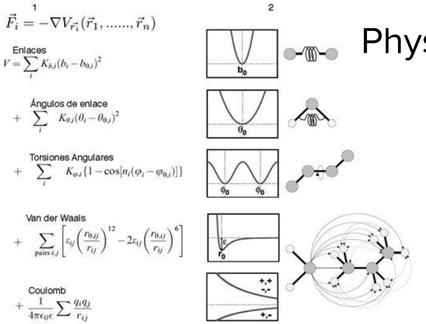
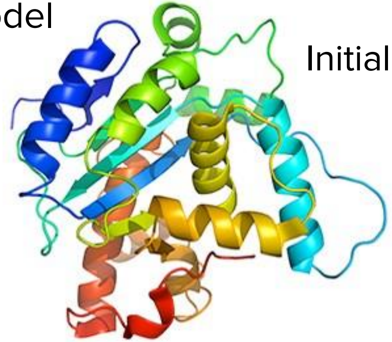
Hyperparameter optimization in RL based multiple path sampling

Samarjeet Prasad
JHMI / NHLBI

Molecular dynamics simulation



Physics Model



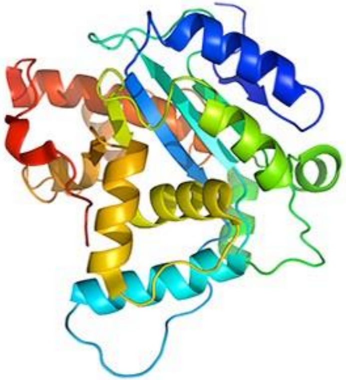
Initial Structure

Molecular Dynamics Simulations
CHARMM

GPU

Newton

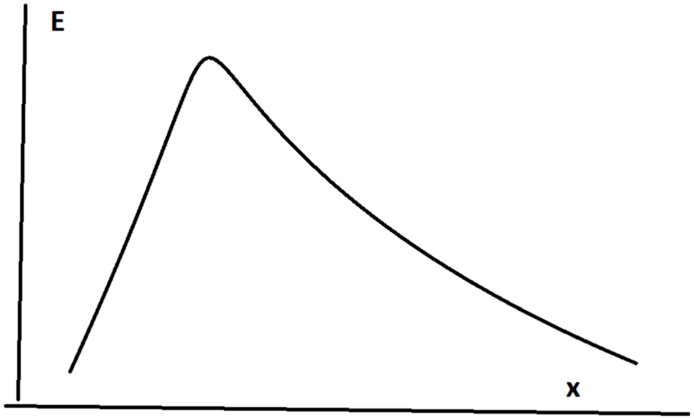
Final Structure

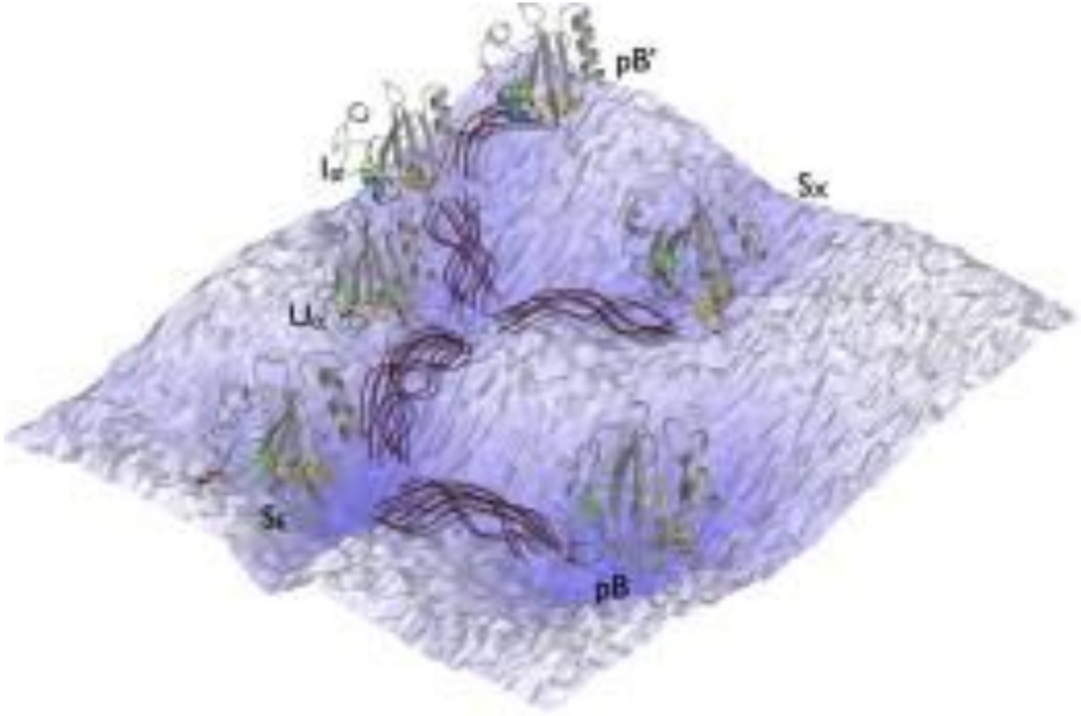


Challenges in MD simulation

- Time scale
    - Individual time step - 10^{-15}s
    - Modes on interest - 10^{-6}s ++

