Shape analysis and structure preservation

Elena Celledoni

Department of Mathematical Sciences, NTNU, Trondheim, Norway

Woudschoten Conferentiecentrum, Zeist,
24-26 September, 2025

Woudschoten Conference, 2025

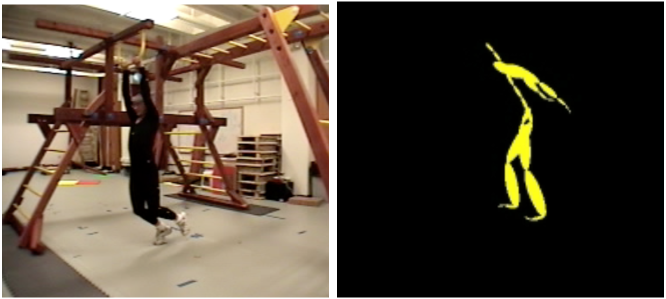
MSCA-SE: REMODEL Project ID: 101131557

Outline

- Introduction: shape analysis for curves and surfaces, and for human motion data.
- Learning the equations of motion from data using a discrete Lagrange d'Alembert principle.
- Examples on synthetic data, pixel data and real-world data.
- Other ongoing and future work.

Given N *observation of previous values of a time series* $\{q_n\}_n$ *forecast future values:*

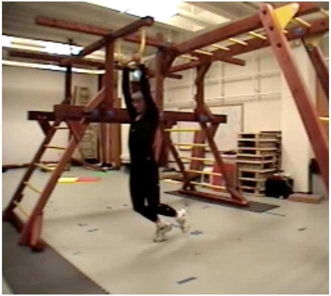
- Directly from the data.
- By learning (discrete) mechanical equations first.

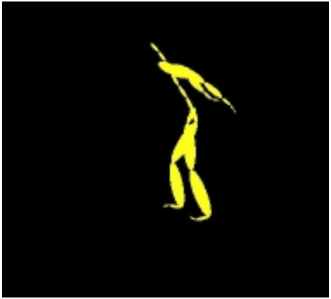


CMU Graphics Lab Motion Capture Database

Given N observation of previous values of a time series $\{q_n\}_n$ forecast future values:

- Directly from the data.
- By learning (discrete) mechanical equations first.





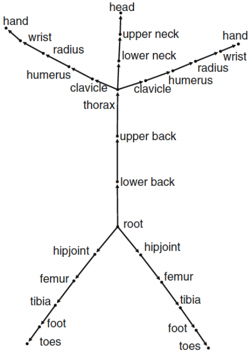
CMU Graphics Lab Motion Capture Database

Some challenges:

- Find appropriate metrics.
- Deal with constraints.
- Learning Hamiltonians / Lagrangians for mechanical systems.
- Learn external forces.
- Provide enough data.

Skeletal animation

Skeleton consisting of bones connected by joints. One 3D rotation for each joint.

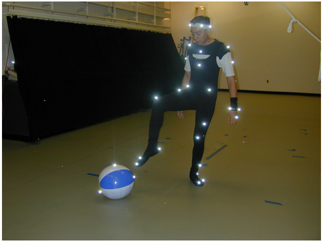


Human activity: $g : [0, T] \to \mathcal{J}$, $\mathcal{J} = \mathrm{SO}(3)^n$, where $[0, T]$ is a time interval.

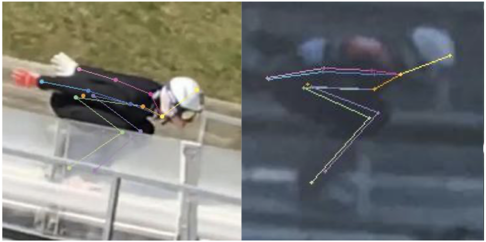
- Data obtained by motion capturing.
- Motion manipulation is the processing of the data.
- CMU Motion Capturing Database.

Motion capturing: only data and one simple differential equation

Only data: $g : [0, T] \to SO(3)^n$



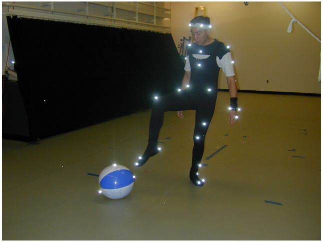
CMU Motion Capture Database



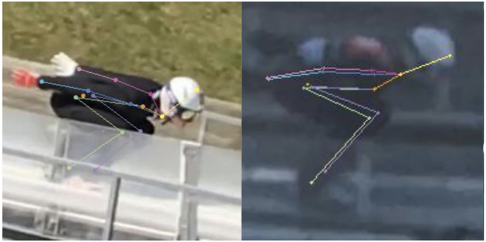
R. O. Hide, NTNU, master thesis

Motion capturing: only data and one simple differential equation

Only data: $g : [0, T] \to SO(3)^n$



CMU Motion Capture Database



R. O. Hide, NTNU, master thesis

Motion manipulation done via shape analysis

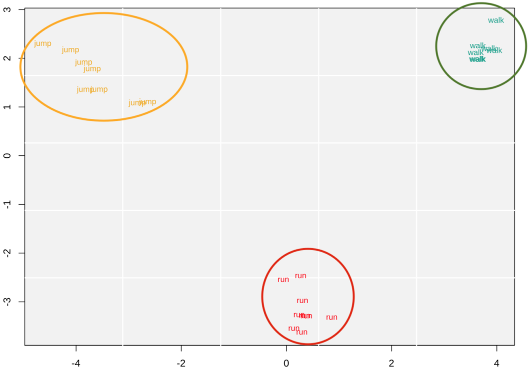
and we use one simple differential equation

