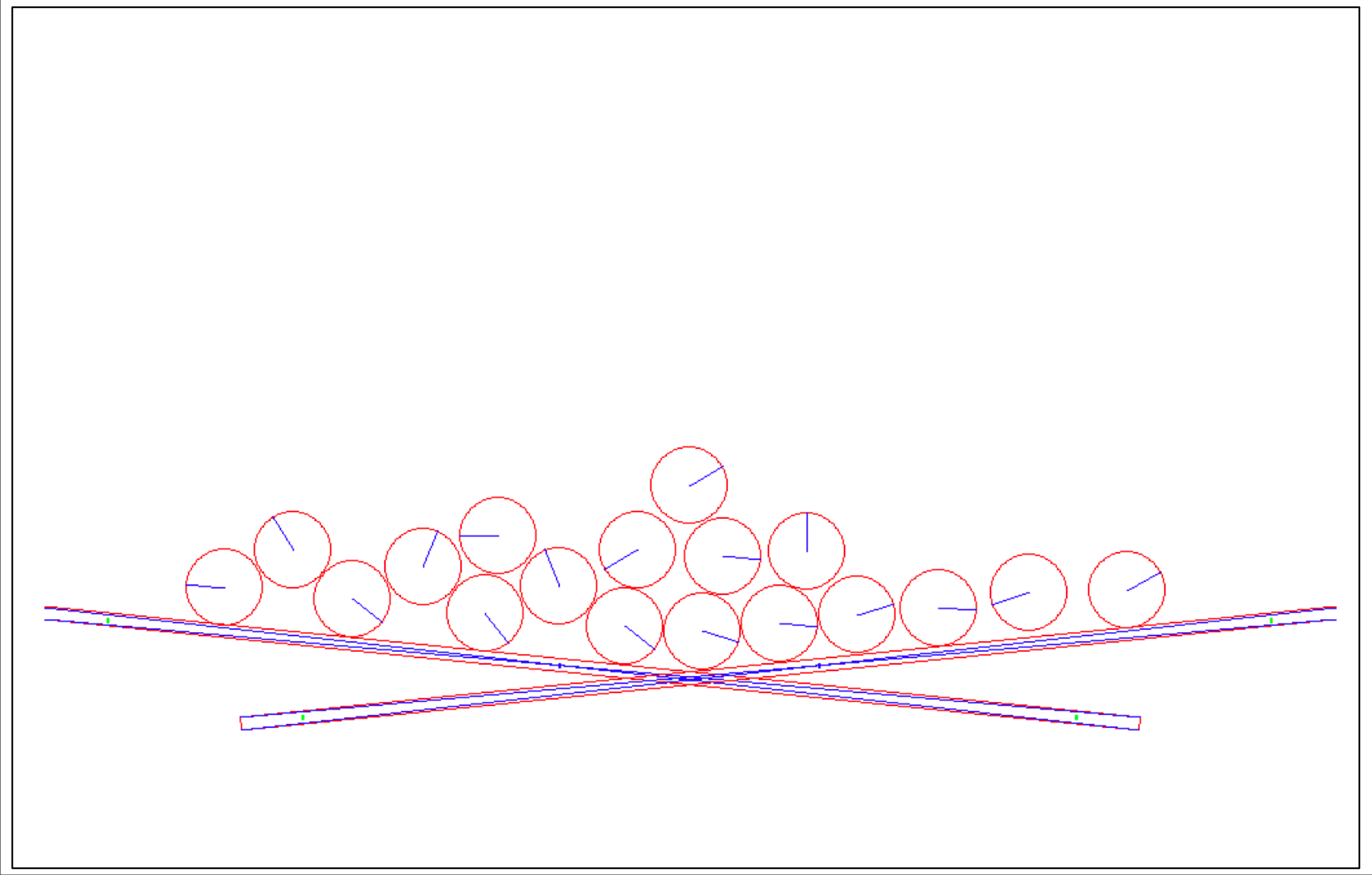
another

- Simulation-based models have strong physical knowledge but are hard-coded, inflexible; learned models are flexible, but require extensive (re)training and precision decays fast
- We develop a differentiable physics engine, which has both precise knowledge of physics and can be embedded in an end-to-end learning system

