

Learning mechanical systems from real-world data using discrete forced Lagrangian dynamics

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 We introduce a data-driven method for learning the equations of motion of me-
 2 chanical systems directly from position measurements, without requiring access
 3 to velocity data. This is particularly relevant in system identification tasks where
 4 only positional information is available, such as motion capture, pixel data or
 5 low-resolution tracking. Our approach takes advantage of the discrete Lagrange-
 6 d'Alembert principle and the forced discrete Euler-Lagrange equations to construct
 7 a physically grounded model of the system's dynamics. We decompose the dy-
 8 namics into conservative and non-conservative components, which are learned
 9 separately using feed-forward neural networks. In the absence of external forces,
 10 our method reduces to a variational discretization of the action principle naturally
 11 preserving the symplectic structure of the underlying Hamiltonian system. We
 12 validate our approach on a variety of synthetic and real-world datasets, demonstrat-
 13 ing its effectiveness compared to baseline methods. In particular, we apply our
 14 model to (1) measured human motion data and (2) latent embeddings obtained via
 15 an autoencoder trained on image sequences. We demonstrate that we can faith-
 16 fully reconstruct and separate both the conservative and forced dynamics, yielding
 17 interpretable and physically consistent predictions.

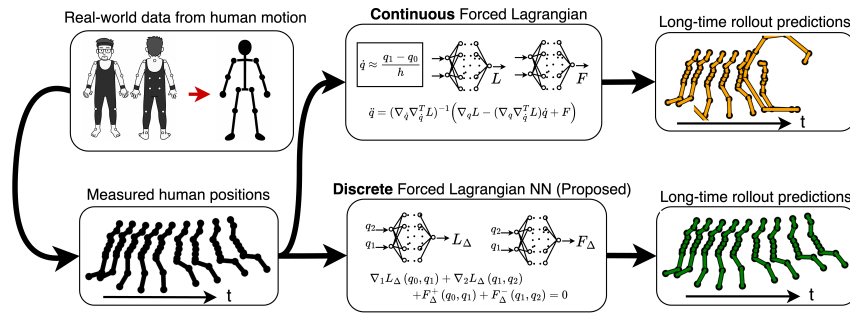


Figure 1: In this paper, we propose a structure-preserving approach for learning non-conservative system that directly learns from position data only (i.e., does not require velocities or momenta).

1 Introduction

19 Incorporating physics principles into machine learning has become a popular strategy for system
 20 identification and accurately predicting dynamics. Seminal examples are the Hamiltonian and
 21 Lagrangian neural networks, which are designed to preserve structure when applied to canonical

systems [1, 2]. These architectures have naturally been extended to include dissipation, enabling accurate modeling of realistic mechanical systems with physically motivated inductive biases [3, 4]. Such models offer advantages over traditional “black-box” modeling strategies that face challenges such as noise sensitivity, data sparsity, poor generalization, and limited interpretability [5, 6, 7, 8, 9]. Incorporating physics into the training processes is becoming an increasingly popular way to increase model generalization while giving physically interpretable predictions.

However, almost all known methods for learning dynamics rely on being able to observe momentum or velocity data, in addition to position data. In practice, velocity and momentum estimates are usually approximated from sequential position measurements using finite differences, which leads to inaccuracies due to noise and truncation errors [10]. Two examples considered in this paper are: (1) learning from pixel data or (2) motion tracking data, where instantaneous velocities are unavailable.

Building on ideas of geometric mechanics and variational integrators, [11], we propose a structure-preserving approach that directly learns from *position data only* (i.e., does not require velocities or momenta) and naturally incorporates dissipation and other non-conservative forces such as control terms. The method, which we refer to as a Discrete Forced Lagrangian Neural Network (DFLNN), is based on the discrete Lagrange-d’Alembert principle, which naturally reduces to a symplectic (variational) integrator on the configuration manifold in absence of non-conservative forces.

In a number of synthetic problems, the DFLNN is demonstrated to yield significant advantages over a non-structure-preserving neural ODE baseline model and, notably, a structure-preserving continuous analogue of the proposed method based on the (continuous) forced Euler-Lagrange equations where velocities are approximated using finite differences.

Additionally, we emphasize the flexibility of combining the proposed model with an autoencoder trained on images of damped pendulum dynamics. We show that we can separate the dissipative forces in the latent space to accurately recover and simulate pixel sequences of a *conservative* pendulum, despite the model only learning from the dissipative pixel images. Finally, we apply the model to real-world 3D human motion skeleton tracking data. The model successfully captures the underlying physics and shows excellent prediction performance over long times. Human motion data is extensively employed in simulations, animations, and biomechanical research [12]. Examining these movements has promising applications across various domains, including healthcare [13, 14] and sports [15]. Ensuring that models generalize effectively to unseen scenarios is crucial, particularly in medical applications.

Table 1: Comparison to other models. Models marked with a $\approx$ denotes approximate preservation of structure

	HNN [1]	LNN [2]	LSI [16]	D-HNN [17, 18]	GLNN [3]	NODE [19]	DFLNN (Ours)
(a) Learns from position only			✓			✓	✓
(b) Structure-preserving	$\approx$	$\approx$	✓	$\approx$	$\approx$		✓
(c) Incorporates dissipation				✓	✓		✓

1.1 Related work

Understanding and modeling mechanical systems from data is vital in fields such as robotics, biomechanics, and structural health monitoring. While conventional system identification often assumes access to fully known physics, recent efforts focus on physics-informed learning, leveraging partial physical knowledge to improve generalization and interpretability [20, 21, 22, 23, 24, 16, 25, 26, 27, 28, 29, 30, 31]. Early approaches like Neural ODEs [19] and sparse identification of nonlinear dynamics [32] rely on black-box models or predefined basis functions. These have been paired with auto-encoders to identify low-dimensional latent ODE models from pixel data [33], but often neglect structural constraints such as conservation laws.