$$\dot{g} = q(t)g(t), \qquad g(0) = e.$$

expressing the need of derivative information

- M. Eslitzbichler, *Modelling character motions on infinite-dimensional manifolds*, The Visual Computer, 2014.
- E. C., M. Eslitzbichler and A. Schmeding, *Shape analysis on Lie groups with applications in computer animation*, J Geometric Mechanics, 2016.
- E. C., S. Eidnes and A. Schmeding, *Shapes on homogeneous manifolds*, The Abel Symposium, 187-220, 2018.
- P. E. Lystad, *Signatures in Shape Analysis*, Master Thesis, NTNU, 2019.
- E. C., H. Glöckner, J. Riseth and A. Schmeding, *Deep learning of diffeomorphisms for optimal reparametrizations of shapes*, BIT, 2023.

Classifying running, walking, jumping animations as shapes (DP)



- $SO(3)^n$-valued time curves: the geometry of rotations is preserved.
- We treat time curves as shapes: use a reparametrisation invariant distance function: $d_{\mathcal{P}}$
- Dynamic programming, to find the optimal reparametrisation.
- Classical multidimensional scaling to visualize distances in a lower dimensional space.

- E. C., P.E. Lystad, N.Tapia, Signatures in Shape Analysis: an Efficient Approach to Motion Identification, Proceedings of the GSI conference 2019.

- Shapes are *unparametrized curves* (or surfaces) in a vector space or on a manifold.
- The use of shapes is natural in applications where one wants to compare curves or surfaces independently of their parametrisation.

Definition of **shapes** via an equivalence relation: let $\mathrm{I} \subset \mathbb{R}$ an interval, consider

$$\mathcal{P} := \mathrm{Imm}(\mathrm{I}, \mathbb{R}^n) = \{c \in C^\infty(\mathrm{I}, \mathbb{R}^n) \mid \dot{c}(t) \neq 0\},$$

$\mathcal{P}$ is called pre-shape space and is an **infinite dimensional manifold**. Let $c_0, c_1 \in \mathcal{P}$ then

$$c_0 \sim c_1 \Longleftrightarrow \exists\, \varphi : \quad c_0 = c_1 \circ \varphi$$

with $\varphi \in \mathrm{Diff}^+(\mathrm{I})$ a orientation preserving diffeomorphism on I

Shape space:

$$\mathcal{S} := \mathrm{Imm}(\mathrm{I}, \mathbb{R}^n)/\mathrm{Diff}^+(\mathrm{I})$$

Distance function on $\mathcal{S} = \mathcal{P}/\mathrm{Diff}^+(I)$

Definition of a **distance** function $d_{\mathcal{S}}$ to measure similarities between two shapes $[c_0]$ and $[c_1]$. The distance function on $\mathcal{S}$ should be independent of the choice of representatives.

Definition of a **distance** function $d_{\mathcal{S}}$ to measure similarities between two shapes $[c_0]$ and $[c_1]$. The distance function on $\mathcal{S}$ should be independent of the choice of representatives.

Let $d_{\mathcal{P}}$ be a reparametrization invariant distance function on $\mathcal{P}$ i.e.

$$d_{\mathcal{P}}(c_0, c_1) = d_{\mathcal{P}}(c_0 \circ \varphi, c_1 \circ \varphi) \quad \forall \varphi \in \mathrm{Diff}^+(\mathrm{I}).$$

Definition of distance on $\mathcal{S}$:

$$d_{\mathcal{S}}([c_0],[c_1]) := \inf_{\varphi \in \mathrm{Diff}^+(\mathrm{I})} d_{\mathcal{P}}(c_0, c_1 \circ \varphi).$$

Proposition

If $d_{\mathcal{P}}$ is a reparametrization invariant distance function on $\mathcal{P}$, then $d_{\mathcal{S}}([c_0],[c_1])$ is independent of the choice of representatives of $[c_0]$ and $[c_1]$.

Distance on $\mathcal{P}$ via **SRVT** *and Q-transform*

One can proceed first transforming the curves and then computing L_2 distances:

$$Q : \mathcal{P} \to C^\infty(I, \mathbb{R}^n), \quad c \mapsto q,$$

$$Q : \mathcal{P} \to C^\infty(I, \mathbb{R}^n), \quad c \mapsto q, \quad Q(c) := \begin{cases} \frac{\dot{c}}{\sqrt{\|\dot{c}\|}} & \text{SRVT} \\ \sqrt{\|\dot{c}(\cdot)\|}\, c(\cdot) & Q-\text{transform} \end{cases}$$

$$d_{\mathcal{P}}(c_0, c_1) = d_{L^2}(Q(c_0), Q(c_1)) = \|q_0 - q_1\|_{L_2}.$$

With this definition $d_{\mathcal{P}}$ is reparametrization invariant and $d_{\mathcal{S}}$ is well defined, because Q is equivariant wrt reparametrizations:

$$Q(c \circ \varphi)(t) = \sqrt{\dot{\varphi}(t)} \cdot (Q(c) \circ \varphi)(t)$$

- A. Srivastava, E. Klassen, S.H. Joshi, and I.H. Jermyn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.
- M. Mani, S. Kurtek, C. Barillot, A. Srivastava, *IEEE Symposium on Biomedical Imaging*, 2010.