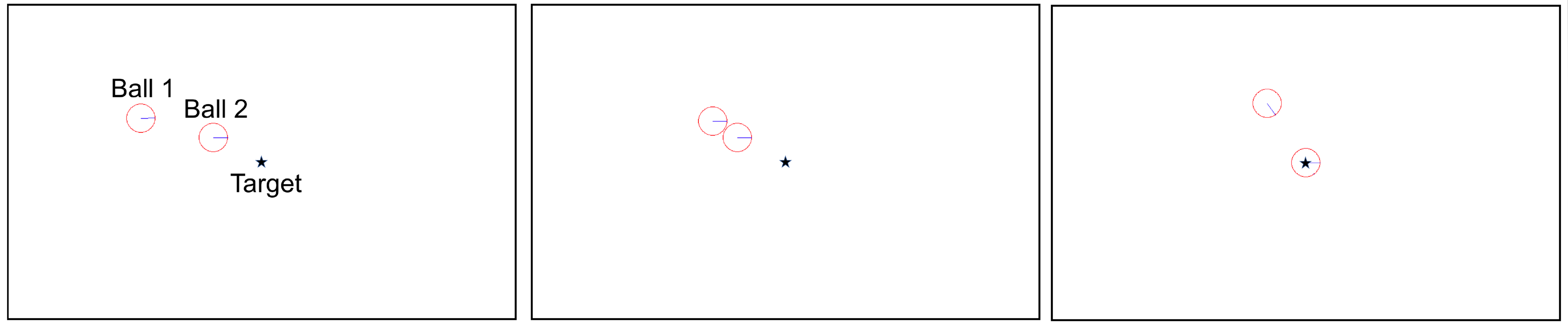
subject to: $\begin{bmatrix} \delta \\ \zeta \end{bmatrix} \geq \mathbf{0}, \begin{bmatrix} \lambda_f \\ \gamma \end{bmatrix} \geq \mathbf{0}, \begin{bmatrix} \delta \\ \zeta \end{bmatrix}^T \begin{bmatrix} \lambda_f \\ \gamma \end{bmatrix} = 0$



- Solvable via a primal dual interior point method [2]

3. Optimization

- Optimizing simulation parameters to minimize a loss function

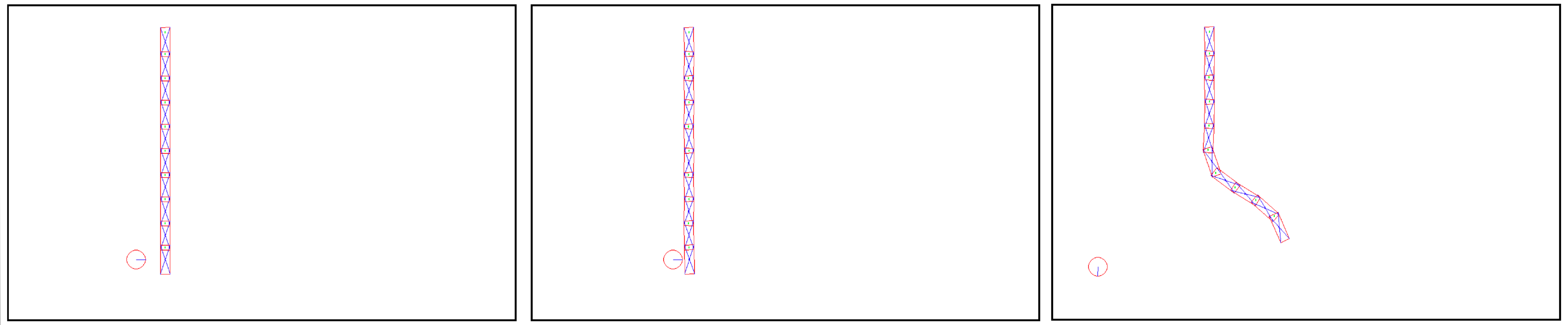


- Choose force applied to Ball 1 in order to minimize distance to goal after 10s
- Random initial force, iteratively updated using the gradient of the distance to goal as follows:

$$f_{i+1} \leftarrow f_i - \alpha \frac{\partial \|pos_{b_2} - pos_{target}\|_2}{\partial f}$$

4. Inference

- Inferring simulation parameters by observation and simulation



- Observe ball and chain collide, infer chain's mass
- Iteratively update estimation using the gradient to minimize mean squared error between simulation and observation:

$$m_{i+1} \leftarrow m_i - \alpha \frac{\partial MSE}{\partial m_i}$$

Carnegie Mellon University

End-to-End Differentiable Physics for Learning and Control

Filipe de Avila Belbute-Peres[1], Kevin Smith[2], Kelsey Allen[2], Josh Tenenbaum[2], J. Zico Kolter[1]

