Learning Contact Dynamics using Physically Structured Neural Networks

Andreas Hochlehnert
Centre for Artificial Intelligence
University College London

Alexander Terenin
Department of Mathematics
Imperial College London

Steindór Sæmundsson
Department of Computing
Imperial College London

Marc Peter Deisenroth
Centre for Artificial Intelligence
University College London

Abstract

Learning physically structured representations of dynamical systems that include contact between different objects is an important problem for learning-based approaches in robotics. Black-box neural networks can learn to approximately represent discontinuous dynamics, but they typically require large quantities of data and often suffer from pathological behaviour when forecasting for longer time horizons. In this work, we use connections between deep neural networks and differential equations to design a family of deep network architectures for representing contact dynamics between objects. We show that these networks can learn discontinuous contact events in a data-efficient manner from noisy observations in settings that are traditionally difficult for black-box approaches and recent physics inspired neural networks. Our results indicate that an idealised form of touch feedback—which is heavily relied upon by biological systems—is a key component of making this learning problem tractable. Together with the inductive biases introduced through the network architectures, our techniques enable accurate learning of contact dynamics from observations.

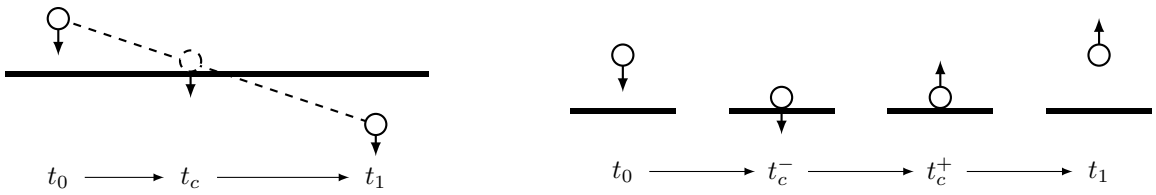
1 Introduction

Deep-learning-based models have accomplished remarkable achievements in a myriad of fields in recent years, ranging from image processing to text generation to reinforcement learning. These models are increasingly being applied to model physical systems, in areas ranging from fluid dynamics (Kutz, 2017) to robotics (Viereck et al., 2018; Lutter et al., 2020; Alvarez et al., 2020). Though neural networks are good at approximating general classes of functions, they often struggle to learn invariant properties of physical systems, such as the conservation of energy or momentum (Greydanus et al., 2019) and other qualitative properties. A rapidly-growing line of work (Raissi, 2018; Greydanus et al., 2019; Lutter et al., 2020; Cranmer et al., 2020; Sæmundsson et al., 2020) has thus focused on how to introduce *inductive biases* into these networks to enable them to learn more accurate models from less data.

One particular kind of physical phenomenon of great interest to areas such as robotics is *contact dynamics*, which describes how collisions between different objects affect the evolution of the system. For example, many of the basic actions that could be relevant for a robot, e.g. walking, jumping or grasping, involve discontinuous contact events. Accurately modelling these dynamics, and related downstream phenomena, such as friction, is a crucial step towards enabling the creation of robots that can learn to interact with unknown objects in the real world. Additionally, stronger inductive biases can improve the data efficiency of such

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Code available at: https://github.com/libeanim/contact-symplectic-integrator-network.



(a) Find the contact time t_c.

(b) Calculate true trajectory.

Figure 1: Example integration scheme for contact dynamics that enforces constraints exactly, in the context of a bouncing ball. Initially, the ball is time-stepped until a contact with the floor is detected through interpenetration at time t_1. Then, the trajectory is (a) linearly interpolated to find the *contact time* t_c where contact occurs between the ball and floor. Finally, the contact state at time t_c is calculated, a transfer of momentum between t_c^- and t_c^+ is performed, and the ball is time-stepped as usual to time t_1.

models (Deisenroth et al., 2015; Greydanus et al., 2019; Lutter et al., 2020; Cranmer et al., 2020; Sæmundsson et al., 2020). Due to the presence of non-linear and non-smooth behaviour, accurate multi-body contact dynamics are widely considered notoriously difficult to compute (Cirak and West, 2005), due to challenges with numerically enforcing non-interpenetration constraints and other issues. A robust and mature literature in physics and numerical analysis for handling these issues has developed in recent decades (Jean, 1999; Cirak and West, 2005; Leyendecker et al., 2008).

By bringing these ideas together with the rapidly-developing literature on neural ordinary differential equations (E, 2017; Haber and Ruthotto, 2017; Chen et al., 2018; Ruthotto and Haber, 2018) and physically-inspired neural networks (Raissi, 2018; Greydanus et al., 2019; Lutter et al., 2020; Sæmundsson et al., 2020), we study the problem of *learning* unknown contact dynamics from data. Our work is based on the approach of Sæmundsson et al. (2020), which derives neural network architectures by discretising the variational principle underlying the physical equations of motion under study. Specifically, we propose specialised neural network architectures for modelling contact dynamics. These architectures combine discretisation schemes designed for contact dynamics with flexible parameterised networks for efficiently and accurately learning system behaviour. We study these networks under different scenarios, develop schemes to ensure accurate learning of dynamics, and demonstrate empirically that the addition of an *idealised touch feedback sensor*—rarely explicitly considered in deep learning, but widely utilised by biological systems—significantly improves model performance.

This suggests that the precise details of what hardware sensors the robot has available and how the learning problem is formulated to utilise those sensors, are both likely to have a significant impact on machine learning performance and should be studied further. This includes understanding the effect of different forms of touch feedback, such as observed tactile sensors, and inferred feedback using proprioceptive sensors (Rotella et al., 2018; Ortenzi et al., 2016). Our work provides a starting point for addressing these questions within a physically structured deep learning framework.