Optimal reparametrization problem

Optimal reparametrization problem: let $q_0 = Q(c_0)$, $q_1 = Q(c_1)$, given shapes $[c_0]$ and $[c_1]$

$$d_{\mathcal{S}}([c_0],[c_1]) = \inf_{\varphi \in \mathrm{Diff}^+(\mathrm{I})} E(\varphi), \qquad E(\varphi) = d_{\mathcal{P}}(c_0, c_1 \circ \varphi) = \|q_0 - \sqrt{\dot{\varphi}}\,(q_1 \circ \varphi)\|^2_{L_2}.$$

Optimisation problem on an infinite dimensional Lie group $\mathrm{Diff}^+(\mathrm{I})$, with "Lie algebra" $T_{\mathrm{id}}\mathrm{Diff}^+(\mathrm{I})$, $I = [0,1]$.
We parametrize the diffeomorphisms with deep neural networks as follows:

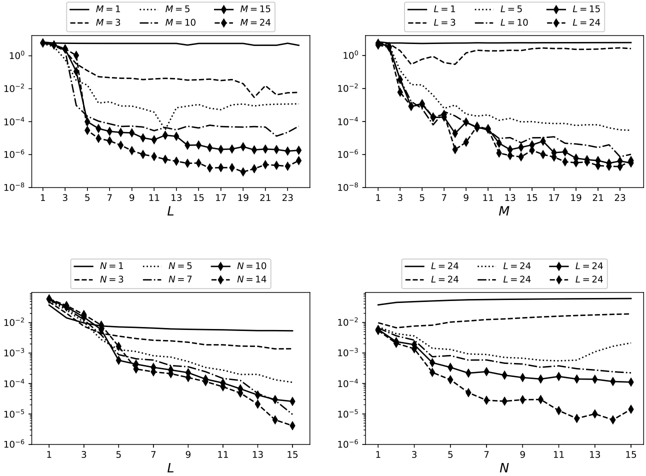
Consider a basis $v_1, v_2, \dots$ of $T_{\mathrm{id}}\mathrm{Diff}^+(\mathrm{I})$ write

$$\varphi_\theta = (\mathrm{id} + h_1 f_{\theta_1}) \circ \cdots \circ (\mathrm{id} + h_L f_{\theta_L}).$$

$$f_{\theta_\ell} = \sum_{j=1}^{M} \beta_j^\ell v_j, \qquad \theta_\ell = \{\beta_j^\ell\}_{j=1,\dots,M}^{\ell=1,\dots,L}$$

Optimise on these "approximate diffeomorphisms".

- E. Celledoni, H. Glöckner, J. Riseth and A. Schmeding, *Deep neural networks on diffeomorphism groups for optimal shape reparametrization*, BIT, 2023.


M = 1
M = 5
M = 15
M = 3
M = 10
M = 24
L
L = 1
L = 5
L = 15
L = 3
L = 10
L = 24
M
N = 1
N = 5
N = 10
N = 3
N = 7
N = 14
L
L = 24
L = 24
L = 24
L = 24
L = 24
L = 24
N

Parametric surfaces embedded in $\mathbb{R}^3$

Domain $\Omega = [0,1]^2$. Denoting by $f : \Omega \to \mathbb{R}^3$ the parametric surface, f_x and f_y the partial derivatives.

Normal vector and area scaling factor:

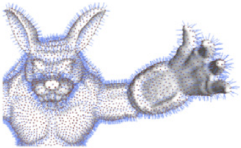
$$\mathrm{n}_f(x,y) = f_x \times f_y, \qquad a_f(x,y) = |f_x \times f_y|$$

then

$$\mathcal{P} = \{f \in C^\infty(\Omega,\mathbb{R}) \,|\, a_f(x,y) > 0, \forall (x,y) \in \Omega\}$$

Implicit neural representations: $g : \mathbb{R}^3 \to \mathbb{R}$, e.g. the *signed distance function* approximated by a neural network g_θ

$$\mathcal{M} = g^{-1}(0).$$

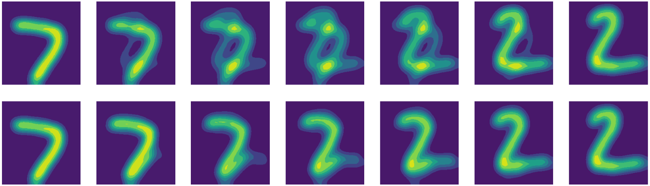


oriented point cloud

- I.H. Jermyn, et al., *Elastic shape matching of parameterized surfaces using square root normal fields.* In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.) Computer Vision, ECCV 2012, Springer, (2012).
- S. Kurtek, et al., *A novel Riemannian framework for shape analysis of 3d objects.* In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2010).
- L. Schirmer et al., *Geometric implicit neural representations for signed distance functions*, Computers and Graphics, 2022.

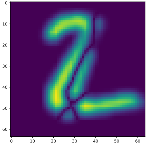
Data generation, optimal deformation

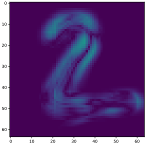
MNIST, matching of images, handwritten digits

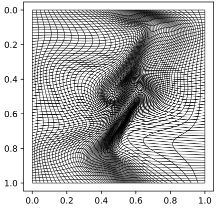


Top $\lambda(f_1, f_2, \tau) := \tau f_1 + (1-\tau) f_2$

Bottom $\gamma(f_1, f_2, \tau) := \tau f_1 + (1-\tau) f_2 \circ \varphi^*$, φ^* optimal reparametrization.