1. Carnegie Mellon University 2. Massachusetts Institute of Technology

Motivation

- Simulation-based models have strong physical knowledge and can learn efficiently, but are hard-coded, inflexible; learned models are flexible, but require extensive (re)training and predictive precision decays fast
- We develop a differentiable physics engine, which has both precise knowledge of physics and can be embedded in an end-to-end learning system

The Engine

- Rigid body dynamics framed as an LCP:

$$\begin{bmatrix} 0 \\ 0 \\ a \\ \sigma \\ \zeta \end{bmatrix} - \begin{bmatrix} \mathcal{M} & -\mathcal{J}_e & -\mathcal{J}_c & -\mathcal{J}_f & 0 \\ \mathcal{J}_e & 0 & 0 & 0 & 0 \\ \mathcal{J}_c & 0 & 0 & 0 & 0 \\ \mathcal{J}_f & 0 & 0 & 0 & E \\ 0 & 0 & \mu & -E^T & 0 \end{bmatrix} \begin{bmatrix} v_{t+h} \\ \lambda_e \\ \lambda_c \\ \lambda_f \\ \gamma \end{bmatrix} = \begin{bmatrix} \mathcal{M}v_t + hf \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{subject to: } \begin{bmatrix} a \\ \sigma \\ \zeta \end{bmatrix} \geq \mathbf{0}, \begin{bmatrix} \lambda_c \\ \lambda_f \\ \gamma \end{bmatrix} \geq \mathbf{0}, \begin{bmatrix} a \\ \sigma \\ \zeta \end{bmatrix}^T \begin{bmatrix} \lambda_c \\ \lambda_f \\ \gamma \end{bmatrix} = 0$$

- Solvable via primal-dual interior point method
- Cheap analytic gradients at solution (for some loss ℓ):

$$\frac{\partial \ell}{\partial q} = -d_x \qquad \frac{\partial \ell}{\partial \mathcal{M}} = -\frac{1}{2}(d_x x^T + x d_x^T)$$

$$\frac{\partial \ell}{\partial m} = D(z^\star) d_z \qquad \frac{\partial \ell}{\partial G} = -D(z^\star)(d_z x^T + z d_x^T)$$

$$\frac{\partial \ell}{\partial A} = -d_y x^T - y d_x^T \qquad \frac{\partial \ell}{\partial F} = -D(z^\star) d_z z^T$$

where

$$x := -v_{t+dt} \qquad q := -\mathcal{M}v_t - dt f_t \qquad s := \begin{bmatrix} a \\ \sigma \\ \zeta \end{bmatrix}$$

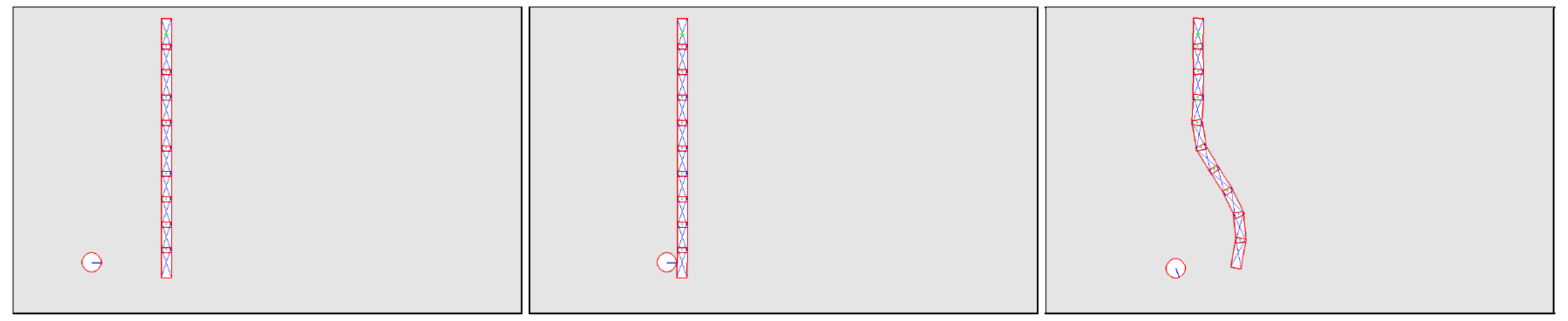
$$y := \lambda_e \qquad A := \mathcal{J}_e \qquad F := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & E \\ \mu & -E^T & 0 \end{bmatrix}$$