To address this, recent work embeds physical priors directly into the learning process. Hamiltonian and Lagrangian-based learning methods learn energy-based formulations that define the equations of motion, yielding models with physically consistent inductive biases [1, 34, 35, 36, 37, 38, 39, 2, 8, 40, 41, 42, 43, 44, 16]. When paired with symplectic or variational integrators, these networks can

66 preserve geometric properties such as the symplectic form or modified energy over long trajectories
 67 [26, 30, 27, 25, 29]. More recently, extensions to non-conservative systems have emerged, combining
 68 structure-preserving principles with dissipative or driven dynamics [37, 45, 3, 46, 47]. These
 69 approaches mark a shift toward hybrid models that respect both physical structure and the realities of
 70 real-world data.

71 1.2 Our contribution.

72 Our main contributions are now summarized, with a comparison to other models in Table I.

- 73 a) We propose a discrete forced Lagrangian neural network that allows us to learn forced,
 74 dissipative dynamics from position data only, it does not require instantaneous velocity or
 75 momentum observations.
- 76 b) In the absence of external forces, our method reduces to a variational discretization of the
 77 action principle naturally preserving the symplectic structure of the underlying Hamiltonian
 78 system.
- 79 c) As it is based on the Lagrange-d’Alembert principle the DFLNN yields physically inter-
 80 pretable predictions, allowing us to separate conservative from non-conservative dynamics.
- 81 d) We show the method generalizes well on both synthetic and real-world data, as demonstrated
 82 in the computed predictions rollouts reproducing human motion.
- 83 e) We show that by implementing the DFLNN on learned autoencoder embeddings from pixel
 84 data, we can separate dissipative dynamics from the conservative dynamics in *latent* space.

85 2 Background

86 **Continuous d’Alembert Principle and Forced Euler-Lagrange Equations** In the presence of
 87 non-conservative forces, it is useful to consider the generalized formulation of Hamilton’s principle,
 88 commonly referred to as the Lagrange-d’Alemberts principle. Let Q denote a configuration manifold
 89 with TQ its tangent bundle. The Lagrange-d’Alemberts principle seeks a curve $q : [t_0, t_N] \rightarrow \mathbb{R}^d$
 90 that satisfies:

$$\delta \left(\int_{t_0}^{t_N} L(q(t), \dot{q}(t)) dt \right) + \int_{t_0}^{t_N} F(q(t), \dot{q}(t), u(t)) \cdot \delta q(t) dt = 0, \quad (1)$$

91 where δ denotes variations that vanish at the endpoints $q(t_0) = q_0$ and $q(t_N) = q_N$, $L : TQ \rightarrow \mathbb{R}$ is
 92 the Lagrangian function and $F : TQ \rightarrow TQ^*$ is the Lagrangian force, which is a fiber preserving
 93 map $F : (q, \dot{q}) \mapsto (q, F(q, \dot{q}))$, [11, 421]. This principle is equivalent to the *forced Euler-Lagrange*
 94 *equations*,

$$0 = \mathcal{E}(L, F)(q, t) := \frac{\partial L}{\partial q}(q(t), \dot{q}(t)) - \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}}(q(t), \dot{q}(t)) \right) + F(q(t), \dot{q}(t)). \quad (2)$$

95 In absence of non-conservative forces, (1) expresses the extremization of the action functional,
 96 realizing Hamilton’s principle, and (2) are the Euler-Lagrange equations. In previous work [3] Xiao
 97 et al. formulated their learning problem starting from equations (2), and proposed the Generalised
 98 Lagrangian Neural Networks (GLNNs), see also [2]. In their framework, L and F are functions
 99 approximated by neural networks using position and velocity data $(q, \dot{q})$ observed at discrete times;
 100 see [3], and Appendix C for details on this method and our implementation.

101 Since we are addressing scenarios where only position data are observed, following [11, 48], we
 102 propose instead to discretize the Lagrange d’Alembert principle, and derive discrete equations directly
 103 from the discrete principle.

104 **Discrete d’Alembert Principle and Forced Euler-Lagrange Equations** A discretization of (1)
 105 leads to the Discrete Lagrange-d’Alembert principle [11, 49]. Consider a partition of $[t_0, t_N]$ in N
 106 subintervals of equal size, h . Consider a choice of numerical approximations of F and L on each

subinterval $[t_n, t_{n+1}]$, denoted by

$$\begin{aligned} L_\Delta(q_n, q_{n+1}) &\approx \int_{t_n}^{t_n+h} L(q(t), \dot{q}(t)) dt, \\ F_\Delta^+(q_n, q_{n+1}, h) &\approx \int_{t_n}^{t_n+h} F(q(t), \dot{q}(t)) \cdot \frac{\partial q(t)}{\partial q_{n+1}} dt, \\ F_\Delta^-(q_n, q_{n+1}, h) &\approx \int_{t_n}^{t_n+h} F(q(t), \dot{q}(t)) \cdot \frac{\partial q(t)}{\partial q_n} dt. \end{aligned} \quad (3)$$

Here, $L_\Delta : Q \times Q \rightarrow \mathbb{R}$ and $F_\Delta^\pm : Q \times Q \rightarrow T^*Q$ represent the discrete Lagrangian and the discrete force, respectively. The discrete Lagrange-d'Alembert principle (dLDA) seeks a discrete trajectory, $\{q_n\}_{n=1}^N$, that satisfies the following condition:

$$\delta \sum_{n=0}^{N-1} L_\Delta(q_n, q_{n+1}) + \sum_{n=0}^{N-1} [F_\Delta^-(q_n, q_{n+1}) \cdot \delta q_n + F_\Delta^+(q_n, q_{n+1}) \cdot \delta q_{n+1}] = 0, \quad (4)$$

for all variations $\{\delta q_n\}_{n=0}^N$ vanishing at the endpoints $\delta q_0 = \delta q_N = 0$. Equivalently, (4) can be expressed in terms of the discrete forced Euler-Lagrange equations:

$$\begin{aligned} 0 = \mathcal{E}_\Delta(L_\Delta, F_\Delta)(q_{n-1}, q_n, q_{n+1}) &:= \nabla_2 L_\Delta(q_{n-1}, q_n) + \nabla_1 L_\Delta(q_n, q_{n+1}) \\ &\quad + F_\Delta^+(q_{n-1}, q_n) + F_\Delta^-(q_n, q_{n+1}), \quad n = 2, \dots, N-1, \end{aligned} \quad (5)$$

where ∇_1 and ∇_2 denote differentiation with respect to the first and second variable.