2 Contact Dynamics

Here we briefly review the mathematical formulation of contact dynamics. The state of a physical system is defined through position-velocity pairs $(\boldsymbol{q}, \dot{\boldsymbol{q}})$. From analytical mechanics, the trajectories of a physical dynamical system are assumed to be a stationary point of the action functional

$$S(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \int_{t_0}^{t_1} L(\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t)\,\mathrm{d}t, \tag{1}$$

where L is the Lagrangian—typically, the difference between kinetic and potential energy. By the d'Alembert–Lagrange principle, at instances where there is no contact, these trajectories follow the *Euler-Lagrange equations*

$$\frac{\partial L}{\partial \boldsymbol{q}} - \frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{0}. \tag{2}$$

These equations possess a number of important physical properties, such as conservation of energy, momentum, and phase volume. A long line of work in numerical analysis has built efficient numerical integration schemes for these equations by maintaining these properties under discretisation (Marsden and West, 2001). We are particularly interested in *symplectic integrators*, which conserve phase volume exactly and conserve energy and momentum to a high order of accuracy (Sanz-Serna, 1992).

At time points, where there is contact, the variational principle is augmented with contact constraints that depend on the precise physical setting. These constraints ensure that states which are assumed impossible, such as those with interpenetration or deformation, cannot occur. At these time points, the d'Alembert–Lagrange

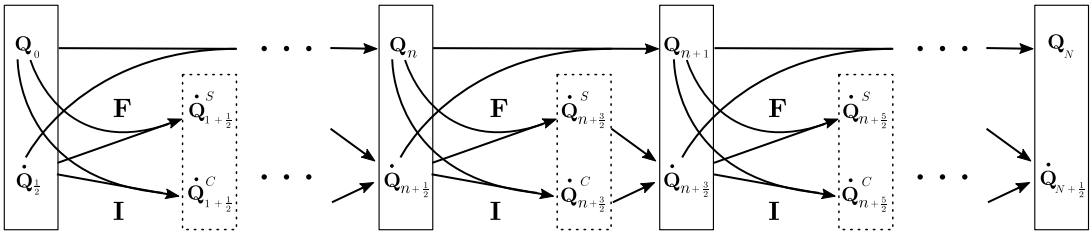


Figure 2: A CD-Lagrange network. Here, we begin from initial states $(\mathbf{Q}_0, \dot{\mathbf{Q}}_{\frac{1}{2}})$. We calculate the next position $\mathbf{Q}_1$, and proceed to calculate the next velocity $\dot{\mathbf{Q}}_{1+\frac{1}{2}} = \dot{\mathbf{Q}}^S_{1+\frac{1}{2}} + \dot{\mathbf{Q}}^C_{1+\frac{1}{2}}$ as a sum of smooth and contact terms. These terms are in turn calculated using the conservative forces $\mathbf{F}$ and the impulse $\mathbf{I}$, which are calculated from the parameterised Lagrangian, whose potential energy is given by a fully connected network.

principle does not apply, and the Euler–Lagrange equations (2) do not hold. To handle this, one splits the action integral in equation (1) into contact-free intervals with non-smooth contact-driven transitions in between, which are analysed separately.

Applying the variational principle at the contact time points yields *transfer of momentum* equations, usually resulting from Newton's restitution law and the law of conservation of momentum (Fetecau et al., 2003; Halliday et al., 2013). These equations relate the state directly before contact to the state directly after contact. To calculate this, one isolates the components of momentum, which are normal to the contact surface, and transfers them appropriately.

Implementing contact dynamics numerically yields a number of issues, which depend on the regime employed. For example, certain regimes involve calculating the precise time when contacts occur, which might happen between integration time steps. In such cases, transfer of momentum is applied at the contact time point. This yields good numerical accuracy, but often requires the solution of an optimisation problem to determine the precise contact time (Fetecau et al., 2003). Other regimes might instead relax the dynamics to allow object interpenetration in order to avoid expensive fixed-point iterations (Fekak et al., 2017). This entails carefully considering how to handle interpenetration to ensure accuracy. Figure 1 illustrates a sample numerical trajectory of a bouncing ball, including numerical transfer of momentum at a contact time point.

3 Physically Structured Networks for Contact Dynamics

To define neural network models for contact dynamics, we begin with the perspective of neural ODEs (E, 2017; Haber and Ruthotto, 2017; Chen et al., 2018), which will be helpful for this purpose. Here, we specify a system of ODEs driven by a single-layer neural network, and discretise it to obtain a deep or recurrent neural network architecture. In the classical case, an Euler discretisation yields a deep residual network, where the depth is given by the number of discretisation steps.

Building on these ideas, a number of recent works have proposed replacing the black-box system of ODEs with other systems that are more structured (Raissi, 2018; Greydanus et al., 2019; Lutter et al., 2020; Cranmer et al., 2020; Sæmundsson et al., 2020). By applying structure-preserving discretisation schemes to these ODEs, one obtains neural network architectures with built-in inductive biases that improve generalisation and allow the networks to learn with less data.