- Energy barriers
    - Difficult to sample barriers as $p \propto \exp(-kE)$





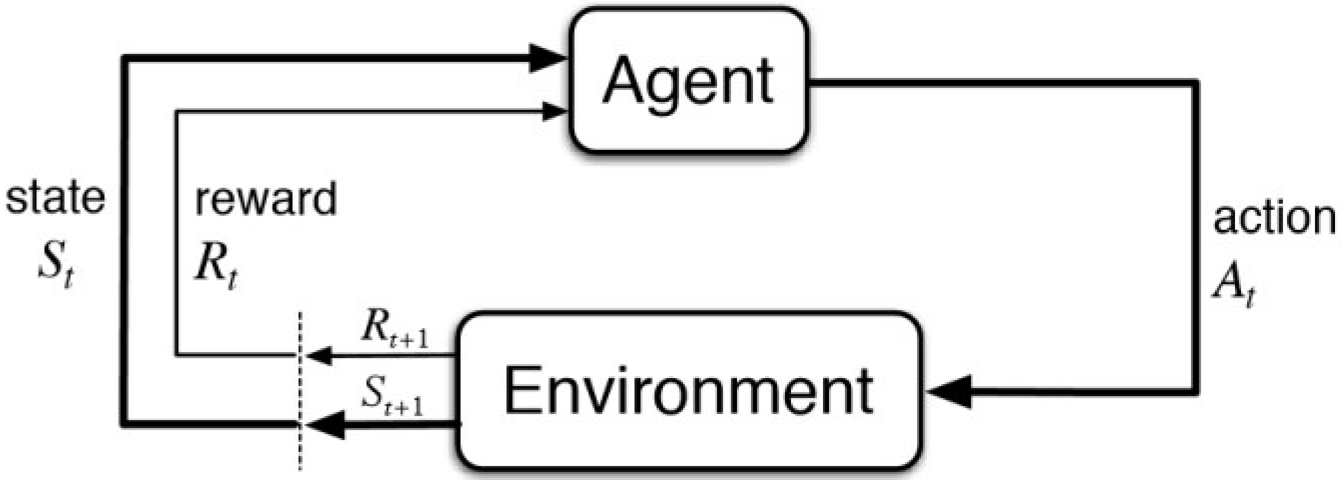
The reaction network of conformational changes of the photo-active yellow protein studied with transition path sampling

Juraszek et. al. Chemical Physics Volume 396, 2 March 2012, Pages 30-44

Path sampling

- Multiple strategies exist to tackle it
  - Reaction path sampling
  - Replica exchange sampling
  - Action-CSA sampling
  - Nudged elastic band methods