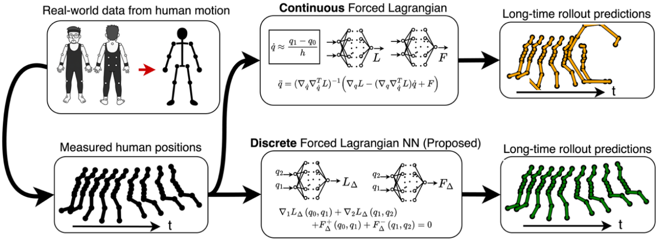




- E. C., H. Glöckner, J. Riseth and A. Schmeding, *Deep learning of diffeomorphisms for optimal reparametrizations of shapes*, BIT, 2023.

Learning Mechanical systems from data

Given N observation of previous values of a time series $\{q_n\}_n$ forecast future values



CMU Graphics Lab Motion Capture Database

- We assume to measure **only position data**.
- Necessary to take into account external forces and dissipative forces.
- We **use the laws of physics**.
- We want to learn separately the the **external forces** from the **conservative forces** (coming from the the Lagrangian).
- Optional: we learn separately kinetic energy (the mass matrix) and potential energy.

Joint work with M. D. Hansen and B.K. Tapley

Lagrange d'Alembert principle - Balance of forces

LdA principle

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

where δ denotes variations that vanish at the endpoints

$q(t_0)=q_0$ and $q(t_N)=q_N$,

$L:TQ\to\mathbb{R}$ is the Lagrangian function

$F:TQ\to TQ^*$ is the Lagrangian force, a fiber preserving map
$F:(q,\dot{q})\mapsto(q,F(q,\dot{q}))$,

Lagrange d'Alembert principle - Balance of forces

LdA principle

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

where δ denotes variations that vanish at the endpoints

- $q(t_0)=q_0$ and $q(t_N)=q_N$,
- $L:TQ\to\mathbb{R}$ is the Lagrangian function
- $F:TQ\to TQ^*$ is the Lagrangian force, a fiber preserving map $F:(q,\dot{q})\mapsto(q,F(q,\dot{q}))$,

Forced Euler- Lagrange Equations (equivalent to LdA)

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

LdA suggests that we should learn L separately from F.

Lagrange d'Alembert principle - Balance of forces

LdA principle

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

where δ denotes variations that vanish at the endpoints

$q(t_0)=q_0$ and $q(t_N)=q_N$,

$L: TQ\to\mathbb{R}$ is the Lagrangian function

$F: TQ\to TQ^*$ is the Lagrangian force, a fiber preserving map
$F:(q,\dot{q})\mapsto(q,F(q,\dot{q}))$,

Forced Euler- Lagrange Equations (equivalent to LdA)

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

LdA suggests that we should learn L separately from F.

In the spirit of *Geometric Mechanics-Variational Integrators*, we use a **discrete LdA**.

- J E Marsden and M West, *Discrete variational integrators*, Acta Numerica, (2001);
- DM de Diego, R. Sato Martin de Almagro, *Variational order for forced Lagrangian systems*, Nonlinearity, 2018.

Discrete Lagrange d'Alembert principle: using only position data

$$L_\Delta(q_n, q_{n+1}) \approx \int_{t_n}^{t_n+h} L\big(q(t), \dot{q}(t)\big)\, dt,$$

$$F_\Delta^-(q_n, q_{n+1}, h)\cdot \delta q_n + F_\Delta^+(q_n, q_{n+1}, h)\cdot \delta q_{n+1} \approx \int_{t_n}^{t_n+h} F\big(q(t), \dot{q}(t)\big)\cdot \delta q(t)\, dt,$$

$L_\Delta : Q \times Q \to \mathbb{R}$ (discrete Lagrangian), $F_\Delta^\pm : Q \times Q \to T^*Q$ (discrete force)

$$\pi_Q^- = \pi_Q \circ F_\Delta^- \qquad \pi_Q^- : Q\times Q \to Q, \qquad \pi_Q^+ = \pi_Q \circ F_\Delta^+ \qquad \pi_Q^+ : Q\times Q \to Q,$$

with $\pi_Q : T^*Q \to Q$ the projection.

The discrete Lagrange-d'Alembert principle: find a discrete trajectory $\{q_n\}_{n=1}^N$ s.t.

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

for all variations $\{\delta q_n\}_{n=0}^N$ vanishing at the endpoints $\delta q_0 = \delta q_N = 0$.

Discrete forced Euler-Lagrange equations:

$$0 = \mathcal{E}_\Delta(L_\Delta, F_\Delta)(q_{n-1}, q_n, q_{n+1}) := D_2\, L_\Delta(q_{n-1}, q_n) + D_1\, L_\Delta(q_n, q_{n+1}) \\ + F_\Delta^+(q_{n-1}, q_n) + F_\Delta^-(q_n, q_{n+1}), \quad n = 2, \ldots, N-1,$$

where D_1 and D_2 denote differentiation with respect to the first and second variable.

Goal: learn NNs $L_\theta \approx L$ and $F_\theta \approx F$ from the observed trajectories of the mechanical system

Direct Discrete Lagrange d'Alembert setting: Given L Lagrangian and F forces compute trajectories $\{q_n\}_{n=0}^N$ of the mechanical system satisfying Discrete LdA principle.