In physical settings, a number of recent works have paired structured equations from mechanics, such as the Euler–Lagrange equations or Hamilton's equations, with structure-preserving integrators, such as the Störmer–Verlet method, giving rise to *physically structured neural networks* (Greydanus et al., 2019; Lutter et al., 2020; Cranmer et al., 2020; Sæmundsson et al., 2020). These networks use the mathematical structure of classical mechanics as an inductive bias, ensuring the network mirrors important qualitative physical properties, such as conservation of momentum or conservation of energy. These inductive biases have been shown to improve data efficiency and facilitate accurate long-term forecasting (Raissi, 2018; Greydanus et al., 2019; Lutter et al., 2020; Cranmer et al., 2020; Sæmundsson et al., 2020).

3.1 Central-Difference Lagrange Networks

We now employ these techniques to design neural network architectures specifically suited for modelling contact dynamics. There are three main properties present in the true physics that we aim to encode into the neural network architecture.

(a) The network should be expressive enough to model a general Newtonian rigid body system.

(b) The network should be constrained to exclude non-physical dynamics that black-box networks allow, so that it learns in a data-efficient manner.

(c) The network should be able to handle periods without contact separately from contact events, as said phenomena differ in character.

Following Sæmundsson et al. (2020), one can construct a network satisfying the first two properties by applying a *symplectic integrator* to the Euler–Lagrange equations induced by a free-form Lagrangian parameterised by a fully connected network, which are interpreted as a structured neural ODE. We extend these constructions to explicitly handle contact events. We begin by defining a parameterised Lagrangian L_θ for a free-form Newtonian potential system, given by

$$L_\theta(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \mathbf{M} \dot{\boldsymbol{q}} - V_\theta(\boldsymbol{q}), \tag{3}$$

where $\mathbf{M}$ is a symmetric positive semi-definite position-independent inertia matrix, and V_θ is modelled by a fully connected neural network.

To obtain a network architecture for contact dynamics, we apply the symplectic *Central-Difference Lagrange integrator* of Fekak et al. (2017), modified for rigid body impacts by Di Stasio et al. (2019). We work with a simplified variant designed for modelling rigid non-deformable bodies—see Appendix A, as well as Fekak et al. (2017) and Di Stasio et al. (2019) for the general case. This construction yields a recurrent network architecture specialised for modelling contact dynamics, which we now showcase. A schematic overview can be found in Figure 2.

CD-Lagrange uses separate time grids for the position $\boldsymbol{q}_n$ and velocity $\dot{\boldsymbol{q}}_{n+\frac{1}{2}}$, at times t_n and $t_{n+\frac{1}{2}}$. We combine the position of all bodies in the matrix

$$\mathbf{Q}_n = \left(\boldsymbol{q}_n^1, \ldots, \boldsymbol{q}_n^K\right), \tag{4}$$

where K is the total number of bodies in the system. Given a position-velocity pair $(\mathbf{Q}_n, \dot{\mathbf{Q}}_{n+\frac{1}{2}})$, CD-Lagrange calculates the next position using the midpoint velocity, given by

$$\mathbf{Q}_{n+1} = \mathbf{Q}_n + \frac{h}{2}\dot{\mathbf{Q}}_{n+\frac{1}{2}}, \tag{5}$$

where h is the size of the time step. At the next time step, the change in velocity is calculated as a sum of changes due to smooth dynamics and due to contact, yielding

$$\dot{\mathbf{Q}}_{n+\frac{3}{2}} = \dot{\mathbf{Q}}^S_{n+\frac{3}{2}} + \dot{\mathbf{Q}}^C_{n+\frac{3}{2}}, \tag{6}$$

$$\dot{\mathbf{Q}}^S_{n+\frac{3}{2}} = \dot{\mathbf{Q}}_{n+\frac{1}{2}} + h\mathbf{M}^{-1}\mathbf{F}(\mathbf{Q}_{n+1}, \dot{\mathbf{Q}}_{n+\frac{1}{2}}), \tag{7}$$

$$\dot{\mathbf{Q}}^C_{n+\frac{3}{2}} = \mathbf{M}^{-1}\mathbf{I}(\mathbf{Q}_{n+1}, \dot{\mathbf{Q}}_{n+\frac{1}{2}}), \tag{8}$$

where $\mathbf{F}$ is the conservative force and $\mathbf{I}$ the impulse that occurs during contact, both defined below. The conservative forces are calculated from the parameterised Lagrangian as

$$\mathbf{F}(\mathbf{Q}_{n+1}, \dot{\mathbf{Q}}_{n+\frac{1}{2}}) = -\frac{\partial L_\theta(\mathbf{Q}_{n+1}, \dot{\mathbf{Q}}_{n+\frac{1}{2}})}{\partial \mathbf{Q}_{n+1}} \tag{9}$$

$$= -\frac{\partial V_\theta(\mathbf{Q}_{n+1})}{\partial \mathbf{Q}_{n+1}}, \tag{10}$$

which, in our setting, are the conservative forces arising from the potential $V_\theta(\mathbf{Q}_{n+1})$. Rigid-body impacts are handled by Newton's restitution law (Fekak et al., 2017; Di Stasio et al., 2019)

$$\mathbf{U}_{n+\frac{3}{2}} = -e\mathbf{U}_{n+\frac{1}{2}}, \tag{11}$$

where $\mathbf{U}_{n+\frac{1}{2}} = \dot{\mathbf{Q}}_{n+\frac{1}{2}}\mathbf{L}^T_{n+1}$ with $\mathbf{L}$ defined below, and $e \in [0, 1]$ is the elasticity parameter with $e = 1$ defining elastic impacts with no dissipation. Furthermore, for rigid body-body impacts, the law of conservation of momentum