3 Discrete Forced Lagrangian Neural Networks

The proposed model learns the dynamics of an observed system through neural network approximations of the Lagrangian and external forces,

$$L_\theta \approx L \quad L_\theta : TQ \rightarrow \mathbb{R}, \quad F_\theta \approx F, \quad F_\theta : TQ \rightarrow TQ^*,$$

where L_θ and F_θ are parameterized with learnable parameters θ .

The loss function minimized during training combines a physics term and a regularization term:

$$\mathcal{L} = \frac{\omega_{\text{physics}}}{N_{\mathcal{T}}(N+1)} \sum_{\mathcal{T}} \sum_{n=1}^{N-1} \mathcal{L}_{\text{physics}}(L_\theta, F_\theta)(q_{n-1}, q_n, q_{n+1}) + \frac{\omega_{\text{reg}}}{R} \sum_{r=1}^R \mathcal{L}_{\text{reg}}(L_\theta)(q_r, q_{r+1}). \quad (6)$$

Here, $\mathcal{L}_{\text{physics}}$ is deduced from the discrete forced Euler-Lagrange equations (5), while $\mathcal{L}_{\text{reg}}$ promotes the regularity of L_θ [379], each weighted with the hyperparameters ω_{physics} and ω_{reg} . The dataset $\mathcal{T}$ contains $N_{\mathcal{T}}$ trajectories of N steps each, and R denotes the number of point pairs (q_n, q_{n+1}) used for regularization. The regularization may be applied over all such pairs in the training dataset. However, in the experiments, we demonstrate that it is sufficient to regularize only over a moderate number of pairs of data points.

As (6) only evaluates $\mathcal{L}_{\text{physics}}$ on local triplets (q_{n-1}, q_n, q_{n+1}) , training is performed independently on such segments, without leveraging longer-range temporal dependencies present in the full trajectories. For generalizations of this approach using longer trajectory segments see Appendix A.

3.1 Learning Physics from the Observed Data ($\mathcal{L}_{\text{physics}}$)

Usually, the numerical discretization of a forced Lagrangian system would suggest for the exact L and F to be known, while the discrete trajectories $\{q_n\}_{n=0}^N$ are the unknowns of the problem. In the inverse setting considered here, the roles are reversed: the trajectories $\{q_n\}_{n=0}^N$ are given, and the goal is to recover (L_θ, F_θ) as approximations to the underlying Lagrangian and forces.

A discretization scheme can be applied to obtain the discrete Lagrangian and discrete force (L_Δ, F_Δ) as a function of (L_θ, F_θ) , (3). A concrete example using a mid-point approximation is

$$L_\Delta(q_n, q_{n+1}) := h L_\theta \left(\frac{q_n + q_{n+1}}{2}, \frac{q_{n+1} - q_n}{h} \right) = h L_\theta \left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}} \right) \quad (7)$$

$$F_\Delta^\pm(q_n, q_{n+1}, h) := \frac{h}{2} F_\theta \left(\frac{q_n + q_{n+1}}{2}, \frac{q_{n+1} - q_n}{h} \right) = \frac{h}{2} F_\theta \left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}} \right) \quad (8)$$

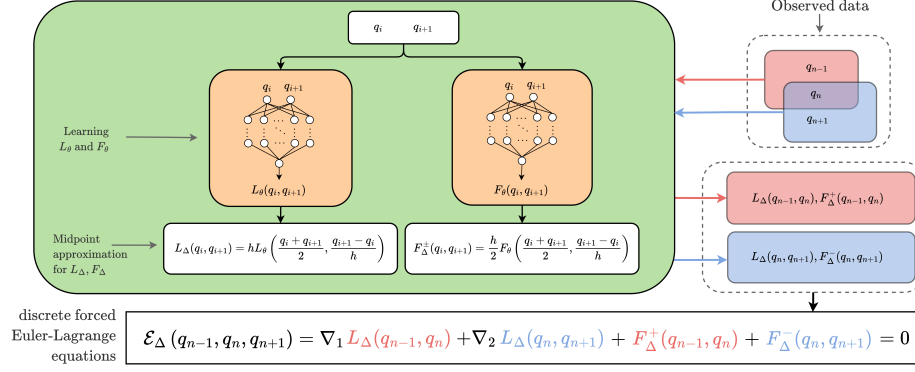


Figure 2: A schematic illustration of the data flow through the proposed model. We use segments of observed (q_{n-1}, q_n, q_{n+1}) to evaluate the discrete forced Euler-Lagrange equations.

where we have introduced the notation $\bar{q}_{n+\frac{1}{2}} := \frac{1}{2}(q_n + q_{n+1})$, $\bar{\dot{q}}_{n+\frac{1}{2}} := \frac{1}{h}(q_{n+1} - q_n)$ for brevity. We will use this discretization in our numerical experiments.

Assuming that the observed position data satisfy the dLDA principle (4) and, equivalently, the dLDA equations (5), a schematic illustration considering the flow of an observed trajectory of data points through the neural networks L_θ and F_θ —used to construct the dLDA equations in (5)—is provided in Figure 2. Summing over all observed trajectories, we define the following physics term for the loss function:

$$\begin{aligned} \mathcal{L}_{\text{physics}}(L_\theta, F_\theta)(q_{n-1}, q_n, q_{n+1}) = & \frac{h}{2} \left\| \nabla_1 L_\theta \left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}} \right) + \frac{2}{h} \nabla_2 L_\theta \left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}} \right) \right. \\ & + \nabla_1 L_\theta \left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}} \right) - \frac{2}{h} \nabla_2 L_\theta \left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}} \right) \\ & \left. + F_\theta \left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}} \right) + F_\theta \left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}} \right) \right\|_2. \end{aligned} \quad (9)$$