Inverse discrete LdA setting:
Given multiple observed (discrete) trajectories $\{q_n\}_{n=0}^N$ we seek approximations to the Lagrangian and forces (L, F).
We replace (L, F) with neural networks approximations

$$L_\theta \approx L, \qquad F_\theta \approx F$$

(L_θ, F_θ) are then the unknowns of our problem which we find minimizing the loss function

$$\mathcal{L} = \mathcal{L}_{\text{mech}}(L_\theta, F_\theta) + \mathcal{L}_{\text{reg}}$$

Example - discretization of the LdA - loss function to find the parameters

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

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

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

where

$$\bar{q}_{n+\frac{1}{2}} = \frac{1}{2}(q_n + q_{n+1})$$
$$\bar{\dot{q}}_{n+\frac{1}{2}} = \frac{1}{h}(q_{n+1} - q_n).$$

Loss function is the residual of the discrete forced Euler- Lagrange equations:

$$\mathcal{L}_{\text{mech}}(L_\theta, F_\theta) = \frac{h}{2}\sum_{\mathcal{T}}\sum_{n=2}^{N-1}\Bigg\| D_1\,L_\theta\left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}}\right) + \frac{2}{h}\,D_2\,L_\theta\left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}}\right)$$
$$+ D_1\,L_\theta\left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}}\right) - \frac{2}{h}\,D_2\,L_\theta\left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}}\right)$$
$$+ F_\theta\left(\bar{q}_{n-\frac{1}{2}}, \bar{\dot{q}}_{n-\frac{1}{2}}\right) + F_\theta\left(\bar{q}_{n+\frac{1}{2}}, \bar{\dot{q}}_{n+\frac{1}{2}}\right)\Bigg\|_2$$

Structure of Lagrangians, Forces and Regularization

- **free**: $L_\theta = L_\theta(q,\dot{q})$, $F_\theta = F_\theta(q,\dot{q})$, feed forward NNs.
- **structured**:
  - a mechanical Lagrangian

    $$L_\theta(q,\dot{q}) = \frac{1}{2}\dot{q}^T M_\theta(q)\dot{q} - U_\theta(q),$$

    with $M_\theta(q)$ SPD:

    $$M_\theta(q) = \epsilon I + C_\theta^T(q)C_\theta(q)$$

    learn M_θ through the factors C_θ, and learn U_θ,
  - and dissipative force $F_\theta = -K_\theta\dot{q}$ with $K_\theta = K_\theta^T$ a symmetric positive definite matrix.

- **free**: $L_\theta = L_\theta(q,\dot{q})$, $F_\theta = F_\theta(q,\dot{q})$, feed forward NNs.
- **structured**:
  - a mechanical Lagrangian

$$L_\theta(q,\dot{q}) = \frac{1}{2}\dot{q}^T M_\theta(q)\dot{q} - U_\theta(q),$$

with $M_\theta(q)$ SPD:

$$M_\theta(q) = \epsilon I + C_\theta^T(q)C_\theta(q)$$

learn M_θ through the factors C_θ, and learn U_θ,

  - and dissipative force $F_\theta = -K_\theta\dot{q}$ with $K_\theta = K_\theta^T$ a symmetric positive definite matrix.

Regularization: Impose L_θ to be a *regular Lagrangian*, i.e.

$$S := \left(\frac{\partial^2 L_\theta(q,\dot{q})}{\partial\dot{q}_i\partial\dot{q}_j}\right),$$

$$\mathcal{L}_{\text{reg}} = \frac{1}{N_r}\sum_{r=1}^{N_r}\left|\log\left(|\det(S_r)|\right)\right|, \qquad S_r = S(\bar{q}_{r+\frac{1}{2}},\bar{\dot{q}}_{r+\frac{1}{2}})$$

on N_r regularization points given from the training dataset.

Related work: by Offen and Ober-Blöbaum

- Propose to learn the Lagrangian of Euler-Lagrange dynamics, with no external forces. Approximate the Lagrangian using Gaussian Processes.
- Discuss the problem of non uniqueness of the Lagrangian and propose ways to learn regular Lagrangians.
- Introduce the concept of *inverse modified Lagrangian* $L_{invmod} \approx L$ and

$$L = L_{invmod} + \tilde{L}\, h^p + \mathcal{O}(h^{p+1})$$

- For more accurate approximations of L use *Lagrangian Backward Error Analysis* and L_{invmod}.
- Use only position data.
- Experiments on synthetic data.

- S Ober-Blöbaum, C. Offen, *Variational Learning of Euler-Lagrange Dynamics from Data*. Journal of Computational and Applied Mathematics, 421, 2023.
- C. Offen, S. Ober-Blöbaum, *Symplectic integration of learned Hamiltonian systems*, Chaos, 2022.
- C. Offen, S. Ober-Blöbaum, *Learning discrete Lagrangians for variational PDEs from data and detection of travelling waves*, 2023, arXiv preprint arXiv:2302.08232.

