**Algorithm** Minibatch stochastic gradient descent training of generative adversatial nets for Pendulum. The number of steps to apply to the discriminator, *k*, is a hyperparameter. I used *k* = 1, the least expensive option, in my experiments.

Sample *t* continue time point {t(0), t (1),…,t(t)} in the pirod of time T

Use ordinary differential equation produce *t* states {s(0), s(1), …,s(t)} as S, each state include current angular and velocity [, *v*]

**for** number of training iterations **do**

**for** *k* steps **do**

* Sample minibatch of *m* states samples {s(0), s(1), …., s(m-1)}as C from S
* For each state in C, move one bit forward {s(1), s(2), …., s(m)}, indicating the next state corresponding to each state in C as C’
* Obtaining generated *m* states {s(1’), s(2’), …, s(m’)} as G form Generator
* Combine each state from C and C’ to {x(0), x(1), …., x(m-1)} as D\_real, each x(i) corresponding to[s(i), s(i+1)]
* Combine each state from C and G to {y(0), y(1), …., y(m-1)} as D\_fake, each x(i) corresponding to[s(i), s(i+1’)]
* Update the discriminator by ascending its stochastic gradient:

**end for**

* Sample minibatch of m states samples {s(0), s(1), …., s(m-1)}as C from S
* Update the generator by descending its stochastic gradient:

**end for**

The gradient-based optimizer is **Adam**.