3.2 The Learned Lagrangian (L_θ)

The Lagrangian function L_θ can be learned either as a generic function or with embedded physical structure. In the most general case L_θ can be a feed-forward neural network. Alternatively, structural priors can be imposed. Assuming the Lagrangian takes a mechanical form, we can let

$$L_\theta(\bar{q}_k, \bar{\dot{q}}_k) = \bar{\dot{q}}_k^T M_\theta(\bar{q}_k) \bar{\dot{q}}_k - U_\theta(\cdot), \quad k = n \pm \frac{1}{2} \quad (10)$$

with $M_\theta(\bar{q}_k) = \epsilon \mathbb{I} + \Lambda_\theta^T(\bar{q}_k) \Lambda_\theta(\bar{q}_k)$ ensuring M_θ is symmetric and positive definite. Here, $\Lambda_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ is a lower triangular matrix parameterized by feed-forward neural network, and $\epsilon > 0$ a small constant ensuring the strict positiveness on M_θ . $U_\theta(\cdot)$ is the potential energy function, also modeled as a feed-forward neural network, and may depend on $\bar{q}_k$ and/or $\bar{\dot{q}}_k$.

Maximizing the regularity of the Lagrangian Learning L_θ by minimizing (9) can lead to trivial solutions, preventing the discovery of meaningful physical relationships (e.g., the network might learn L_θ as a constant, causing the derivative terms in (9) to vanish without properly fitting the data). This is a well-known challenge in data-driven Lagrangian-based models, and there are different strategies proposed to address this problem [44, 16, 50]. A Lagrangian is regular, or non-degenerate, if and only if its Hessian with respect to the second argument is invertible, [11, 379]. We will require the invertibility to hold point-wise: specifically on a selected number of pairs of points $(\bar{q}_{r+\frac{1}{2}}, \bar{\dot{q}}_{r+\frac{1}{2}})$, we require that

$$S(\bar{q}_{r+\frac{1}{2}}, \bar{\dot{q}}_{r+\frac{1}{2}}) := \left(\frac{\partial^2 L(\bar{q}_{r+\frac{1}{2}}, \bar{\dot{q}}_{r+\frac{1}{2}})}{\partial \bar{\dot{q}}_{r+\frac{1}{2}}^2} \right), \quad (11)$$

is invertible. Here $\bar{q}_{r+\frac{1}{2}}$ and $\bar{\dot{q}}_{r+\frac{1}{2}}$ are as defined in Section 3.1 and depend on the data (q_r, q_{r+1}) . We then include in the loss function a regularization term $\mathcal{L}_{\text{reg}}$ as a logarithmic barrier that maximizes the regularity of S by penalizing the absolute value of its determinant [44],

$$\mathcal{L}_{\text{reg}}(q_r, q_{r+1}) = |\log(|\det(S(\bar{q}_{r+\frac{1}{2}}, \bar{\dot{q}}_{r+\frac{1}{2}})|)|). \quad (12)$$

3.3 The Learned External Force (F_θ)

To learn the dynamics of systems with completely unknown forces, we parameterise $F_\theta = F_\theta^{\text{Free}}$ by a neural network. To prevent F_θ^{Free} from learning conservative dynamics described by the Lagrangian term, we apply dropout regularization [51]. This encourages slower learning of the force term and encourages the Lagrangian term to fit the data when possible.

When prior knowledge suggests a dissipative structure, we instead let F_θ be a Rayleigh dissipation function [52],

$$F_\theta(\bar{q}_k, \bar{\dot{q}}_k) = -K_\theta(\bar{q}_k) \bar{\dot{q}}_k, \quad k = n \pm \frac{1}{2} \quad (13)$$

with $K_\theta(\bar{q}_k) = A_\theta^T(\bar{q}_k)A_\theta(\bar{q}_k)$ parameterized by a Cholesky factorization with the lower triangular matrix A_θ to ensure positive-semi-definiteness. For linear dissipation, F_θ is independent of $\bar{q}_k$.

4 Experiments

To demonstrate that the proposed model can learn and separate the conservative and non-conservative components of the observed system, we perform the following experiments: we train on data obtained from non-conservative systems, but after training, we turn off the learned external force term F_θ , and evaluate the rollout predictions for the corresponding conservative system. We will present the results obtained from the proposed model using a specific configuration of L_θ and F_θ architectures described in the previous section. Alternative configurations, based on the assumed priors, are presented in the appendix.

The model is evaluated in four learning tasks. Tasks 1 through 3 involve a damped double pendulum, a dissipative charged particle moving within a magnetic field, and pixel frames from a damped pendulum. Training data for these tasks are generated by solving the corresponding analytical ODEs with Gaussian noise applied to each sample. In task 4, we assess the model using a real-world dataset focused on human motion tracking. The primary experiments are now presented, while additional supporting experiments are given in the appendix.

Table 2: Extrapolation error (expressed as mean $\pm$ std over the test dataset) evaluated on a test dataset that possesses the same non-conservative characteristics as the training dataset. All evaluations are calculated at timestep $k = 35$. The best results are highlighted in **bold** text.

	NODE	GLNN	DFLNN (proposed)
Task 1: Damped Double Pendulum	0.050$\pm$0.034	0.42 $\pm$ 0.19	0.051 $\pm$ 0.022
Task 2: Dissipative Charged Particle	0.041 $\pm$ 0.015	4.0 $\cdot$ 10 ⁷ $\pm$ 1.7 $\cdot$ 10 ⁸	0.0072$\pm$0.0029
Task 3: Pixel Pendulum	0.023 $\pm$ 0.025	0.022 $\pm$ 0.009	0.0030 $\pm$ 0.0019
Task 4: Human Motion Capture	2.4 $\cdot$ 10 ⁷ $\pm$ 5.5 $\cdot$ 10 ⁷	NaN	11.75$\pm$10.89

Prediction workflow Unlike training, the prediction workflow involves forward time-stepping. The model is trained by minimizing the ODE residuals of the discrete Lagrange-d’Alembert equations (5) over observed segments to learn L_θ and F_θ . During the prediction phase, given two initial positions $\{q_0, q_1\}$, the model solves $\mathcal{E}_\Delta(q_0, q_1, \hat{q}_2) = 0$ for $\hat{q}_2$ using the learned L_θ and F_θ . This process is repeated recursively by feeding predictions as new inputs to generate longer rollouts.