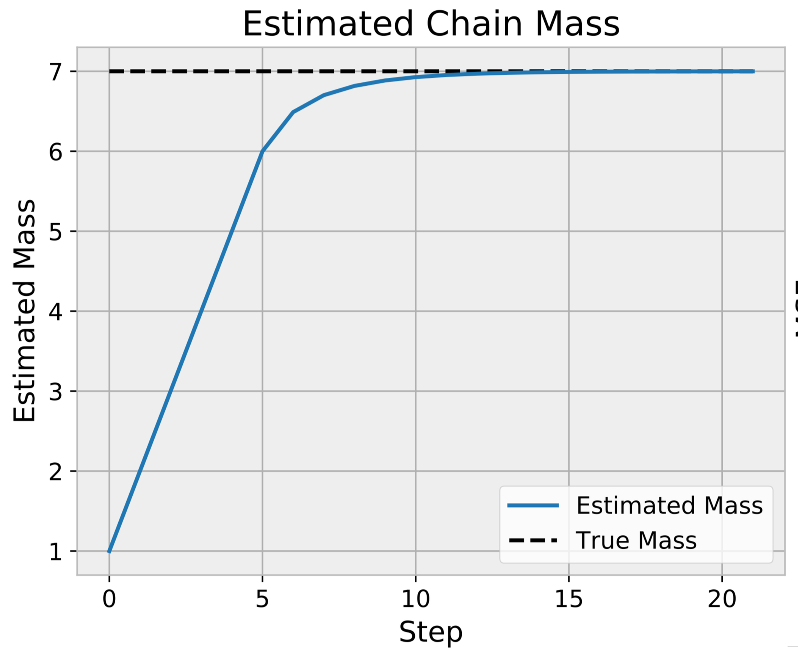
$$z := \begin{bmatrix} \lambda_c \\ \lambda_f \\ \gamma \end{bmatrix} \qquad G := \begin{bmatrix} \mathcal{J}_c & 0 \\ \mathcal{J}_f & 0 \\ 0 & 0 \end{bmatrix} \qquad m := \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}$$

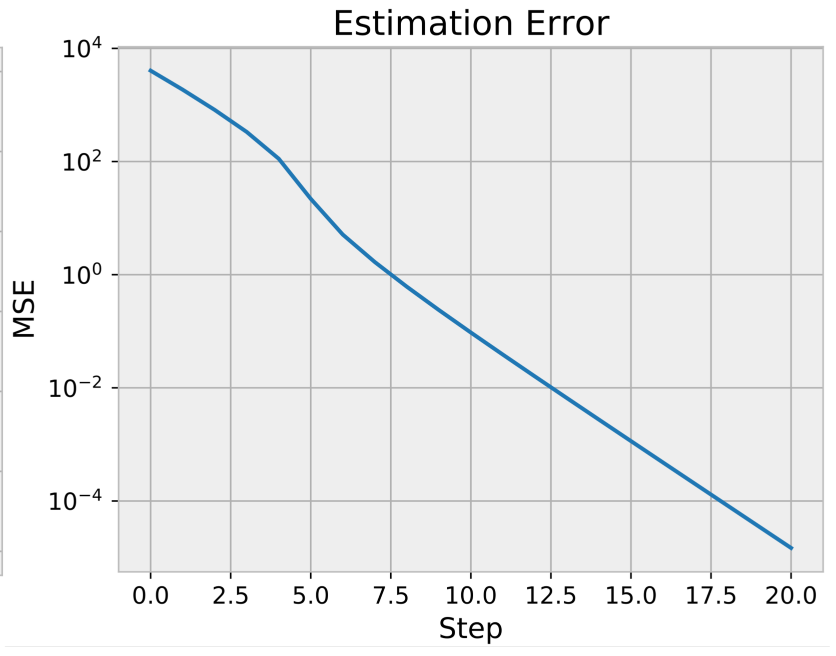
$$\begin{bmatrix} d_x \\ d_z \\ d_y \end{bmatrix} := \begin{bmatrix} \mathcal{M} & G^T & A^T \\ D(z^\star)G & D(Gx^\star + Fz^\star - m) + F & 0 \\ A & 0 & 0 \end{bmatrix}^{-T} \begin{bmatrix} \left(\frac{\partial \ell}{\partial x^\star}\right)^T \\ 0 \\ 0 \end{bmatrix}$$