$$\sum_{k=1}^{K} m^k \dot{\boldsymbol{q}}^k_{n+\frac{3}{2}} = \sum_{k=1}^{K} m^k \dot{\boldsymbol{q}}^k_{n+\frac{1}{2}} \tag{12}$$

needs to be considered in order to uniquely resolve collision events. Define the projection operator

$$\mathbf{L} = \begin{bmatrix} \boldsymbol{n}_1 & \ldots & \boldsymbol{n}_k & \ldots & \boldsymbol{n}_K \end{bmatrix}, \tag{13}$$

which contains the normal vectors of the surface each body it is in contact with. For each body k, the corresponding impulse $\boldsymbol{I}^k$ is given by

$$\boldsymbol{I}^k_{n+1} = \begin{cases} \boldsymbol{L}^k_{n+1}\lambda^k_{n+\frac{3}{2}} & \text{if } c^k_{n+1} = 1 \\ \mathbf{0} & \text{otherwise} \end{cases}, \tag{14}$$

where c^k_{n+1} is a discrete contact signal for body k and

$$\lambda^k_{n+\frac{3}{2}} = \left[\mathbf{H}_{n+1}\left(e\dot{\mathbf{Q}}_{n+\frac{1}{2}} + \dot{\mathbf{Q}}^S_{n+\frac{3}{2}}\right)\mathbf{L}^T_{n+1}\right]_{kk} \tag{15}$$

represents the impulse acing on body k. The operator $\mathbf{H}$ is defined as

$$[\mathbf{H}_{n+1}]_i = \left[\mathbf{A}_{n+1}\mathbf{M}^{-1}\mathbf{A}^T_{n+1}\right]^{-1}_i, \tag{16}$$

and ensures that the mass ratios between the bodies in contact, resulting from the law of conservation of momentum, are applied to the correct bodies. Here, the operator $\mathbf{A}$ selects the components of the bodies that are in contact with each other, and is given by

$$[\mathbf{A}_n]_{ij} = \begin{cases} -1 & \text{if } i = j \text{ and } c^i_n = 1 \\ 1 & \text{if body } i \text{ and } j \text{ are in contact} \\ & \text{(implicitly } c^i_n = c^j_n = 1) \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

In total, these expressions define the general *CD-Lagrange network*. Further details, including derivation of these expressions, are provided in Appendix A.

Idealised touch feedback. When using a CD-Lagrange network to predict future system states, one needs to determine whether or not objects are in contact at each time step in order to calculate whether or not the impulse component of the network comes into play. To do so, we introduce an additional *contact network* $\hat{\boldsymbol{c}}_\theta$ that learns to predict a discrete contact signal $\boldsymbol{c} \in \{0,1\}^K$ defined by

$$c_n^k = \begin{cases} 1 & \text{if contact for body } k \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

at time step n. We consider two regimes: one in which $\boldsymbol{c}_n$ is unobserved, and another where it is fully observed at training time. The latter case can be conceptually thought of as the addition of an *idealised touch feedback* sensor to the system, which determines whether or not contact is present. We explore this difference and its effect on performance in the sequel.

3.2 Learning from noisy observations

Given a dataset of J observed trajectories $(\boldsymbol{y}_j, \boldsymbol{c}_j)$, $j = 1, \dots, J$, of length N_j each with step size h and $\boldsymbol{y}_j = (\mathbf{Q}_j, \dot{\mathbf{Q}}_j)$. We train the network $\hat{y}_{\theta,h}$ by minimising the mean squared error loss between the (noisy) state observations and predicted states with respect to the parameters of the potential network V_θ. We also jointly train the contact network $\hat{c}_\theta$ by minimising cross-entropy loss between its output and the contact signal c. For a single trajectory, this is given by

$$\mathcal{L}_T(\boldsymbol{y}_j, \theta) = \frac{1}{N_j D} \sum_{n=1}^{N_j} \|\boldsymbol{y}_n - \hat{y}_{\theta,h}(\boldsymbol{y}_0, \boldsymbol{c}_0)\|^2. \tag{19}$$

The corresponding contact network loss is given by

$$\mathcal{L}_C(\boldsymbol{c}_j, \boldsymbol{y}_j, \theta) = -\frac{1}{N_j K} \sum_{n=1}^{N_j} \sum_{k=1}^{K} \Big(c_n^k \log \hat{c}_\theta^k(\boldsymbol{y}_n) + (1 - c_n^k) \log(1 - \hat{c}_\theta^k(\boldsymbol{y}_n)) \Big), \tag{20}$$

where the activation of the output of the contact network $\hat{\boldsymbol{c}}_\theta$ is the sigmoid (logistic) function, and $\hat{c}_\theta^k(\boldsymbol{y}_n)$ is the contact network's prediction for body k at time step n. We further add a regularisation term $\mathcal{L}_R(\theta)$ on the parameters of the model, which is described in the sequel. In the full dataset, the observed trajectories might contain many steps or have unequal lengths N_j. To avoid vanishing gradients, instead of optimising with respect to the full trajectories, we instead split the data into batches of horizon H. The optimal parameters are found by minimising

$$\hat{\theta} = \min_\theta \mathcal{L}_T + \mathcal{L}_C + \mathcal{L}_R \tag{21}$$

using mini-batch stochastic gradient descent.

