Supplementary Materials for: Coping With Simulators That Don't Always Return

Andrew Warrington
University of Oxford

Saeid Naderiparizi
University of British Columbia

Frank Wood
University of British Columbia

1 Additional Experimental Information

Here we include additional details on the experimental results presented in the main text.

1.1 Annulus Example

The expressions used for generating the true data is:

$$\delta t = 1, \tag{1}$$
$$t \in [0, \dots, 100], \tag{2}$$
$$x_0 \sim \mathcal{N}(0, 1), \tag{3}$$
$$y_0 \sim \mathcal{N}(0, 1), \tag{4}$$
$$r = \sqrt{x_0^2 + y_0^2}, \tag{5}$$
$$s_0 \sim \mathcal{N}(0, 0.1\sqrt{2}), \tag{6}$$
$$a_0 \leftarrow \arctan\left(\frac{y_t}{y_t}\right), \tag{7}$$
$$\dot{x}_0 \sim -s_0 \times \sin(a_0), \tag{8}$$
$$\dot{y}_0 \sim s_0 \times \cos(a_0), \tag{9}$$
$$x_t \leftarrow x_{t-1} + \delta t \times \dot{x}_{t-1}, \tag{10}$$
$$y_t \leftarrow y_{t-1} + \delta t \times \dot{y}_{t-1}, \tag{11}$$
$$s_t \leftarrow \sqrt{\dot{x}_{t-1}^2 + \dot{y}_{t-1}^2}, \tag{12}$$
$$a_t \leftarrow \arctan\left(\frac{y_t}{y_t}\right), \tag{13}$$
$$\dot{x}_t \leftarrow -s_t \times \sin a_t, \tag{14}$$
$$\dot{y}_t \leftarrow s_t \times \cos a_t, \tag{15}$$
$$\mathbf{y}_t \leftarrow [\mathcal{N}(x_t, 0.1),\ \mathcal{N}(y_t, 0.1)], \tag{16}$$

where we index time by t.

The model used at inference is:

$$\delta t = 1, \tag{17}$$
$$t \in [0, \dots, 100], \tag{18}$$
$$x_0 \sim \mathcal{N}(0, 1), \tag{19}$$
$$y_0 \sim \mathcal{N}(0, 1), \tag{20}$$
$$\dot{x}_0 \sim \mathcal{N}(0, 0.1), \tag{21}$$
$$\dot{y}_0 \sim \mathcal{N}(0, 0.1), \tag{22}$$
$$x_t \leftarrow x_{t-1} + \delta t \times \dot{x}_{t-1} + z_{x_t}, z_{x_t} \sim \mathcal{N}(0, 0.1), \tag{23}$$
$$y_t \leftarrow y_{t-1} + \delta t \times \dot{y}_{t-1} + z_{y_t}, z_{x_t} \sim \mathcal{N}(0, 0.1), \tag{24}$$
$$\dot{x}_t \leftarrow \dot{x}_{t-1} + z_{\dot{x}_t}, z_{\dot{x}_t} \sim \mathcal{N}(0, 0.1), \tag{25}$$
$$\dot{y}_t \leftarrow \dot{y}_{t-1} + z_{\dot{y}_t}, z_{\dot{y}_t} \sim \mathcal{N}(0, 0.1). \tag{26}$$

Simulator failure is then defined as when the change in radius is greater than 0.03 in a single iteration. This effectively constrains the state to orbit in a roughly circular orbit around the origin, with large perturbations to velocity tangentially allowed, but only relatively small perturbations to the velocity in the radial direction.

To compute the variances of the SMC sweep we generate 100 random traces, on traces of length 100. We then perform 100 SMC sweeps per trace, using 100 particles, and compute a pseudo-marginal estimate of the evidence as the product of the expected likelihoods at each observation.

1.2 Bouncing Balls

We simulate the elastic collisions between bouncing balls using the standard Newtonian equations of motion. We use two balls with radius 5 and equal mass, in a square enclosure with size 40. We perturb the position and velocity with Gaussian distributed noise with zero mean and standard deviation 0.5 and 0.1 respectively. The prior over the initial position is uniform in the allowed region, and a zero mean Gaussian over velocity in the x and y directions, with standard deviation $1.0/\sqrt{2}$. When performing state-space inference we observe only the centre of mass of the ball perturbed by Gaussian noise with a standard deviation of 2.0.

In Figure 1a we include the curve of the failure rate as a function of epoch during training, and box plot of the variance of the pseudo-marginal evidence estimate.

To compute the variances of the SMC sweep we generate 100 random traces. We then perform 20 SMC