Parameter learning

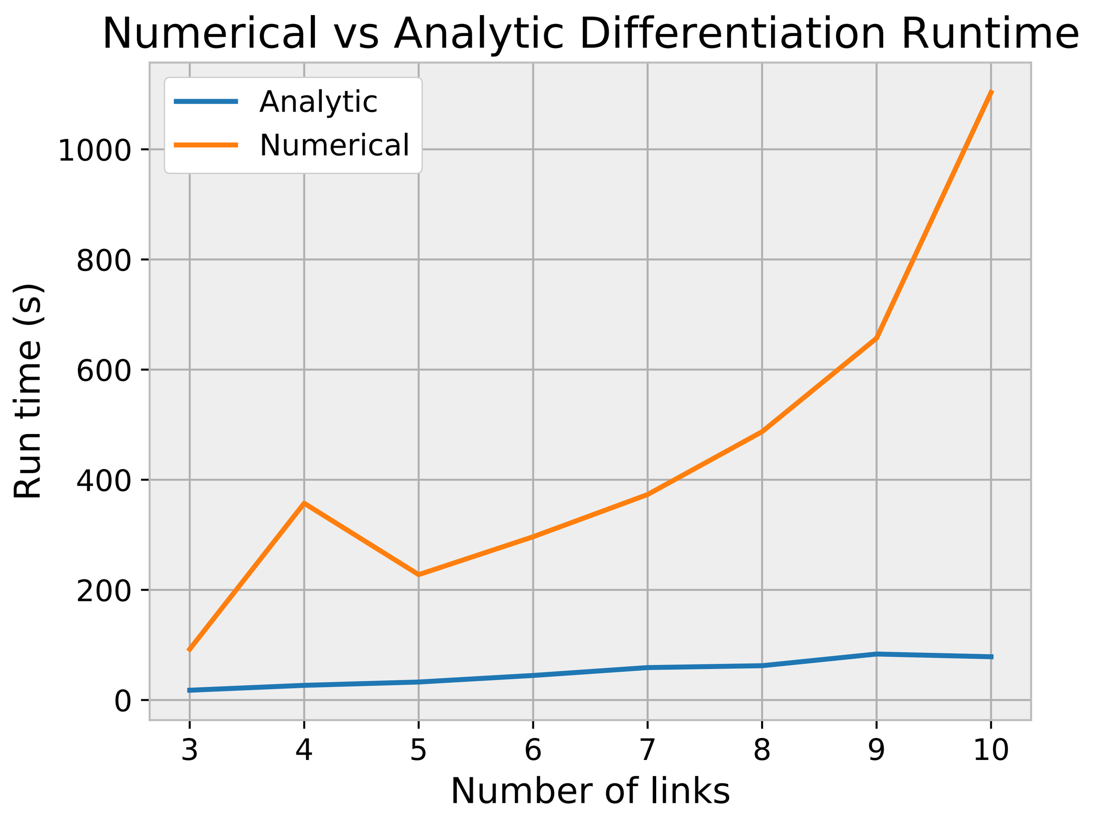
- A ball of known mass hits a chain. Positions of the objects are observed for 10s. Task is inferring the mass of the chain