ℓ^2 regularisation. Since the change in velocity in CD-Lagrange $\dot{\mathbf{Q}}_{n+\frac{3}{2}} = \dot{\mathbf{Q}}^S_{n+\frac{3}{2}} + \dot{\mathbf{Q}}^C_{n+\frac{3}{2}}$ is *additive* over conservative and contact components, as part of training, a CD-Lagrange network needs to learn to distinguish which component should be used to predict a given trajectory in the training data. ℓ^2 regularization affects this aspect of the learning problem in an subtle but outsized manner: as a consequence of shrinking the potential network's weights to zero, the regularizer shrinks the *function* $\boldsymbol{F}(\mathbf{Q}_{n+1}) : \mathbb{R}^d \to \mathbb{R}^d$ to the *zero function* $\mathbb{0} : \mathbf{Q} \mapsto \mathbf{0}$. This encourages the network to explain contact events with contact forces rather than conservative forces where possible.

4 Experiments

In order to investigate the properties of the proposed CD-Lagrange networks, we run experiments on a number of reference rigid body systems. In Section 4.1, we study performance on an ideal pendulum system without contacts, focusing on the physical properties of the conserved dynamics defined in equations (7) and (9), thereby checking the network's performance relative to baselines. In Section 4.2, we perform experiments on a rigid bouncing ball system, exploring the behaviour of the network when contact happens and the effect of including an idealised touch sensor on the ability of recovering the underlying dynamics. Finally, in Section 4.3, we study body-body impacts in an idealised Newton's cradle and look at the network's ability to recover the underlying dynamics and effects of numerical interpenetration during contact events.

We compare CD-Lagrange networks with residual networks (ResNets) and variational integrator networks (VINs) by evaluating them in terms of predictive performance on held-out test data as well as qualitatively evaluating the corresponding phase diagrams. For contact experiments, we additionally include a modified residual network baseline (ResNetContact), which takes as input both the state as well the contact signal, for comparison against the CD-Lagrange networks' idealised touch feedback sensor. To generate data, we simulate trajectories of motion and add independent Gaussian noise to the positions and velocities. Full details for experiment hyperparameters and training are given in Appendix B.

4.1 Learning to predict motion without contacts: an ideal pendulum

We first examine whether the performance of CD-Lagrange networks matches previous work on physically structured networks in cases where there is no contact. To this end, we consider learning to predict the trajectory of a simple *ideal pendulum* from observed

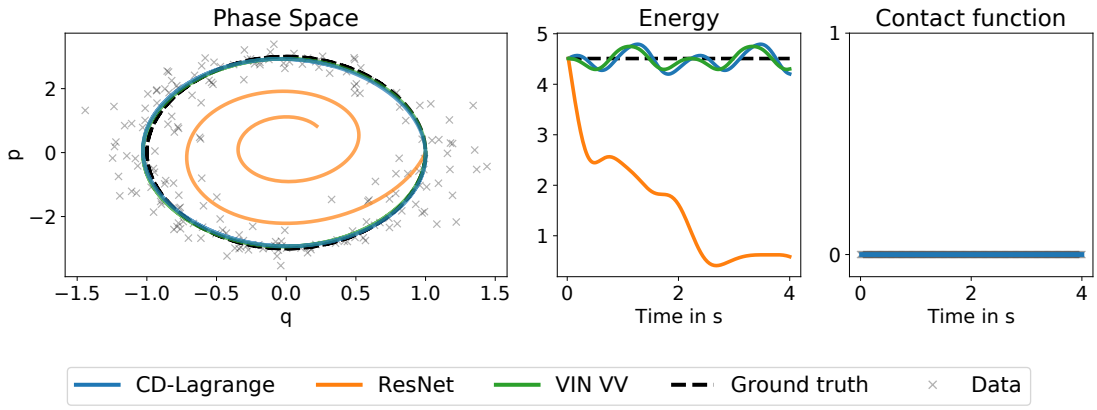


Figure 3: Learning the equations of motion of an ideal pendulum, which has no contacts. Given a set of initial conditions, we forecast a path in phase space and predict against ground truth. We display ground truth, training data, and model predictions, and energy over time for each of the models. For the CD-Lagrange network, we display the learned contact function, which is approximately zero everywhere. Root mean squared errors for each network are given as follows: CD-Lagrange RMSE: 0.538, ResNet RMSE: 1.156, VIN VV RMSE: 0.509.

data. The ideal pendulum is a point-mass attached to a mass-less rigid rod, suspended from a pivot. The pendulum swings back and forth due to gravity, without friction, so that the system conserves energy. This task has been studied previously by a number of authors, such as Greydanus et al. (2019) and Sæmundsson et al. (2020), who propose to use a network constructed from a variational *velocity Verlet* integrator in order to learn in a data-efficient manner. The variational integrator network preserves more physical properties than the CD-Lagrange network, and therefore we expect it to perform better in settings where there are no contacts. We generate 20 training trajectories consisting of 10 points each by simulating trajectories and adding independent Gaussian noise to all position-velocity pairs.

Figure 3 plots the phase-space trajectories for the ground truth system, the observed training data and the predicted evolution for each model. The CD-Lagrange network and variational integrator network achieve comparable error, which significantly improves upon the baseline residual network, which incorrectly dissipates energy. The network also recovers the correct contact function, which is the zero function. This shows that relative to prior works on physically structured neural networks, accurate performance in contact-free scenarios is not compromised by the extra structure added to handle contacts.

4.2 Learning body-wall contacts: an ideal bouncing ball

Next, we study using CD-Lagrange networks to learn contact dynamics for an ideal rigid *bouncing ball*. This system, shown in Figure 4, is a commonly studied example in the non-smooth contact dynamics literature (Di Stasio et al., 2019). The ball accelerates towards the ground due to the force of gravity, hits the ground, and bounces back up. The effect of the impact is determined by Newton's restitution law, given in equation (11), where the elasticity parameter e controls the energy behaviour. We focus on the regime $e = 1$, where energy is conserved.

