A Learning Details

Hyperparameter	CCN, real	CCN, sim	DiffSim, real	DiffSim, sim
w_{comp}	0.001	0.001	N/A	N/A
w_{diss}	0.1	0.1	N/A	N/A
w_{pen}	100	100	N/A	N/A
w_{res}	1	0.001	1000	1
$w_{\text{res, w}}$	0.1	0	1	0

Table 2: Tuned hyperparameters. Rows for residual norm (w_{res}) and weight ($w_{\text{res, w}}$) regularization only apply for -R variations. Real versus simulated experiments performed best with different residual regularization weights since the simulations featured larger model-to-actual dynamics gaps.

A.1 Model-Based Parameter Learning

For our CCN and CCN-R method, we performed a hyperparameter search to determine the most effective set of weights for balancing the loss terms in (5). See Table 2 for these sets of weights.

A.2 Residual Network Architecture and Regularization

The residual network featured in both CCN-R and DiffSim-R has the same architecture. The first layer takes in the full state of the system and converts the quaternion orientation representation into a 9-vector of the elements of the corresponding rotation matrix, letting the remaining state positions and velocities pass through to layer 2. Beyond the first layer, the network is a fully-connected multi-layer perceptron (MLP) with two hidden layers of size 128. The last layer outputs values in the acceleration space of the system. All activations are ReLU.

We regularized the residual via both output norm regularization and weight regularization, with associated weight hyperparameters w_{res} and $w_{\text{res, w}}$, respectively. See Table 2 for the optimal values. Since the simulation examples were specifically designed to test the capabilities of the residual network, we found the optimal weights for the residual terms were much lower for simulated examples than for the real data. We also note that the optimal residual weights were much higher for DiffSim than for CCN. This is a direct result from the DiffSim residual's attempts to explain some of the contact dynamics, whose accelerations are orders of magnitude larger than the continuous accelerations. Our CCN method avoids this by better containing its residual in the continuous domain, and thus could use lower residual regularization weights.

A.3 End-to-End Network Architecture

The best performing network for the End-to-end baseline is an MLP with 4 hidden layers each of size 256 with Tanh activation. Its input is the full state of the system, and its output is the next velocity. The next configuration is obtained from predicted next velocity with an Euler step (6b).

B Inertia Evaluation Metric Details

A body's set of inertial parameters is $\mathcal{I} = [m, p_x, p_y, p_z, I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}]$. Since true inertia parameter vectors feature values at wildly different scales, the vector s is selected to normalize $\mathcal{I}$ to more equally evaluate all inertial parameter errors. For example, the true inertial parameters for the simulated asymmetric object used in the vortex example are

$$\mathcal{I}_{\text{asym}} = [0.25, 0, 0, 0, 0.00081, 0.00081, 0.00081, 0, 0, 0] \,. \tag{14}$$

Choosing 3.5cm as a reasonable center of mass location distance, the associated s_{asym} normalizer is

$$s_{\text{asym}} = \left[\frac{1}{0.25}, \frac{1}{0.035}, \frac{1}{0.035}, \frac{1}{0.035}, \frac{1}{0.00081}, \frac{1}{0.00081}, \frac{1}{0.00081}, \frac{1}{0.00081}, \frac{1}{0.00081}, \frac{1}{0.00081}\right]. \tag{15}$$