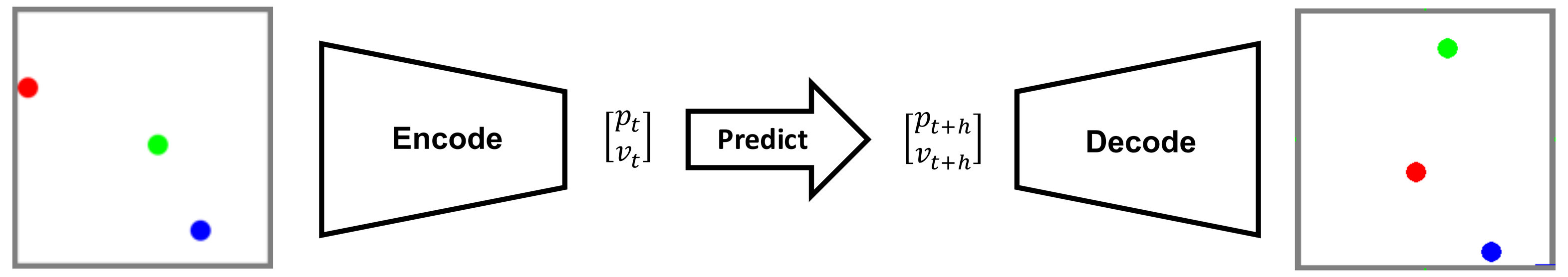


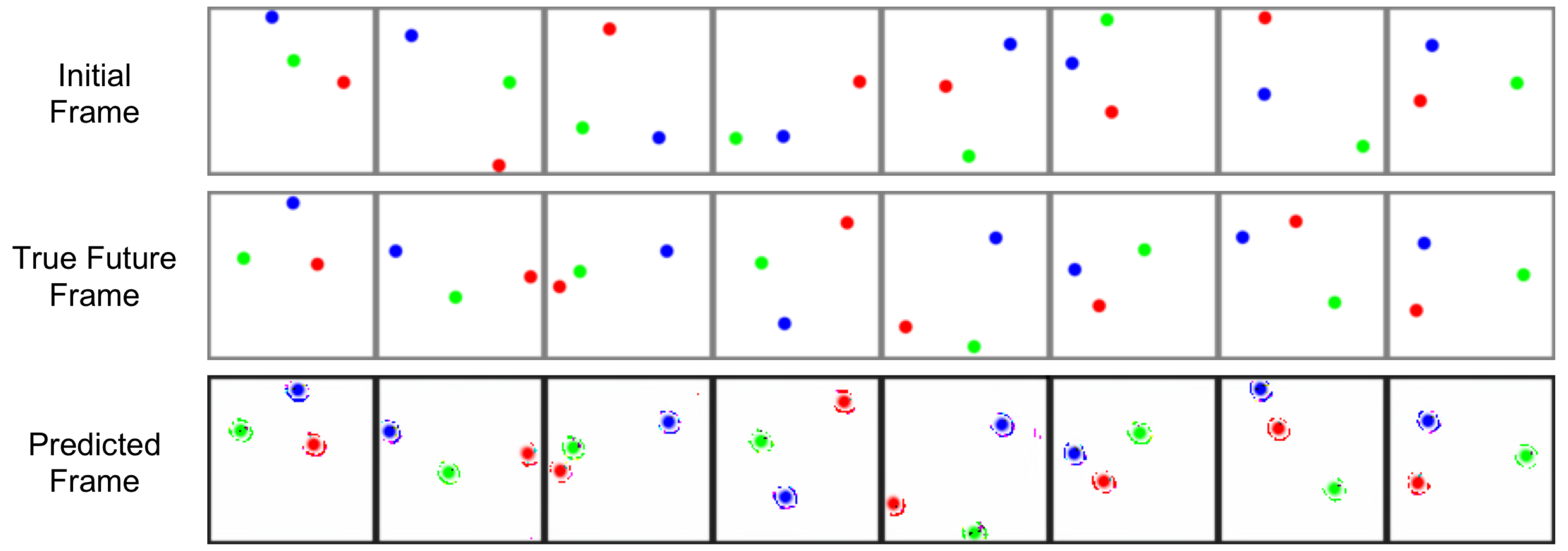
- Runtime is compared using the engine's analytic gradients and numerical gradients (finite-differences)