To evaluate the models, we compare the *extrapolation error* as the root mean square associated with the k -th step for a predicted rollout: $\text{Extrapolation Error}_k := \frac{1}{N_\tau} \sum_{i=0}^{N_\tau} \|q_k^{\{i\}} - \hat{q}_k^{\{i\}}\|_2^2$, where i indexes each trajectory. The true trajectory is the solution at time step k given the same initial condition for $\{q_0, q_1\}$,

Baseline models A natural method to compare with is the Generalized Lagrangian Neural Network (GLNN) [3], which models forced Euler-Lagrange dynamics [2] using a finite difference approximation to $\dot{q}$. We also compare with Neural ODE [19], trained directly on position sequences without structural priors. The loss function is equal to the one given in [16]. See Appendix C for details.

Learning on learned latent variables from an autoencoder We also train a model on high-dimensional, unstructured pixel data $q \in \mathbb{R}^d$ by training an autoencoder to map the latent space $z \in \mathbb{R}^l$ (see Figure 6), where dynamics are modeled. The encoder $\phi_\theta : \mathbb{R}^d \rightarrow \mathbb{R}^l$ and decoder $\psi_\theta : \mathbb{R}^l \rightarrow \mathbb{R}^d$ are implemented as convolutional networks for image data or feedforward networks otherwise. The autoencoder is trained per timestep separately via the reconstruction loss

$$\mathcal{L}_{\text{AE}}(\phi_\theta, \psi_\theta)(q_n) = \frac{d}{l} \|q_n - \psi_\theta(\phi_\theta(q_n))\|_2^2. \quad (14)$$

Allowing the autoencoder to recover meaningful latent coordinates [33], we train the autoencoder simultaneously as the proposed model, weighted by ω_{AE} , resulting in the complete loss function:

$$\begin{aligned} \mathcal{L} = & \frac{\omega_{\text{physics}}}{N_{\mathcal{T}}(N-1)} \sum_{\mathcal{T}} \sum_{n=1}^{N-1} \mathcal{L}_{\text{physics}}(L_\theta, F_\theta) \left(\phi_\theta(q_{n-1}), \phi_\theta(q_n), \phi_\theta(q_{n+1}) \right) \\ & + \frac{\omega_{\text{reg}}}{R} \sum_{r=1}^R \mathcal{L}_{\text{reg}}(L_\theta) \left(\phi_\theta(q_r), \phi_\theta(q_{r+1}) \right) + \frac{\omega_{\text{AE}}}{N_{\mathcal{T}}(N+1)} \sum_{\mathcal{T}} \sum_{n=0}^N \mathcal{L}_{\text{AE}}(\phi_\theta, \psi_\theta)(q_n). \end{aligned} \quad (15)$$

During prediction, dynamics are computed fully in latent space; the autoencoder is used only to encode initial states and decode the rollout predictions.

Implementation The experiments were implemented in Python using the PyTorch framework with the Adam optimizer. See Appendix D for selected hyperparameters and additional computational notes. Code is available at [github](#).

4.1 Task 1: Damped Double Pendulum

We consider a double pendulum with two identical masses and synthetically generate displacement angle trajectories $\{\theta_1^t, \theta_2^t\}_{t=1}^T$, where $\theta_i^t \in \mathbb{R}$, over $T = 20$ time step and a step size $h = 0.1$. We train on 320 trajectories and evaluate on 10 trajectories. Gaussian noise $\mathcal{N}(0, \sigma^2)$ with variance $\sigma^2 = 10^{-2}h$ is added to each sample to simulate measurement uncertainty.

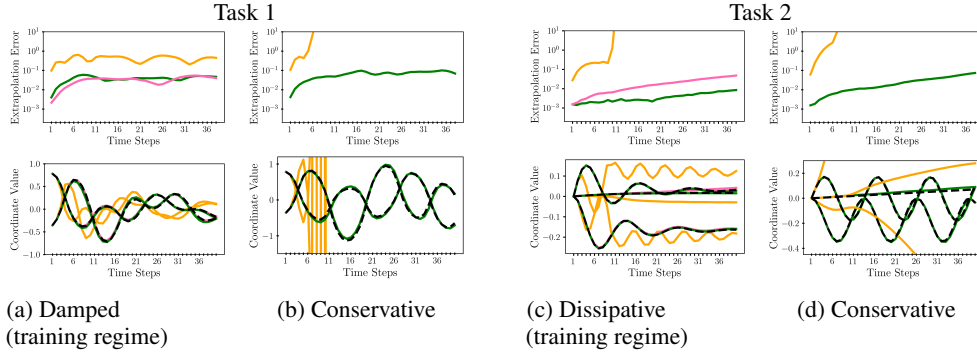


Figure 3: Combined results for Task 1 (left) and Task 2 (right). Solid Green lines are DFLNN (proposed), yellow lines are the GLNN model, and pink lines are a Neural ODE. The ground truth is indicated with dashed black. (a, c): Rollouts and extrapolation error for the trained model. (b, d): Turning off the external force component from the learned model to demonstrate the proposed model’s capabilities to distinguish the conserved dynamics (only applicable for DFLNN and GLNN).

The performance of the proposed model is shown in Figure 3, with metric evaluations in Table 2. Here, the Lagrangian is assumed to have the form [10] where $U = U(q)$, and the external force is linear [13]. The results show that the proposed model achieves comparable performance to the baseline models when applied to a damped system sharing the same characteristics as those present

in the training period. Moreover, when extrapolating onto a conserved regime, the proposed model maintains its accuracy, whereas the baseline models struggle to generalize. Notably, this enables us to make rollouts over a time-span that exceeds the intervals present in the training dataset.