Reinforcement learning



Success stories of RL

MENU

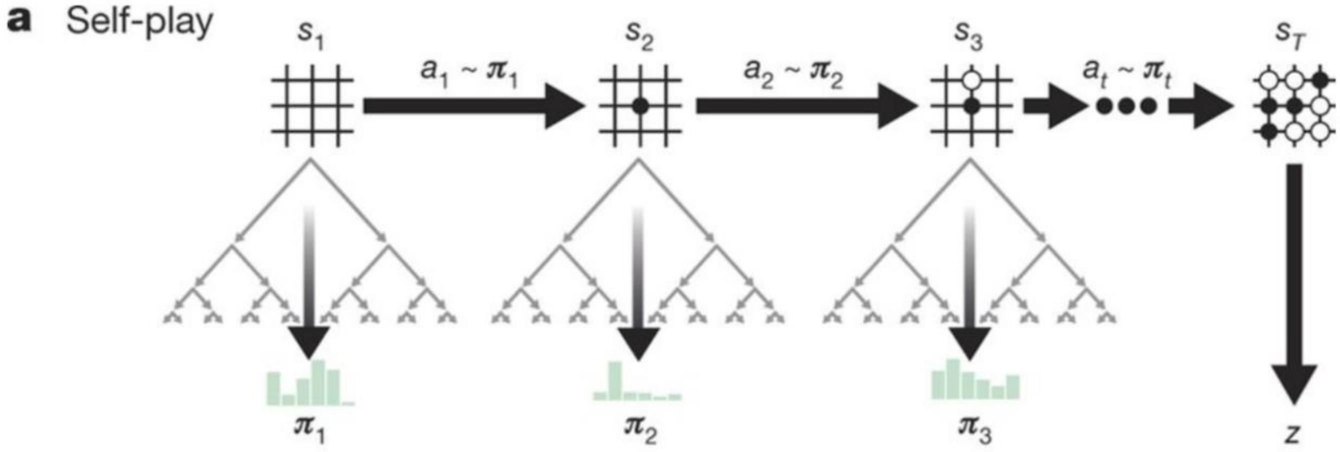
nature
International journal of science

Article | Published: 18 October 2017

Mastering the game of Go without human knowledge

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel & Demis Hassabis

Nature **550**, 354–359 (19 October 2017) | Download Citation



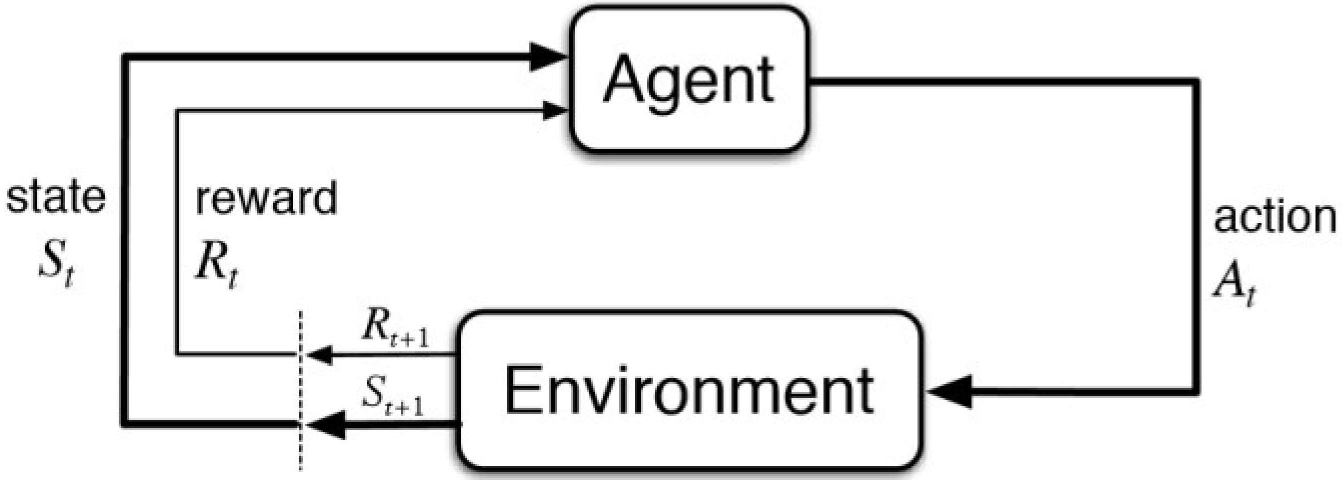
Marrying path sampling and RL

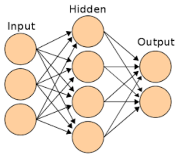
- Continuous action space,
- continuous sample space

- Policy gradient method
  - Optimized policy by performing gradient ascent on the policy parameters
  - Vanilla implementation has very high variance (though pretty low bias)

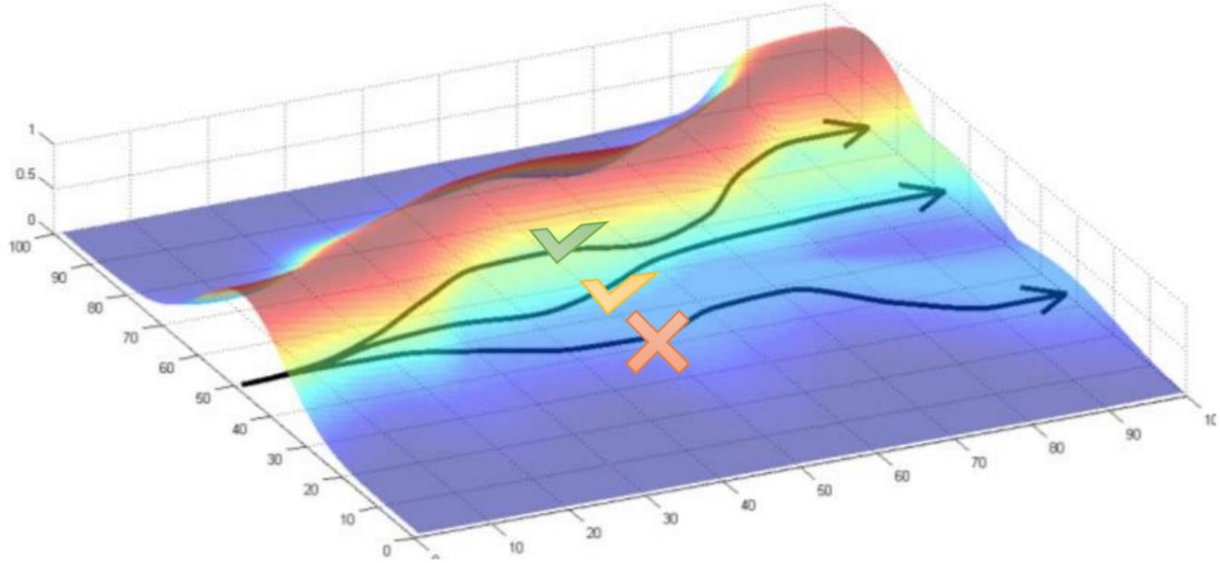
$$g = \mathbb{E}\left[\sum_{t=0}^{\infty} \Psi_t \nabla_\theta \log \pi_\theta(a_t \mid s_t)\right]$$