We consider two different data regimes. In the first regime, only observed trajectories of motion are provided at training time. In the second regime, we add

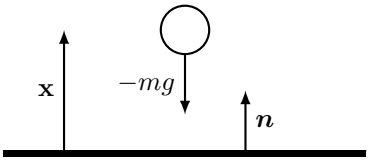


Figure 4: Schematics of the bouncing ball. Here, the ball falls due to the gravitational pull $-mg$, and experiences a contact force in the direction of the contact normal vector $\boldsymbol{n}$ with respect to the floor.

	RMSE
ResNet	6.6 ± 1.2
ResNetContact	4.8 ± 0.8
CD-Lagrange	1.9 ± 1.0

Table 1: Average root-mean squared error and standard error of the bouncing ball experiment averaged over 5 runs with 40 trajectories each consisting of 10 data points.

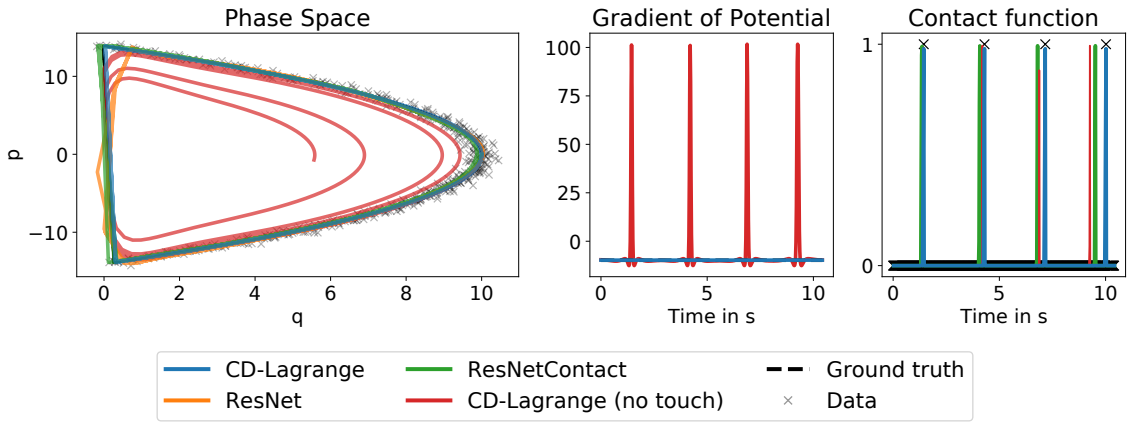


Figure 5: Learning the equations of motion of the bouncing ball. Given a set of initial conditions, we forecast a path in phase space and predict against ground truth. We display ground truth, training data, and model predictions for each of the models. For the CD-Lagrange network, we display the gradient of the potential energy, and the learned contact function. Root mean squared errors for each network's predictions are as follows: CD-Lagrange: 0.067, CD-Lagrange (no touch): 7.578, ResNet: 6.778, ResNetContact: 8.866.

idealised touch sensor data that indicates at each time point whether or not contact has occurred, given in equation (18). This can be viewed as representing an ideal impact sensor located inside the ball, or along the floor. To learn this system's equations of motion from data, we generate 52 training trajectories of 10 points each, and train a CD-Lagrange network, a residual network, and a modified residual network that also takes contact data as input. Results can be seen in Figure 5. Here, we see that the CD-Lagrange network's performance depends critically on whether or not it has access to touch feedback data. With touch feedback, the CD-Lagrange network predicts the ball's trajectory much more accurately than the residual network. Without touch feedback, the CD-Lagrange network fails to learn the correct dynamics—instead, the network attempts to incorrectly explain noise using contact events, and contact events using smooth dynamics, because all scenarios lead to similar-looking noisy data from the network's perspective. The residual network struggles to approximate the non-smooth behaviour at impact time, replacing instantaneous contacts with fast movement. Adding contact information to the residual network's inputs does not improve its performance.

From examining the potential and contact network in Figure 5, we see that the CD-Lagrange network with touch feedback determines the impact times near-exactly. The gradient of the potential energy remains very close to the ground truth value, even as the system evolves and contact events occur. This shows that the network correctly determines that contact-driven changes in system states are caused by contacts, and not by spurious potential energy within the smooth dynamics—so long as the network is provided with touch feedback that enables it to differentiate between contact events and noise. Root mean squared error, together with standard error, can be seen in Table 1.

4.3 Learning body-body impacts: Newton's cradle

Finally, we consider learning body-body impacts using CD-Lagrange networks in a simple *Newton's cradle* system consisting of two balls suspended by a string, shown in Figure 6. We assume both bodies have no volume, are suspended from a common mounting point, and parameterise their locations using angles relative to the vertical axis. This means that collisions will occur perpendicular to the contact surface. We train on 54 trajectories consisting of 10 data points each, and again consider CD-Lagrange networks with and without idealised touch sensor data, along with residual network baseline within both data regimes.

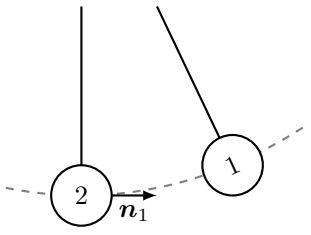


Figure 6: Schematics of the Newton's cradle system with two balls. Here, the first ball swings along the suspended rope due to gravity, and experiences a contact force in the direction of the contact normal vector $\boldsymbol{n}_1$ upon impact.