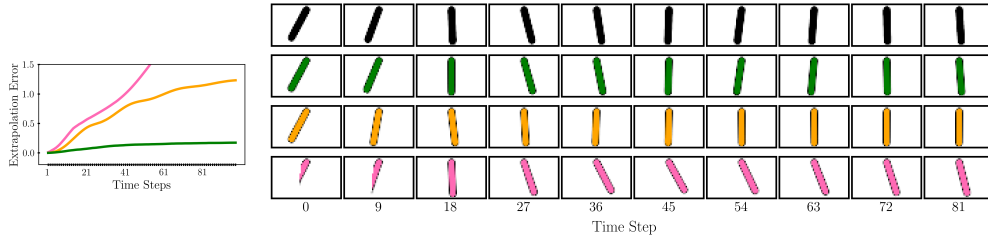
4.2 Task 2: Dissipative Charged Particle in a Magnetic Field

Next, we examine the behavior of a charged particle moving through a magnetic field. Our analysis considers a setup characterized by linear dissipation expressed in Cartesian coordinates. The training and test datasets are generated in a similar fashion to that employed in Task 4.1.

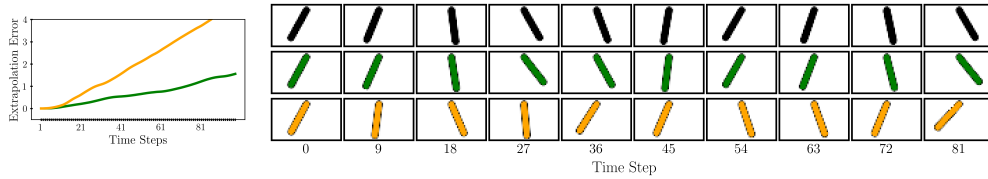
For a charged particle traveling in a magnetic field, the potential energy depends on both the position q and the velocity $\dot{q}$. Thus, we assume that the Lagrangian has the form (10) with $U = U(q, \dot{q})$, and the external force is linear (13). The performance of the proposed model is illustrated in Figure 3 with metric evaluations in Table 2. The proposed model shows its superiority over baseline models when rolling out predictions across several time steps, in both non-conservative regimes and when extrapolating to a conservative setting.

4.3 Task 3: Pixel data for a Damped Simple Pendulum

To assess the ability of the model to learn on latent embedding from an autoencoder, we use synthetic time series of images simulating a damped simple pendulum. Data are generated using the Gymnasium library [53], a successor to OpenAI’s Gym [54], producing $500 \times 500 \times 3$ RGB frames at each timestep. Damping is introduced by modifying the Pendulum-v1 environment to include a linear dissipative force. All images are cropped around the pendulum, converted to grayscale, and down sampled by a factor of 5. Initial displacements are limited to $\pi/6$ radians. The training dataset comprises 320 trajectories of $T = 100$ time steps each, where $q_t \in \mathbb{R}^{50 \times 30}$. The test dataset contains 10 trajectories. The use of an autoencoder is essential in this setting, reducing the state dimension from 50×30 to a latent space of dimension ($l = 1$); see Section 4.



(a) Damped Pixel Pendulum (training regime)



(b) Conservative Pixel Pendulum

Figure 4: Results for a simple pendulum represented through pixel images. Green pendulums are DFLNN (proposed), yellow are the GLNN model, and pink lines are a Neural ODE. The ground truth is indicated with black. (a): Rollouts and extrapolation error for the trained model. (b): Turning off the external force component from the learned model to demonstrate the proposed model’s capabilities to distinguish the conserved dynamics (Only applicable for DFLNN and GLNN).

To reduce the impact of autoencoder reconstruction errors during evaluation, we extract structural features using the Harris corner detection method [55]. Detected edges are binarized—assigning 1 to edge pixels and 0 elsewhere—thus focusing the evaluation on shape consistency rather than pixelwise accuracy. Similarity between predicted and ground truth edge maps is quantified using the

243 Dice coefficient [56], which measures overlap between two binary masks. This allows for robust
 244 extrapolation error evaluation even when raw pixel reconstruction is imperfect.

245 In Figure 4, we display both the extrapolation error and an example rollout. Metric evaluations are
 246 presented in Tabel 2. The findings reveal that the proposed model surpasses the baseline models
 247 in predictive rollouts. Additionally, the model exhibits interpretability by extrapolating beyond the
 248 training regime to a conserved setting by deactivating the identified external force component.

249 4.4 Task 4: Human Motion Capture

250 We evaluate our method on real-world data of a human swinging from a bar, using motion capture
 251 recordings from the CMU Graphics Lab Motion Capture Database [57, 58]. The dataset includes two
 252 recordings (subject 43, trials 2 and 3), which we jointly train on. The model is tasked with learning
 253 a shared Lagrangian and external force field that captures the dynamics underlying the swinging
 254 motion. We model the motion of 10 linked joints on the right side of the body—from the tibia
 255 to the radius—including rtibia, rfemur, rhipjoint, root, lowerback, upperback, thorax,
 256 rclavicle, rhumeral, and rradius. Due to the likely low intrinsic dimensionality of the motion,
 257 we apply an autoencoder to reduce the observed dimension $l = 6$.

258 To augment the data, every 10th frame is extracted as a separate trajectory, yielding 10 trajectories
 259 per trial— 20 trajectories in total. Joint angles, recorded at 120 Hz relative to a fixed reference, are
 260 converted to Cartesian coordinates and smoothed using a Savitzky-Golay filter [59].

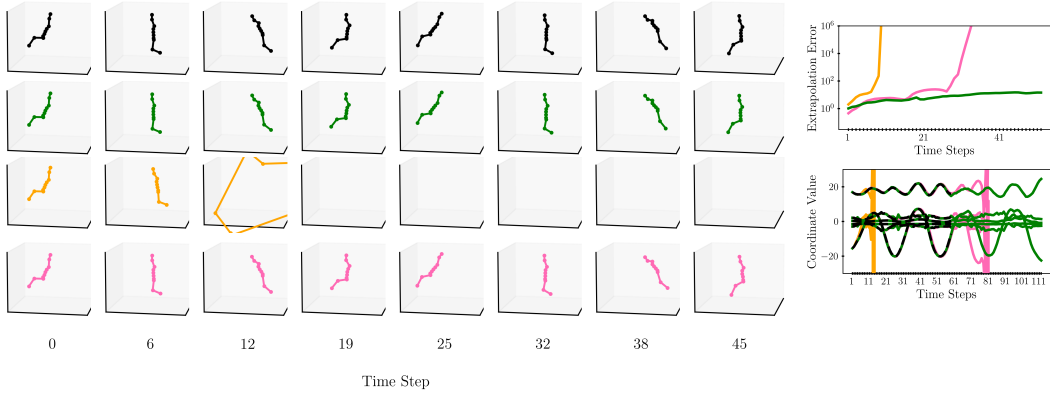


Figure 5: Results and reconstruction (trial 2) of human motion. Green notation is DFLNN (proposed), yellow is the GLNN model, and pink lines a Neural ODE. The ground truth is indicated in black. The left panel depicts the full movement represented as a skeletal sketch. The right panel shows the extrapolation error (top) and the rollout trajectory for the right femur (bottom).