Higher order approximations of L and F

- We use a symmetric multi-step approach.
- Symmetric multi-step variational discretization have good geometric properties. (Hairer and Lubich)

- We use a symmetric multi-step approach.
- Symmetric multi-step variational discretization have good geometric properties. (Hairer and Lubich)

Order 4:

$$L_\Delta(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = hL_\theta(\bar{q}_n, \bar{v}_n),$$
$$F^+_\Delta(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = \frac{h}{2}F_\theta(\bar{q}_n, \bar{v}_n),$$
$$F^-_\Delta(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = \frac{h}{2}F_\theta(\bar{q}_n, \bar{v}_n),$$

where

$$\bar{q}_n = \frac{1}{12}(-q_{n-2} + 8q_{n-1} + 4q_{n+1} + q_{n+2}), \qquad \bar{v}_n = \frac{1}{12h}(q_{n-2} - 8q_{n-1} + 8q_{n+1} - q_{n+2}).$$

- We use a symmetric multi-step approach.
- Symmetric multi-step variational discretization have good geometric properties. (Hairer and Lubich)

Order 4:

$$L_\Delta(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = hL_\theta(\bar{q}_n, \bar{v}_n),$$
$$F_\Delta^+(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = \frac{h}{2}F_\theta(\bar{q}_n, \bar{v}_n),$$
$$F_\Delta^-(q_{n-2}, q_{n-1}, q_{n+1}, q_{n+2}) = \frac{h}{2}F_\theta(\bar{q}_n, \bar{v}_n),$$

where

$$\bar{q}_n = \frac{1}{12}(-q_{n-2} + 8q_{n-1} + 4q_{n+1} + q_{n+2}), \qquad \bar{v}_n = \frac{1}{12h}(q_{n-2} - 8q_{n-1} + 8q_{n+1} - q_{n+2}).$$

Higher order: use coefficients δ_j ($j = -k, \ldots, k$), $\delta_j = -\delta_{-j}$, for a derivative approximation of order $2k$,

$$\bar{v}_n = \frac{1}{h}\sum_{j=-k}^{k}\delta_j q_{n-j}, \quad \bar{q}_n = (1-\delta_1)q_{n+1} - \sum_{j=-k, j\neq 0,1}^{k}\delta_j q_{n-j}, \quad \delta_j = \frac{(-1)^{j-1}}{j}\frac{k!^2}{(k-j)!(k+j)!},$$

for $j = 1, \ldots, k$ and where $\delta_0 = 0$.

- E. Hairer and C. Lubich, *Symmetric multistep methods over long times.* Numer. Math., 2004
- E. Hairer and C. Lubich, *Symmetric multistep methods for charged particle dynamics.* SMAI J. Comput. Math., 2017.

Related work: Lagrangian NN and Generalized Lagrangian NN

- Formulate their learning problem starting from the **(Forced) Euler-Lagrange Equations**:

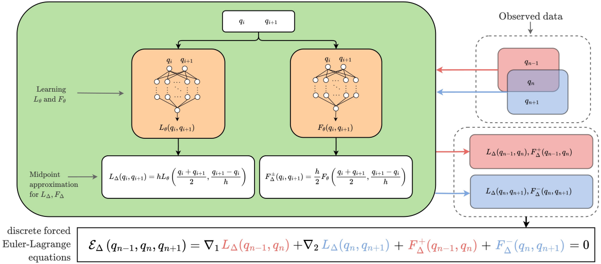
$$0 = \frac{\partial L}{\partial q}\left(q(t), \dot{q}(t)\right) - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\left(q(t), \dot{q}(t)\right)\right) + F\left(q(t), \dot{q}(t)\right).$$

- By expanding $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\left(q(t), \dot{q}(t)\right)\right)$ and **assuming** $\frac{\partial^2 L}{\partial \dot{q}^2}$ **invertible** one obtains

$$\ddot{q} = \left(\frac{\partial^2 L}{\partial \dot{q}^2}\right)^{-1}\left(\frac{\partial L}{\partial q} - \dot{q} \cdot \frac{\partial^2 L}{\partial q \partial \dot{q}} + F\right)$$

  Then solve this equation numerically to obtain $\{\hat{q}_n, \dot{\hat{q}}_n\}_{n=0}^N$ to compare with the observed data $\{q_n, \dot{q}_n\}_{n=0}^N$ in a least squares loss function.
- Learn both L_θ and F_θ.
- Need both position and velocity data (or approximate velocity with finite differences).
- Experiments on synthetic data.

- Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. *Lagrangian neural networks*, arXiv:2003.04630.
- Xiao, S., Zhang, J. and Tang, Y. *Generalized Lagrangian Neural Networks.* arXiv:2401.03728. Jan. 2024. http://arxiv.org/abs/2401.03728 (June 26, 2024).



Extrapolation error: is the error at the k-th step of a trajectory computed numerically, with $k \geq 2$ after L_θ and F_θ are learned.

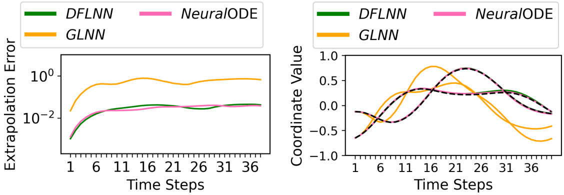
$$\text{Extrapolation Error}_k = \frac{1}{N_\mathcal{T}} \sum_{i=0}^{N_\mathcal{T}} \| q_k^{\{i\}} - \hat{q}_k^{\{i\}} \|_2^2 . \tag{1}$$

Mean square error associated to the k-th step of a predicted trajectory:

- q_k^i true trajectory at time step k given the initial condition for $\{q_0^i, q_1^i\}$;
- $\hat{q}_k^{\{i\}}$ is the solution of $\mathcal{E}_\Delta(\hat{q}_{k-2}^i, \hat{q}_{k-1}^i, \hat{q}_k^i) = 0$ for trajectory i, starting from $\mathcal{E}_\Delta(q_0^i, q_1^i, \hat{q}_2^i) = 0$.

