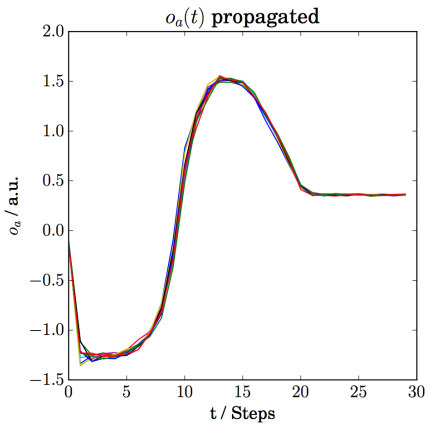


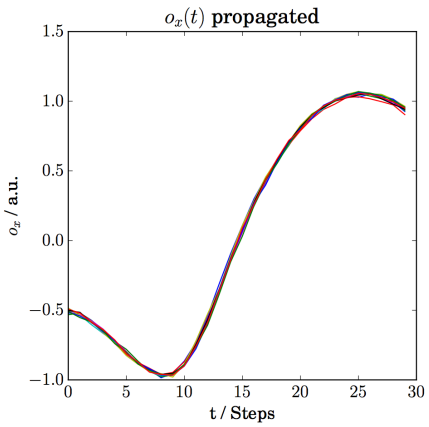



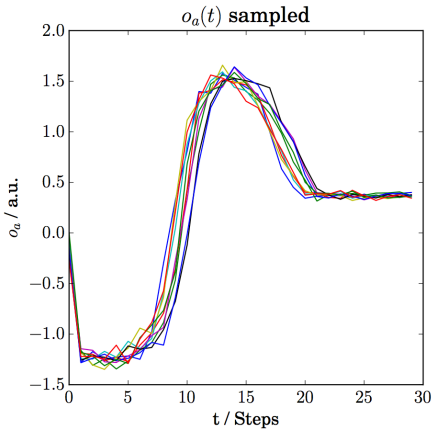



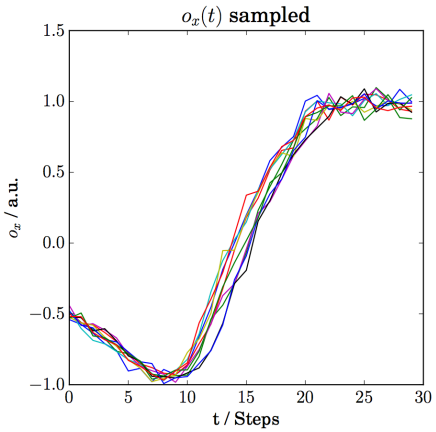



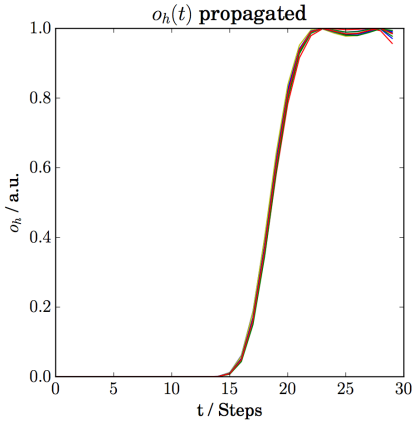
# Results - Comparison

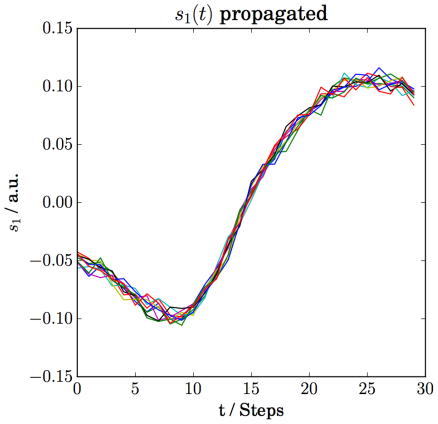


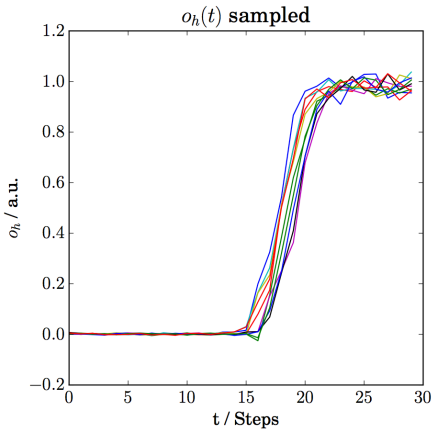



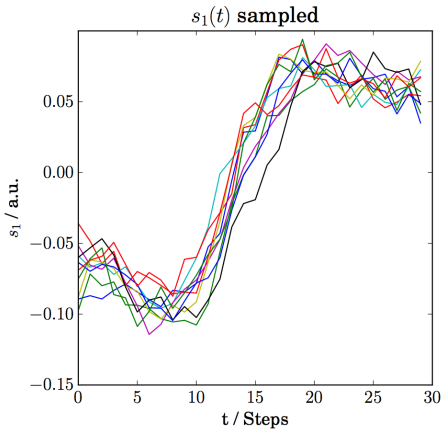