261 We hypothesize that a person swinging from a bar behaves similarly to a multi-pendulum system
 262 and aim to identify a Lagrangian with mechanical structure with generalized potential $U(q, \dot{q})$. The
 263 external force is assumed to comprise both frictional dissipation and active energy input from the
 264 subject. Accordingly, we model it as the sum of a nonlinear dissipative term plus a neural network
 265 F_{θ}^{Free} to model the forces exerted by the human motion $F_{\theta} = -K_{\theta}(q)\dot{q} + F_{\theta}^{\text{Free}}$.

266 Figure 5 shows that the model accurately reconstructs the motion, suggesting it has learned general-
 267 izable governing equations. When extrapolating beyond the training window, (after ~ 60 frames)
 268 the proposed model continues to produce plausible trajectories, whereas baseline models diverge
 269 significantly earlier.

270 5 Conclusions

271 We have proposed a method for learning mechanical systems from data using the Lagrange-
 272 d'Alembert principle. The approach requires the use of position data only, can handle both conserva-
 273 tive and dissipative systems, generalizes well when trained and validated on both synthetic as well as
 274 real-world data.

Acknowledgments

The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

References

- [1] Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.
- [2] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.
- [3] Shanshan Xiao, Jiawei Zhang, and Yifa Tang. Generalized lagrangian neural networks. *arXiv preprint arXiv:2401.03728*, 2024.
- [4] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Dissipative symoden: Encoding hamiltonian dynamics with dissipation and control into deep learning. *arXiv preprint arXiv:2002.08860*, 2020.
- [5] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [6] Maki K Habib, Samuel A Ayankoso, and Fusaomi Nagata. Data-driven modeling: concept, techniques, challenges and a case study. In *2021 IEEE international conference on mechatronics and automation (ICMA)*, pages 1000–1007. IEEE, 2021.
- [7] Shen Yin, Steven X Ding, Xiaochen Xie, and Hao Luo. A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial electronics*, 61(11):6418–6428, 2014.
- [8] Shivani Gupta and Atul Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019.
- [9] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022.
- [10] Rick Chartrand. Numerical differentiation of noisy, nonsmooth data. *International Scholarly Research Notices*, 2011(1):164564, 2011.
- [11] Discrete mechanics and variational integrators. *Acta numerica*, 10:357–514, 2001.
- [12] Thomas B Moeslund, Adrian Hilton, and Volker Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3): 90–126, 2006.
- [13] Trong-Nguyen Nguyen, Huu-Hung Huynh, and Jean Meunier. Skeleton-based abnormal gait detection. *Sensors*, 16(11):1792, 2016.
- [14] Sara Sardari, Sara Sharifzadeh, Alireza Daneshkhah, Bahareh Nakisa, Seng W Loke, Vasile Palade, and Michael J Duncan. Artificial intelligence for skeleton-based physical rehabilitation action evaluation: A systematic review. *Computers in Biology and Medicine*, 158:106835, 2023.
- [15] Jianping Guo, Hong Liu, Xi Li, Dahong Xu, and Yihan Zhang. An attention enhanced spatial-temporal graph convolutional lstm network for action recognition in karate. *Applied Sciences*, 11(18):8641, 2021.
- [16] Sina Ober-Blobbaum and Christian Offen. Variational learning of euler-lagrange dynamics from data. *Journal of Computational and Applied Mathematics*, 421:114780, 2023.
- [17] Shaan A Desai, Marios Mattheakis, David Sondak, Pavlos Protopapas, and Stephen J Roberts. Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E*, 104(3):034312, 2021.

- [18] Andrew Sosanya and Sam Greydanus. Dissipative hamiltonian neural networks: Learning dissipative and conservative dynamics separately. *arXiv preprint arXiv:2201.10085*, 2022.
- [19] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [20] Yuandi Wu, Brett Sicard, and Stephen Andrew Gadsden. Physics-informed machine learning: A comprehensive review on applications in anomaly detection and condition monitoring. *Expert Systems with Applications*, page 124678, 2024.
- [21] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, and Nuno Carvalhais. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.
- [22] Raban Iten, Tony Metger, Henrik Wilming, Lidia Del Rio, and Renato Renner. Discovering physical concepts with neural networks. *Physical review letters*, 124(1):010508, 2020.
- [23] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering*, 29(10):2318–2331, 2017.
- [24] Amirhassan Abbasi, Prashant N Kambali, Parham Shahidi, and C Nataraj. Physics-informed machine learning for modeling multidimensional dynamics. *Nonlinear Dynamics*, 112(24):21565–21585, 2024.
- [25] Symplectic integration of learned hamiltonian systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(1), 2022.
- [26] Takashi Matsubara, Ai Ishikawa, and Takaharu Yaguchi. Deep energy-based modeling of discrete-time physics. *Advances in Neural Information Processing Systems*, 33:13100–13111, 2020.
- [27] Yuhan Chen, Takashi Matsubara, and Takaharu Yaguchi. Neural symplectic form: Learning hamiltonian equations on general coordinate systems. *Advances in Neural Information Processing Systems*, 34:16659–16670, 2021.
- [28] Aleksei Sholokhov, Yuying Liu, Hassan Mansour, and Saleh Nabi. Physics-informed neural ode (pinode): embedding physics into models using collocation points. *Scientific Reports*, 13(1):10166, 2023.
- [29] Elena Celledoni, Sølve Eidnes, and Håkon Noren Myhr. Learning dynamical systems from noisy data with inverse-explicit integrators. *Physica D: Nonlinear Phenomena*, 472:134471, 2025.
- [30] Benjamin K Tapley. Symplectic neural networks based on dynamical systems. *arXiv preprint arXiv:2408.09821*, 2024.
- [31] Jianyu Hu, Juan-Pablo Ortega, and Daiying Yin. A structure-preserving kernel method for learning hamiltonian systems. *arXiv preprint arXiv:2403.10070*, 2024.
- [32] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.
- [33] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.
- [34] Marc Finzi, Ke Alexander Wang, and Andrew G Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. *Advances in neural information processing systems*, 33:13880–13889, 2020.