Where $N_\mathcal{T}$ is the number of trajectories of N steps (segments of triplets $\{q_{n-1}, q_n, q_{n+1}\}$ are selected from each trajectory and used as input to the NNs during training).

Experiments (synthetic data): damped double pendulum



The model was trained on the damped dataset and evaluated on a test dataset with same damping.

Left: Extrapolation error given time step $h = C \cdot 10^{-2}$ and a training dataset containing 320 trajectories. Test dataset contains 10 trajectories.

Right: Prediction rollouts.

Experiments (synthetic data): damped double pendulum

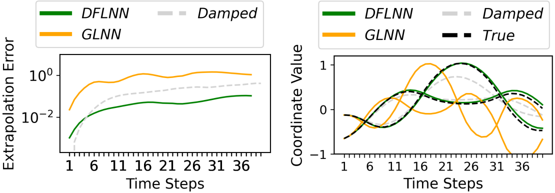
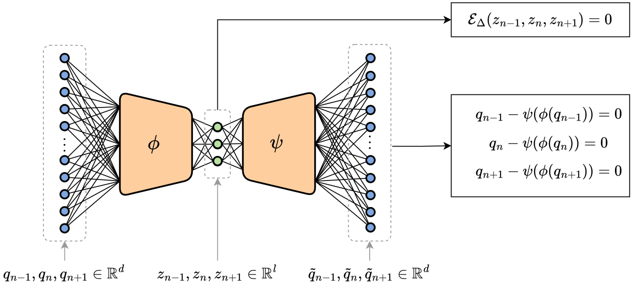


Figure: Evaluating the conservative part of the trained Lagrangian-based models on an undamped double pendulum dataset. The lightgray line is included as a reference to the damped system that the model is being trained on.

Dimensionality reduction for pixel data and human motion data

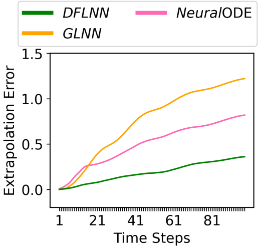


An autoencoder is applied to reduce the high-dimensional input dimension to a lower-dimensional latent space. The proposed model is applied to the dynamics in the latent space.

Experiments (synthetic data): pixel pendulum

Experiment inspired by:

- L. Mars Gao and J. Nathan Kutz, *Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants*, Prooceedings of the Royal Society A, 2024.



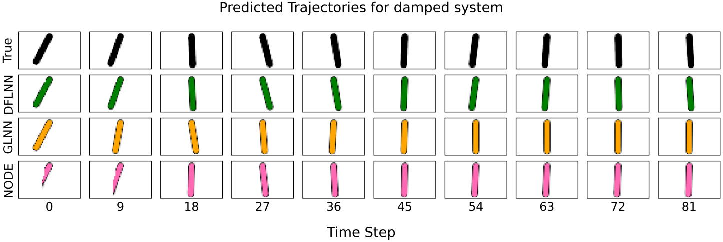
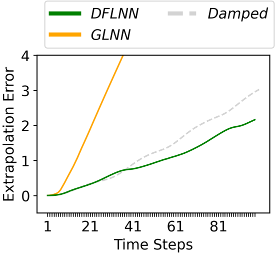


Figure: Results for a simple pendulum represented through pixel images. The models are trained on a damped dataset, and evaluated on a test dataset with the same damping.

Experiments (synthetic data): pixel pendulum



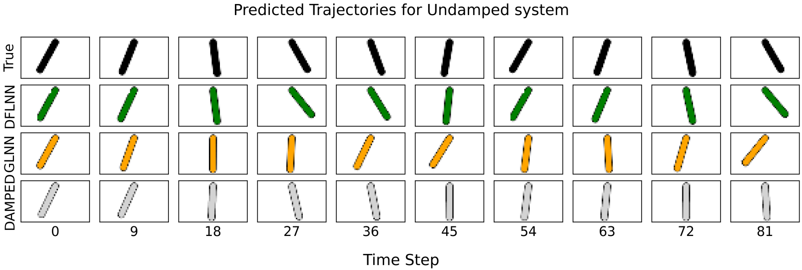
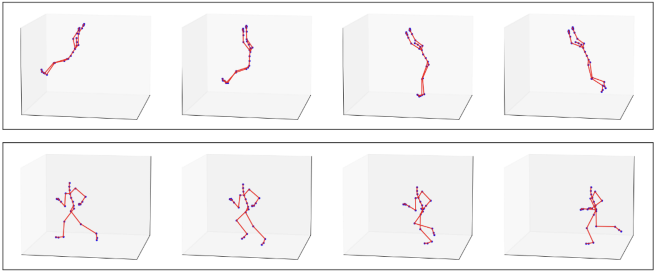
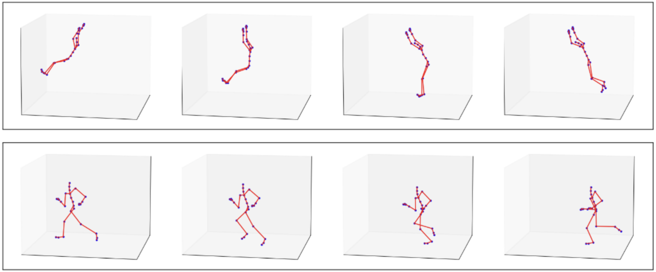


Figure: Results for a simple pendulum represented through pixel images. The models are trained on a damped dataset, the conservative part of the trained Lagrangian-based models is evaluated on an undamped simple pendulum system. The lightgray line and figures are included as a reference to the damped system that the model has been trained on.