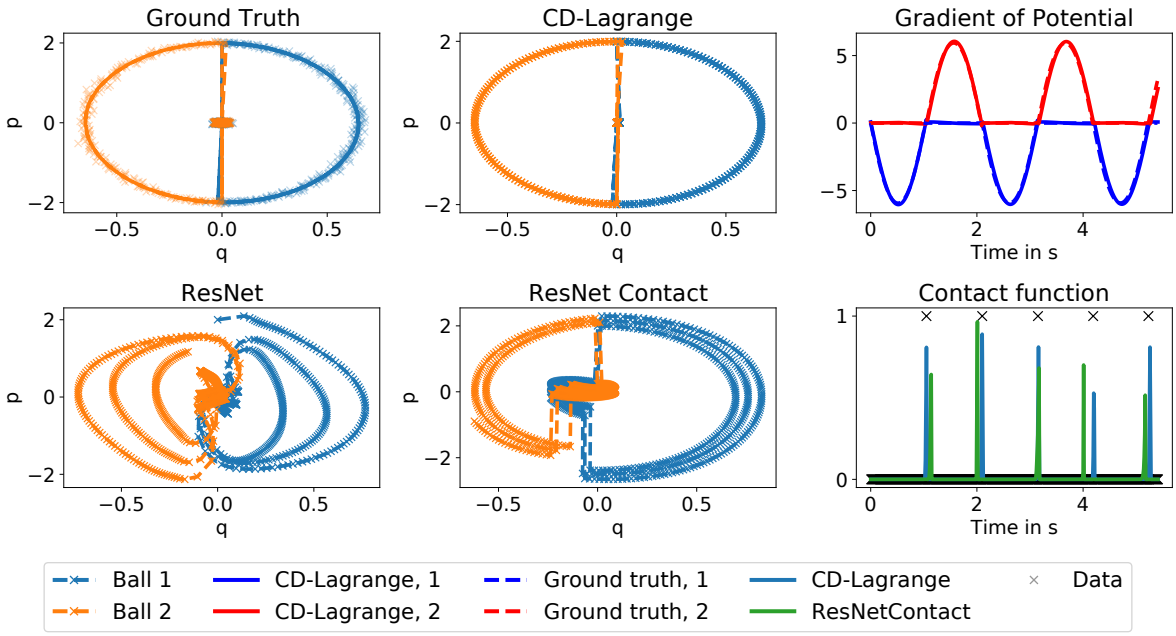


Figure 7: Learning the equations of motion of the Newton's cradle. Given a set of initial conditions, we forecast a path in phase space and predict against ground truth. We display ground truth, training data, and model predictions for each of the models. For the CD-Lagrange network, we display the gradient of the potential energy, and the learned contact function. Root mean squared errors for each network are given as follows: CDL RMSE: 0.406, ResNet RMSE: 1.685, ResNetContact RMSE: 1.034.

Results can be seen in Figure 7. Here, we see that the phase-space trajectory of the CD-Lagrange network is substantially more accurate than for a baseline residual network. Root mean squared error, together with standard error, can be seen in Table 2. As before, adding contact information does not improve the residual network's performance. This replicates the results of the single-body bouncing ball experiment in Section 4.2 in a multi-body setting.

Next, we examine how accurately the potential energy and contact function are recovered—this can also be seen in Figure 7. The CD-Lagrange network learns the potential accurately, with some error around the contact points. In particular, the contact network accurately learns to determine when contact will occur based on the touch sensor data. Mirroring the results of Section 4.2, the residual network struggles to predict contact events even when it is given explicit touch sensor data on when they occur, particularly through mistiming contact events to occur earlier or later than is correct.

	RMSE
ResNet	1.6 ± 0.1
ResNetContact	3.5 ± 1.3
CD-Lagrange	0.4 ± 0.1

Table 2: Average root-mean squared error with standard error of the Newton's cradle experiment averaged over 5 runs with 50 trajectories each consisting of 10 data points.

The CD-Lagrange integrator, and by proxy the CD-Lagrange network, allow some slight interpenetration to occur. In body-wall impact scenarios, this does not strongly affect performance. However, in body-body impacts, this qualitative behaviour difference becomes more significant. Specifically, as momentum gets transferred between bodies, the slight allowed interpenetration can cause the system to significantly violate conservation of energy. In the Newton's cradle example, this causes the resting ball to gain potential energy during collision events, eventually causing both balls to accelerate. To mitigate this problem, we introduce a closest-point projection during impact events (Fetecau et al., 2003), which aligns the resting ball with the closest point on the boundary where no penetration occurs. This improves accuracy and restores correct long-term system behaviour.

5 Discussion

CD-Lagrange networks are a flexible way for data-driven learning of equations of motion that include

contact dynamics. These networks can learn to accurately predict trajectories and resolve collisions in a noise-robust and data-efficient manner. This contributes to a rapidly-expanding line of work on neural ODEs and physically structured learning, and shows that these ideas can work successfully in non-smooth settings.

Compared to black-box neural networks, the implementation of physically structured networks for learning contact dynamics presents a number of additional challenges to ensure their performance. In particular, we find that the presence or absence of touch feedback impacts the model's performance dramatically. This is because without touch feedback the learning problem is ambiguous: abrupt shifts in the system state can be explained either by contacts, or by noise, and it is difficult for the network to tell which is which. This results in weak identifiability in the loss function, which could lead to difficulties with local optima or other issues. It is therefore important that the network is provided with an appropriate means to differentiate between the two, e.g. by means of a touch sensor.