- [35] Elena Celledoni, Andrea Leone, Davide Murari, and Brynjulf Owren. Learning hamiltonians of constrained mechanical systems. *Journal of Computational and Applied Mathematics*, 417: 114608, 2023.
- [36] J.P Ortega J. Hu and D. Yin. A structure-preserving kernel method for learning hamiltonian systems. *arXiv preprint arXiv:2403.10070*, 2024.
- [37] J.P Ortega and D. Yin. Expressiveness and structure preservation in learning Port-Hamiltonian systems. *International Conference on Geometric Science of Information*, pages 313–322, 2024.
- [38] J.P Ortega and D. Yin. Learnability of linear port-hamiltonian systems. *Journal of Machine Learning Research*, 25:1–56, 2024.
- [39] Lyudmila Grigoryeva, Allen Hart, and Juan-Pablo Ortega. Learning strange attractors with reservoir systems. *Nonlinearity*, 36(9):4674, 2023.
- [40] Takehiro Aoshima, Takashi Matsubara, and Takaharu Yaguchi. Deep discrete-time lagrangian mechanics.
- [41] Hong Qin. Machine learning and serving of discrete field theories. *Scientific Reports*, 10(1): 19329, 2020.
- [42] Michael Lutter, Christian Ritter, and Jan Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.
- [43] Christian Offen and Sina Ober-Blöbaum. Learning of discrete models of variational pdes from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(1), 2024.
- [44] Christian Offen. Machine learning of continuous and discrete variational odes with convergence guarantee and uncertainty quantification. *arXiv preprint arXiv:2404.19626*, 2024.
- [45] Juan-Pablo Ortega and Daiying Yin. Learnability of linear port-hamiltonian systems. *Journal of Machine Learning Research*, 25(68):1–56, 2024.
- [46] Sølve Eidnes and Kjetil Olsen Lye. Pseudo-hamiltonian neural networks for learning partial differential equations. *Journal of Computational Physics*, 500:112738, 2024.
- [47] Anthony Gruber, Kookjin Lee, Haksoo Lim, Noseong Park, and Nathaniel Trask. Efficiently parameterized neural metriplectic systems. *arXiv preprint arXiv:2405.16305*, 2024.
- [48] D Martín De Diego and R Sato Martín de Almagro. Variational order for forced lagrangian systems. *Nonlinearity*, 31(8):3814, 2018.
- [49] Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *International Journal for numerical methods in engineering*, 49(10):1295–1325, 2000.
- [50] Yana Lishkova, Paul Scherer, Steffen Ridderbusch, Mateja Jamnik, Pietro Liò, Sina Ober-Blöbaum, and Christian Offen. Discrete lagrangian neural networks with automatic symmetry discovery. *IFAC-PapersOnLine*, 56(2):3203–3210, 2023.
- [51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [52] Rayleigh’s dissipation function at work. *European Journal of Physics*, 36(3):035014, 2015.
- [53] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.
- [54] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- [55] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. Citeseer, 1988.

- 411 [56] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):
412 297–302, 1945.
- 413 [57] Carnegie Mellon University. Cmu graphics lab motion capture database. Data obtained from
414 <https://mocap.cs.cmu.edu>.
- 415 [58] Fernando De la Torre, Jessica Hodgins, Adam Bargteil, Xavier Martin, Justin Macey, Alex
416 Collado, and Pep Beltran. Guide to the carnegie mellon university multimodal activity (cmu-
417 mmac) database.
- 418 [59] Abraham Savitzky. Smoothing and differentiation of data by simplified least squares procedures.
419 *Analytical chemistry*, 36(8):1627–1639, 1964.
- 420 [60] Ernst Hairer and Christian Lubich. Symmetric multistep methods over long times. *Numerische*
421 *Mathematik*, 97:699–723, 2004.
- 422 [61] Ernst Hairer and Christian Lubich. Symmetric multistep methods for charged-particle dynamics.
423 *The SMAI Journal of computational mathematics*, 3:205–218, 2017.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main contributions of this paper have been summarized in section [1.2](#)

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are being discussed in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: All information about hyperparameters and data generation used for the experiments are given in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Link to repository is given in section 4. The link is inactive at submission due to blind review, but will be replaced after acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In addition to the description of data generation and model parameters given in section 4, we provide full tables of hyperparameters in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Results with standard deviations computed across all samples in the test dataset are included in Tabel 2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: This is included in the appendix when providing supporting experiments for the main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The dataset we utilize is publicly accessible and the creator encourages its use for research purposes. The subjects' identities have been anonymized by the creators.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: The paper aims to predict the dynamics of mechanical systems. This has minimal societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We include a paragraph in the Acknowledgements section in accordance with the CMU database.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- 733 • We recognize that the procedures for this may vary significantly between institutions
734 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the
735 guidelines for their institution.
736 • For initial submissions, do not include any information that would break anonymity (if
737 applicable), such as the institution conducting the review.

738 **16. Declaration of LLM usage**

739 Question: Does the paper describe the usage of LLMs if it is an important, original, or
740 non-standard component of the core methods in this research? Note that if the LLM is used
741 only for writing, editing, or formatting purposes and does not impact the core methodology,
742 scientific rigorousness, or originality of the research, declaration is not required.

743 Answer: [NA]

744 Justification:

745 Guidelines:

- 746 • The answer NA means that the core method development in this research does not
747 involve LLMs as any important, original, or non-standard components.
748 • Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>)
749 for what should or should not be described.