Architecture of policy gradient





Parameter update



David Silver

Reward for policy gradient

- Reward : action (not to be confused with RL action)

$$\int Lagrangian = \int Kinetic\ energy - Potenial\ energy$$

- On

$$\begin{aligned}\Theta(\mathbf{x}_i; E) &= \mu_A S_{\text{classical}} + \mu_E \sum_{i=0}^{P-1} (E_i - E)^2 \\ &= \mu_A \sum_{i=0}^{P-1} \left[\frac{(\mathbf{x}_i - \mathbf{x}_{i+1})^2}{2\Delta t^2} - V(\mathbf{x}_i) \right] \Delta t + \mu_E \sum_{i=0}^{P-1} \left\{ \left[\frac{(\mathbf{x}_i - \mathbf{x}_{i+1})^2}{2\Delta t^2} + V(\mathbf{x}_i) \right] - E \right\}^2\end{aligned}$$

Hyperparameter tuning

- Used Google's gin-config
  - Decorators added to functions with the parameters to optimize
  - Grid of parameters in a text-file
  - Functions called repeatedly with the hyperparameter settings

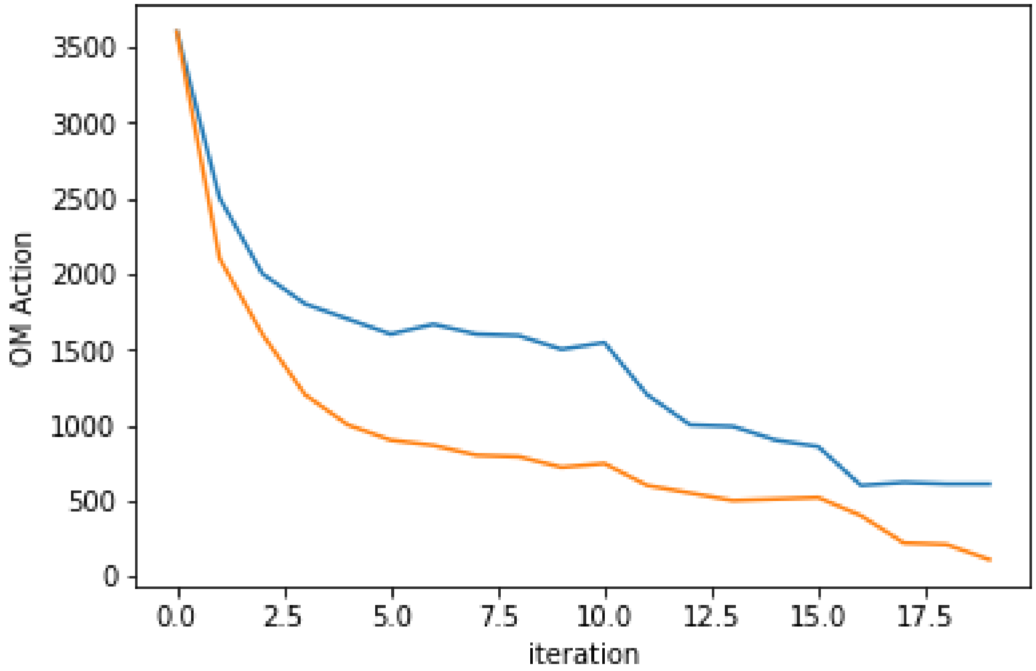
```python
@gin.configurable
class DNN(object):
  # Constructor parameters become configurable.
  def __init__(self,
               num_outputs,
               layer_sizes=(512, 512),
               activation_fn=tf.nn.relu):
    ...

  def __call__(inputs):
    ...
```

```
# Inside "config.gin"
DNN.layer_sizes = (1024, 512, 128)
```

```python
gin.parse_config_file('config.gin')
```

Generalized advantage estimation



Orange : gamma =0.8

Blue : gamma=0.65

Thank you !