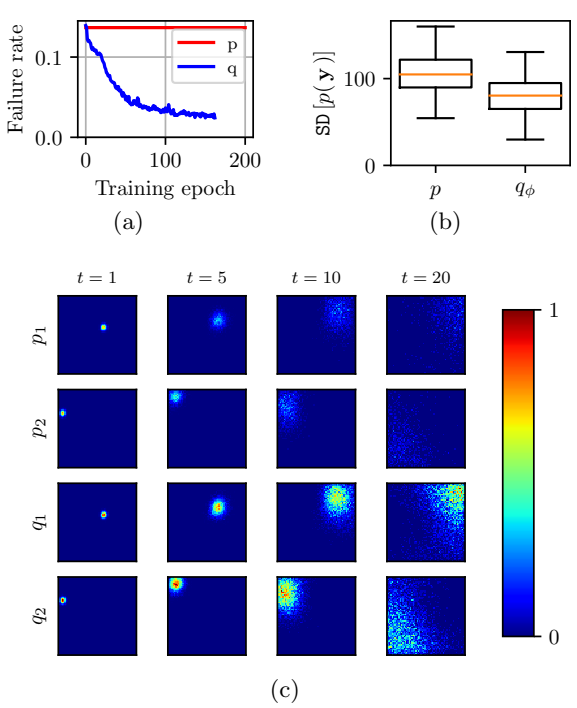

Figure 1: Additional results plots for the bouncing balls example. 1a shows the reduction from approximately 14% rejection under p to 3% rejection under q_ϕ. 1b shows the reduction in the standard deviation of the evidence approximation. The reduction is lower in this example as the system is reasonably "stable" with a low rejection rate. The reduction is still statistically significant, scoring < 0.0001 on a paired t-test. 1c shows the trajectories under p and q_ϕ. Columns are normalized heatmaps plotted at times 1, 5, 10, and 20. The top two rows are the positions of the two balls using p, and the bottom two rows are the positions of the two balls using q_ϕ. For a single starting point, it can be seen that there are many more samples surviving to $t = 20$ under q_ϕ.

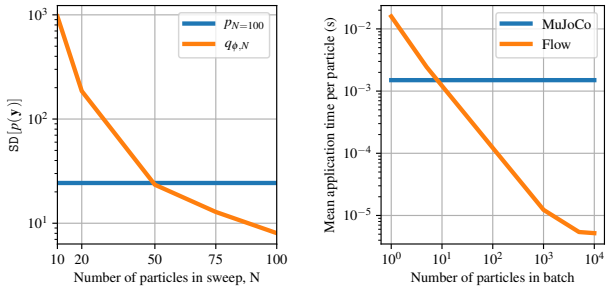

Figure 2: Additional investigation of the computational burden of the tosser example. 2a shows that we need approximately half the number of particles when using q_ϕ to achieve the same variance in the pseudo-marginal evidence approximation when 100 particles are used in conjunction with p. 2b shows the relative computational cost of iteration of the simulator and sampling from the flow. We can see that the computational cost of the flow is less than the cost of the simulator for roughly ten particles or more, as the flow can be sampled in parallel on a GPU, whereas the simulator must be iterated sequentially. For larger batches, the cost of the flow is negligible compared to the cost of the simulator. These results would be further exaggerated for more expensive simulators, such as the WormSim simulator.

sweeps per trace, using 100 particles, and compute the evidence. We also include a figure demonstrating the increased sample diversity under q_ϕ compared with p

1.3 Tosser

We use the configuration "tosser" included in MuJoCo Todorov et al. [2012], only modifying it by removing the second unused bucket. We use completely standard simulation configuration. We introduce the limit on overlap leveraging MuJoCos in-built collision detection, rejecting overlaps above 0.005. Typical overlaps in the standard execution of MuJoCo are below this limit. An integration time of 0.002 is used.

We observe only the x-y position of the capsule with Gaussian distributed noise, with standard deviation 0.1. We perturb the x-y position and velocity of the capsule with Gaussian distributed noise, with standard deviation 0.005 and 0.1 respectively. We perturb the angle and angular velocity of the capsule with Gaussian distributed noise, with standard deviation 0.05 and 0.05 respectively. These values were chosen to be in line with typical simulated values in the tosser example.

We place a prior over the initial position and velocity with standard deviation 0.01 for positions and 0.1 for velocities, and mean equal to their true position.

In this, the state input to the normalizing flow is the position and angle, and derivatives, of the capsule, as well as the state of the actuator. The actuators state is unobserved and is not perturbed under the model. We also input time into the normalizing flow as the control dynamics are not constant with time.

To compute the variances of the SMC sweep we generate 50 random traces. We then perform 20 SMC sweeps per trace, using 100 particles, and compute the evidence.

We also compare the runtime savings using the trained flow in place of the originally specified distribution for

a fixed variance of the pseudo-marginal evidence approximation. The results of this are shown in Figure 2a. We compute the variance of the pseudo-marginal evidence approximation using p for an SMC sweep using 100 particles, shown as a constant line in blue. We then compute the variance of the SMC sweep using 10, 20, 50, 75, and 100 particles when using q_ϕ. We see that the variance under q_ϕ is approximately the same when using 50 particles, corresponding to a halving of the computational effort expended on the simulator. Of course, we then must compare the computational cost of iteration of the simulator to the cost of sampling from the flow, the results of which are shown in Figure 2b. We see that for 50 particles the cost of sampling from the flow is approximately an order of magnitude lower than iteration of the simulator, and as such represents a small computational overhead. As the number of particles grows, this cost drops further due to GPU efficiencies, whereas the cost of the simulator remains constant.

1.4 WormSim

We modify the compiled C++ WormSim Boyle et al. [2012] code only such that it can be interacted with via a Python interface. We apply zero mean Gaussian noise to only the x-y coordinates of the control points of the worm, defining a 98 dimensional flow, where the standard deviation of this noise is 0.000005. The flow is conditioned on the full 510 dimensional state representation of the worm. We normalize the position of the worm such that its head is at the origin before application of the flow.

References

J. H. Boyle, S. Berri, and N. Cohen. Gait modulation in c. elegans: an integrated neuromechanical model. *Frontiers in computational neuroscience*, 6:10, 2012.

E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.