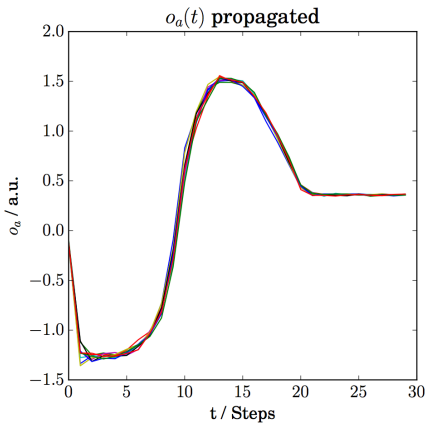



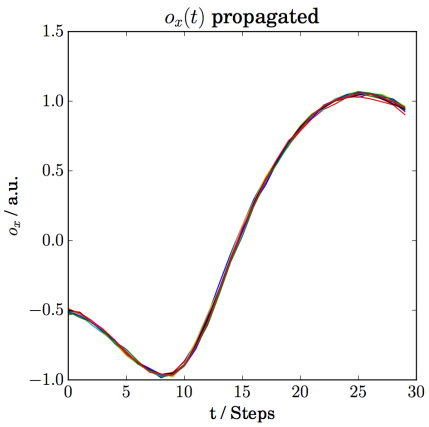



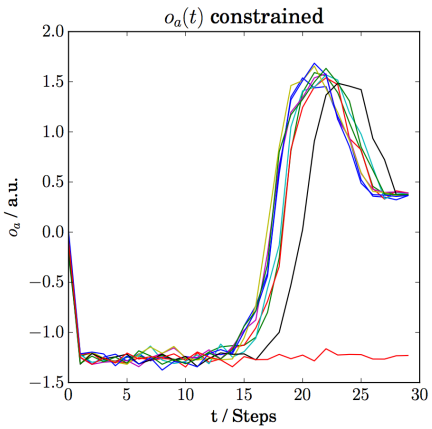



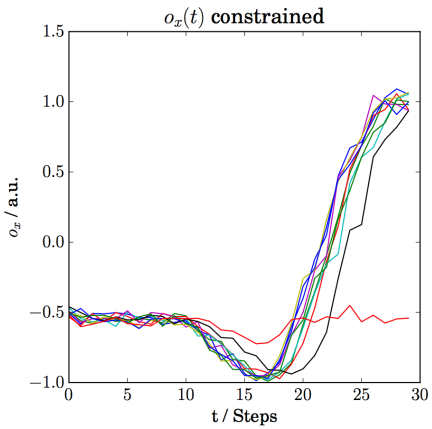
# Constrained Sampling

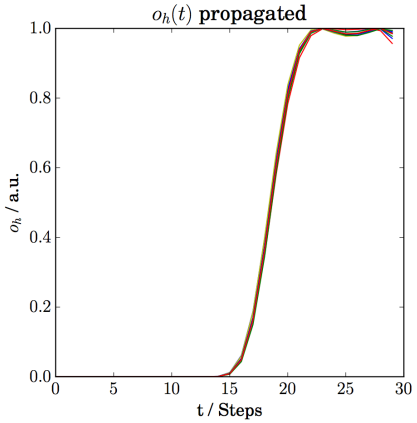
Sampling constrained on average **action** time course, **shifted 10 steps back in time**.