The networks studied here are based on the CD-Lagrange integrator, which is an explicit scheme that fits conveniently within an automatic differentiation framework. This convenience comes at a cost: the integrator allows for some physically incorrect interpenetration, which can affect long-term prediction accuracy. Other schemes, such as variational integrators for contact dynamics (Fetecau et al., 2003), avoid these issues and achieve higher accuracy, but they require the solution of fixed-point iterations or convex optimisation problems during time-stepping, which renders them more expensive and cumbersome to work with. These issues can potentially be mitigated through the use of differential physics engines as components of deep-learning-based models (de Avila Belbute-Peres et al., 2018). Studying these trade-offs in the context of learning could pave the way toward better understanding on how to incorporate inductive biases to improve performance of neural networks to model the real world, thereby facilitating their use in applications such as robotics.

6 Conclusion

In this work, we introduce CD-Lagrange networks, which build on ideas from neural ODEs and physically structured learning to construct networks for learning contact dynamics in data-limited regimes. With the addition of an idealised touch feedback sensor, these networks can learn to accurately reconstruct non-smooth contact dynamics from data, in a way that disentangles contact-driven forces from conservative forces. The simple and explicit structure makes these networks interpretable, well-matched with the underlying physics, and straightforward to implement. A rapidly growing line of work has focused on adapting deep networks to various physical settings. We hope our contributions facilitate the inclusion of non-smooth contact dynamics within these settings.

References

V. M. M. Alvarez, R. Roşca, and C. G. Fălcuţescu. DyNODE: Neural ordinary differential equations for dynamics modeling in continuous control. *arXiv:2009.04278*, 2020. Cited on page 1.

R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018. Cited on pages 2, 3.

F. Cirak and M. West. Decomposition contact response (DCR) for explicit finite element dynamics. *International Journal for Numerical Methods in Engineering*, 64(8):1078–1110, 2005. Cited on pages 2, 11, 12.

M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho. Lagrangian neural networks. In *International Conference on Learning Representations Workshop on Deep Differential Equations*, 2020. Cited on pages 1–3.

F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter. End-to-end differentiable physics for learning and control. *Advances in Neural Information Processing Systems*, 2018. Cited on page 9.

M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015. Cited on page 2.

J. Di Stasio, D. Dureisseix, A. Gravouil, G. Georges, and T. Homolle. Benchmark cases for robust explicit time integrators in non-smooth transient dynamics. *Advanced Modeling and Simulation in Engineering Sciences*, 6(1):1–31, 2019. Cited on pages 4, 6, 11.

W. E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 2017. Cited on pages 2, 3.

F.-E. Fekak, M. Brun, A. Gravouil, and B. Depale. A new heterogeneous asynchronous explicit–implicit time integrator for nonsmooth dynamics. *Computational Mechanics*, 60(1):1–21, 2017. Cited on pages 3, 4, 11, 12.

R. C. Fetecau, J. E. Marsden, M. Ortiz, and M. West. Nonsmooth Lagrangian mechanics and variational collision integrators. *SIAM Journal on Applied Dynamical Systems*, 2(3):381–416, 2003. Cited on pages 3, 8, 9.

S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian Neural Networks. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 1–3, 6.

E. Haber and L. Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017. Cited on pages 2, 3.

D. Halliday, R. Resnick, and J. Walker. *Fundamentals of Physics*. Wiley, 2013. Cited on page 3.

M. Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3–4):235–257, 1999. Cited on page 2.

J. N. Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017. Cited on page 1.

S. Leyendecker, J. E. Marsden, and M. Ortiz. Variational integrators for constrained dynamical systems. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, 88(9):677–708, 2008. Cited on page 2.

M. Lutter, C. Ritter, and J. Peters. Deep Lagrangian networks: using physics as model prior for deep learning. In *International Conference on Learning Representations*, 2020. Cited on pages 1–3.

J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10(1):357–514, 2001. Cited on page 2.

J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In *Nonsmooth mechanics and Applications*, pages 1–82. Springer, 1988. Cited on page 12.

V. Ortenzi, H.-C. Lin, M. Azad, R. Stolkin, J. A. Kuo, and M. Mistry. Kinematics-based estimation of contact constraints using only proprioception. In *International Conference on Humanoid Robots*, 2016. Cited on page 2.

M. Raissi. Deep hidden physics models: deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018. Cited on pages 1–3.

N. Rotella, S. Schaal, and L. Righetti. Unsupervised contact learning for humanoid estimation and control. In *International Conference on Robotics and Automation*, 2018. Cited on page 2.

L. Ruthotto and E. Haber. Deep neural networks motivated by partial differential equations. *arXiv:1804.04272*, 2018. Cited on page 2.

S. Sæmundsson, A. Terenin, K. Hofmann, and M. P. Deisenroth. Variational integrator networks for physically structured embeddings. In *International Conference on Artificial Intelligence and Statistics*, 2020. Cited on pages 1–4, 6, 14.

J. M. Sanz-Serna. Symplectic integrators for Hamiltonian problems: an overview. *Acta Numerica*, 1(243-286):123–124, 1992. Cited on page 2.

J. Viereck, J. Kozolinsky, A. Herzog, and L. Righetti. Learning a structured neural network policy for a hopping task. *IEEE Robotics and Automation Letters*, 3(4):4092–4099, 2018. Cited on page 1.

P. Wriggers and P. D. Panagiotopoulos. *New developments in contact problems*, volume 384. Springer, 1999. Cited on page 11.