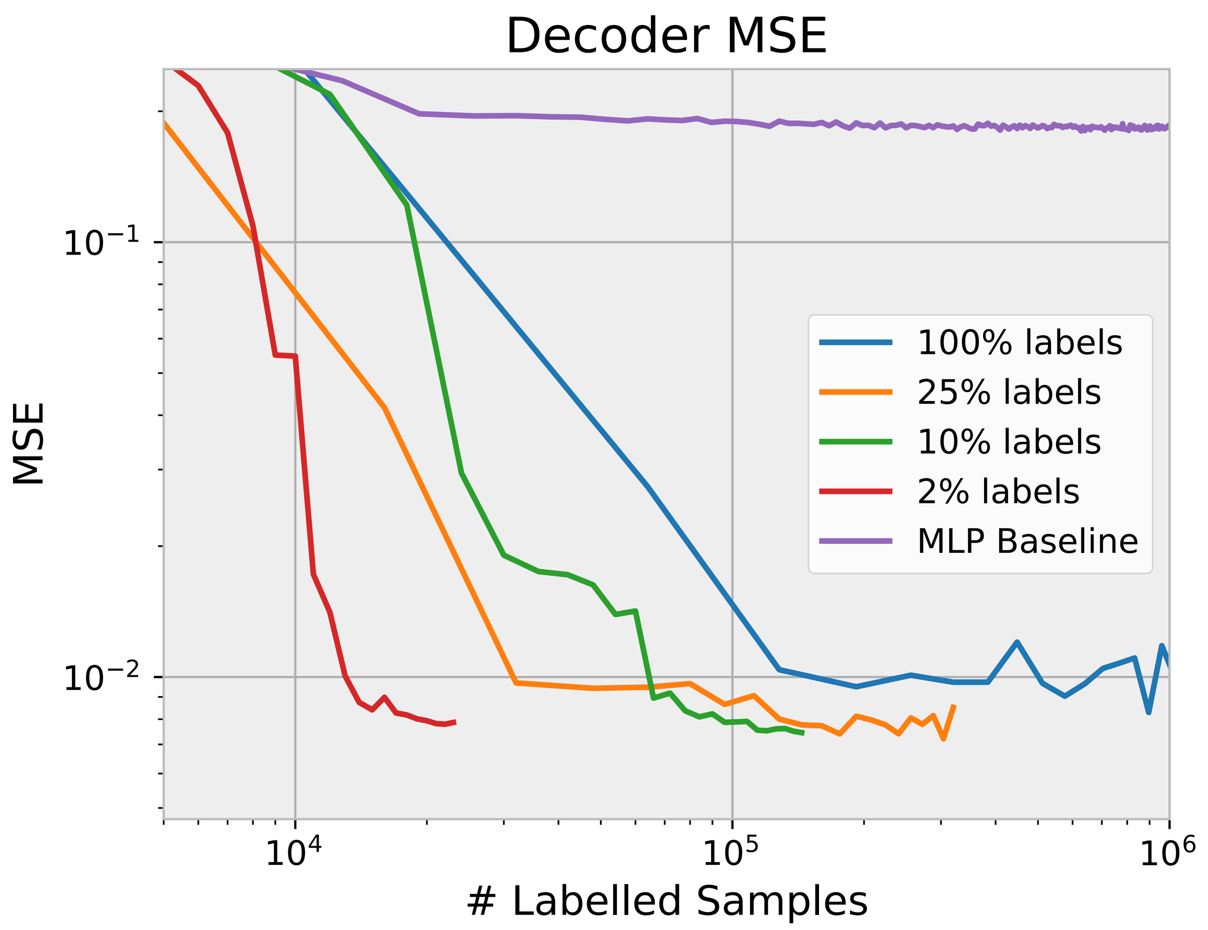
Visual prediction

- After observing 3 frames of a billiard ball-like scene, predict positions 10 frames into the future
- Autoencoder architecture. *Encoder* maps frames into physical predictions. *Engine* steps physics into the future. *Decoder* draws image from physics state