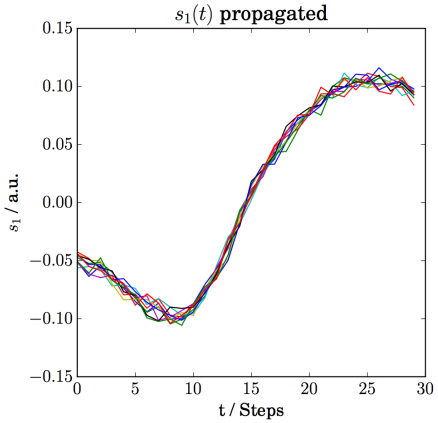


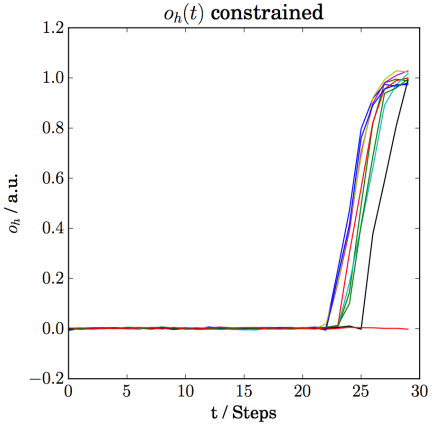



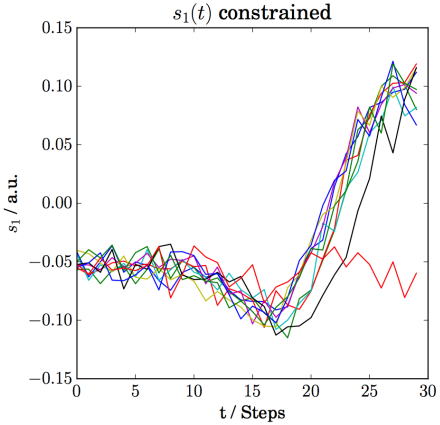



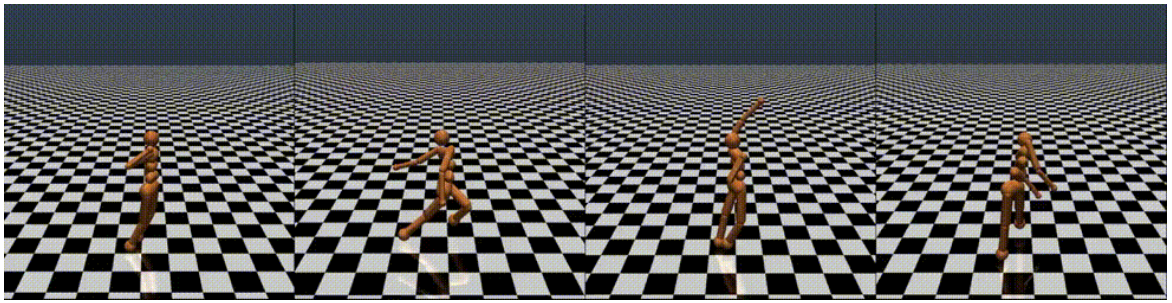

Remember the Imputation of Missing Data using Generative Models (c.f. VAE, Faces, Numbers)

# Take home messages

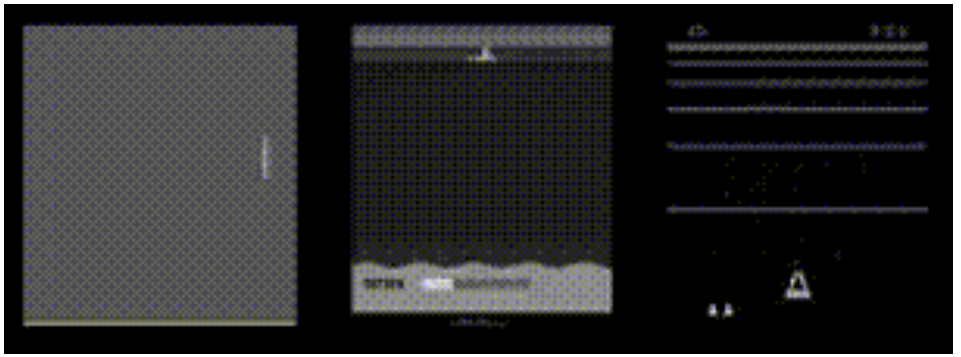
- An agent optimizing the Active Inference objective can reach a not – entirely – trivial goal while concurrently learning a generative model of its sensations.
- Constrained sampling might help to learn more about the agent's beliefs about the world and possible cognitive "biases" that might lead to impaired performance or wrong decisions, without the need for an external simulator.
- By learning a full generative model, which also includes latent representations to predict initially irrelevant sensory channels, the agent should be able to generalize well to new goals in terms of changes in its prior expectations on its latent variables.
- Write-up here: https://arxiv.org/abs/1709.02341
- Full Code here: https://github.com/kaiu85/deepAI_paper

# Next steps

- Show comparable performance to state of the art in complex environments:



Images from: https://blog.openai.com/evolution-strategies/



- Fit more realistic implementation of Active Inference to actual behavioral and neurophysiological data.