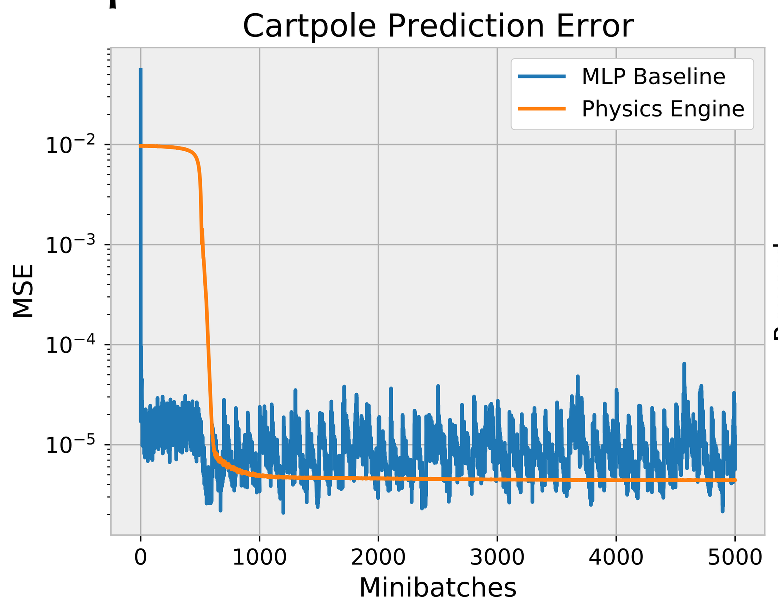


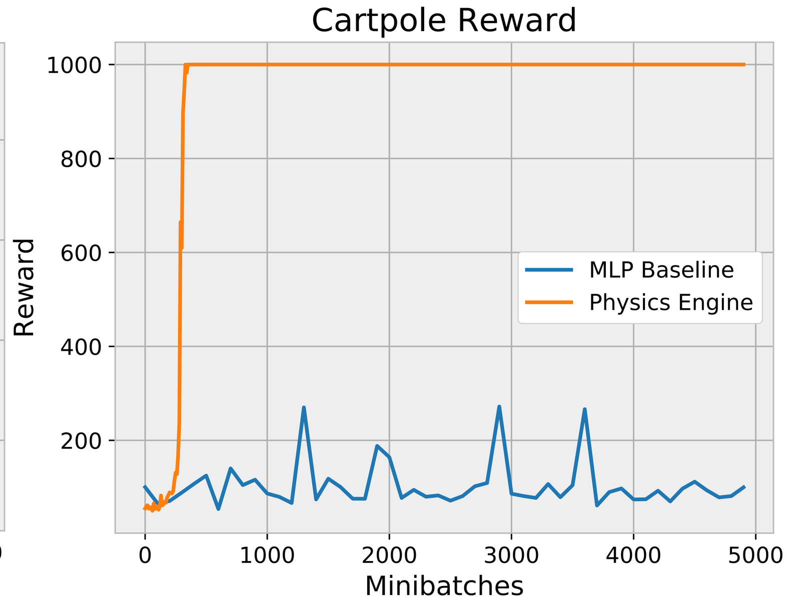
- Physics structure embedded into the model allows for learning with few labeled samples

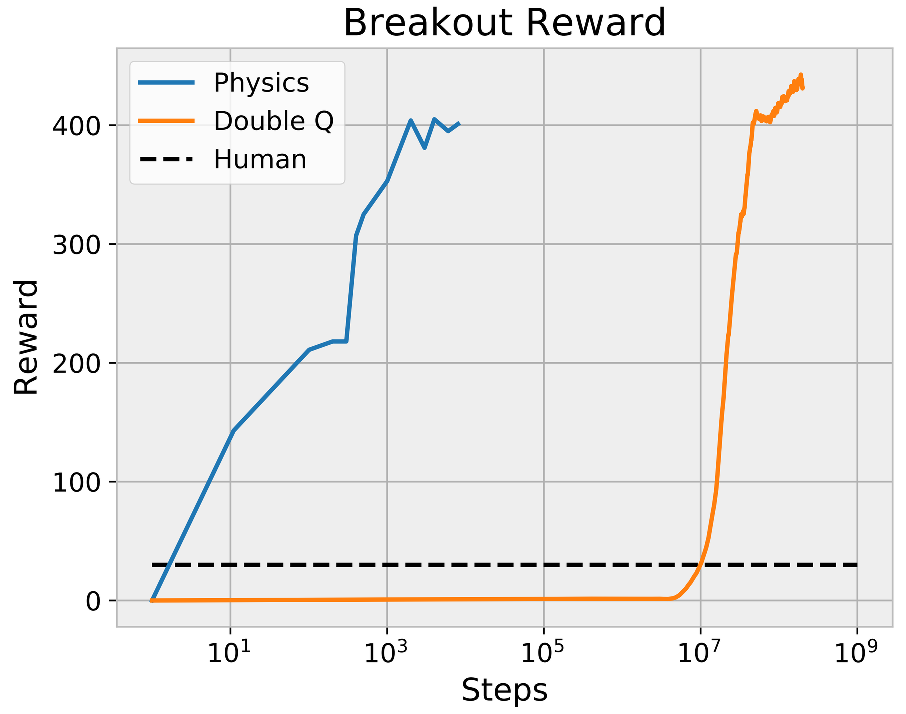


Control

- Since the physics engine is differentiable, we use it in conjunction with iLQR for control in the *Cartpole* and the Atari *Breakout* tasks
- Control performance is measured as simulation parameters are learned







Conclusion

- Unlike similar previous work, we have described a physics engine that provides analytical gradients by differentiating the solution to the physics LCP.
- This system contributes to a recent trend of incorporating components with structured modules into end-to-end learning systems such as deep networks.