# Literature – Variational Autoencoder

- Monograph
  - D.P. Kingma, "Variational Inference and Deep Learning: A New Synthesis", PhD Thesis, http://dpkingma.com
- Papers
  - Basics:
    - Kingma and Welling, Auto-Encoding Variational Bayes, https://arxiv.org/abs/1312.6114
    - Rezende, Mohamed and Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models https://arxiv.org/abs/1401.4082
  - Semi-Supervised Learning:
    - Kingma, Rezende, Mohamed and Welling, Semi-Supervised Learning with Deep Generative Models, https://arxiv.org/abs/1406.5298, Code: https://github.com/dpkingma/nips14-ssl
  - Time-Series Model:
    - Chung et al., Recurrent Latent Variable Model for Sequential Data, https://arxiv.org/abs/1506.02216, Code: https://github.com/jych/nips2015_vrnn, Samples: https://github.com/kastnerkyle/vrnn-samples
  - Normalizing Flows:
    - Rezende and Mohamed, Variational Inference with Normalizing Flows, https://arxiv.org/abs/1505.05770
    - Tomczak and Welling , Improving Variational Auto-Encoders using Householder Flow, https://arxiv.org/abs/1611.09630, Code: https://github.com/jmtomczak/vae_householder_flow
    - Kingma et al., Improving Variational Inference with Inverse Autoregressive Flow, https://arxiv.org/abs/1606.04934, Code: https://github.com/openai/iaf
  - ADAM Optimizer:
    - Ba & Kingma, Adam: A Method for Stochastic Optimization, https://arxiv.org/abs/1412.6980
- Blog Post
  - F. Huszár, Choice of Recognition Models in VAEs: a regularisation view, http://www.inference.vc/choice-of-recognition-models-in-vaes-a-regularisation-view/
- Other implementations:
  - PyTorch Example: https://github.com/pytorch/examples/tree/master/vae

# Literature – Evolution Strategies

- Paper:
  - Salimans et al., Evolution Strategies as a Scalable Alternative to Reinforcement Learning, https://arxiv.org/abs/1703.03864, Code: https://github.com/openai/evolution-strategies-starter
- Blog-Posts
  - OpenAI: https://blog.openai.com/evolution-strategies/
  - F. Huszar:
    - http://www.inference.vc/evolutionary-strategies-embarrassingly-parallelizable-optimization/
    - http://www.inference.vc/evolution-strategies-variational-optimisation-and-natural-es-2/
- Other implementations
  - Very didactic implementation by A. Karpathy: https://gist.github.com/karpathy → nes.py

# Literature – Active Inference

- Papers:
  - Outline of the theory:
    - Friston, Daunizeau, Kilner and Kiebel, Action and behavior: a free-energy formulation, 2012, Biological Cybernetics, Vol. 102(3), 227-260
    - Friston, The free-energy principle: a rough guide to the brain?, 2009, Trends in Cognitive Sciences, Vol. 13(7), 293-301
    - Friston, The free-energy principle: a unified brain theory?, Nature Reviews Neuroscience, 2010, Vol. 11, 127-138
  - Some behavioral evidence:
    - Schwartenbeck, FitzGerald, Mathys, Dolan, Kronbichler and Friston, Evidence for surprise minimization over value maximization in choice behavior, 2015, Scientific Reports, Vol. 5
  - Most recent formulation with review of neurobiological evidence:
    - Friston, FitzGerald, Rigoli, Schwartenbeck and Pezzulo, Active Inference: A Process Theory, Neural Computation, 2017, Vol. 29(1), 1-19
- Karl Friston's Homepage: http://www.fil.ion.ucl.ac.uk/~karl/
- Video Lecture:
  - http://videolectures.net/cyberstat2012_friston_free_energy/

heidelberg.ai

Nov. 27th, 7pm: Accelerating Drug Discovery with Deep Learning

by Dr. Tobias Sikosek