Experiments (real-world data)

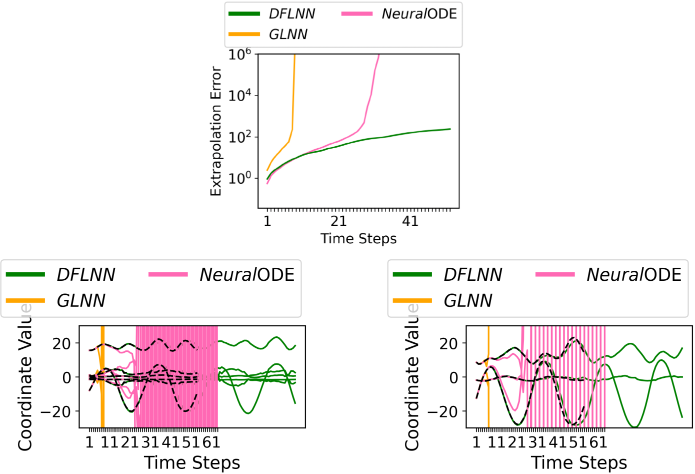


Experiments (real-world data)



- Too few trajectories for the same motion: need to augment the data-set.
- Use two trials of the same motion for the same person.
- Extract 10 trajectories per trial for a total of 20 trajectories.
- We track the motion of 10 joints: `rtibia`, `rfemur`, `rhipjoint`, `root`, `lowerback`, `upperback`, `thorax`, `rclavicle`, `rhumerus`, and `rradius`.
- Use a Savitzky-Golay filter to smoothen the data.
- Perform dimensionality reduction including the parameters of an autoencoder in the training.

Experiments (real-life data): swinging motion



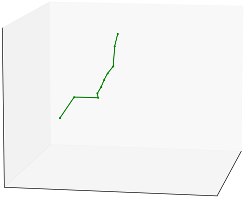
Reconstruction of human motion, capturing a swinging motion. (Black - recorded motion)

Top Extrapolation error.

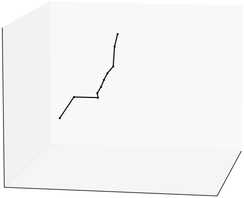
Left Prediction rollouts in coordinate space for the root of the skeleton.

Right Prediction rollouts for the the right femur.

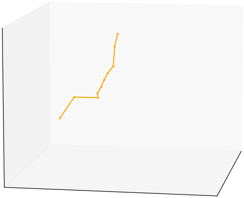
Animated skeletal movement

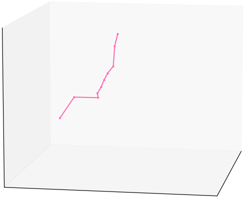


0 / 112

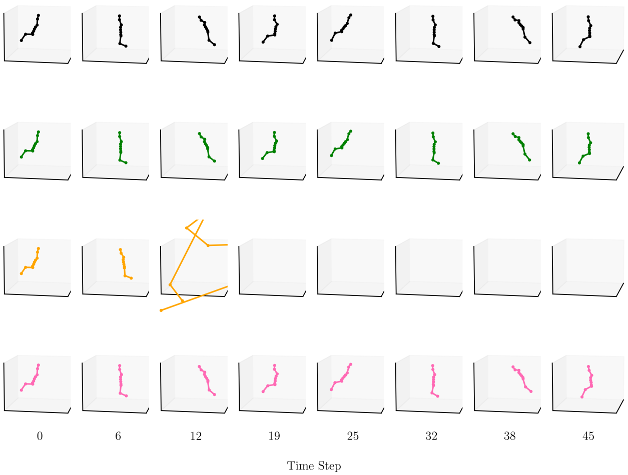


0 / 59





Skeletal movement snapshots



Green notation is DFLNN (proposed), yellow is the GLNN model, and pink lines a Neural ODE. The ground truth is indicated in black. Full movement represented as a skeletal sketch.

References

- J E Marsden and M West, *Discrete variational integrators*, Acta Numerica, (2001);
- DM de Diego, R. Sato Martin de Almagro, *Variational order for forced Lagrangian systems*, Nonlinearity, 2018.
- S Ober-Blöbaum, C. Offen, *Variational Learning of Euler-Lagrange Dynamics from Data*. Journal of Computational and Applied Mathematics, 421, 2023.
- C. Offen, S. Ober-Blöbaum, *Symplectic integration of learned Hamiltonian systems*, Chaos, 2022.
- C. Offen, S. Ober-Blöbaum, *Learning discrete Lagrangians for variational PDEs from data and detection of travelling waves*, 2023, arXiv preprint arXiv:2302.08232.
- E. Hairer and C. Lubich, *Symmetric multistep methods over long times.* Numer. Math., 2004
- E. Hairer and C. Lubich, *Symmetric multistep methods for charged particle dynamics.* SMAI J. Comput. Math., 2017.
- Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. *Lagrangian neural networks*, arXiv:2003.04630.
- Xiao, S., Zhang, J. and Tang, Y. *Generalized Lagrangian Neural Networks.* arXiv:2401.03728. Jan. 2024. http://arxiv.org/abs/2401.03728 (June 26, 2024).
- Aleksandar Botev, Andrew Jaegle, Peter Wirnsberger, Daniel Hennes, and Irina Higgins. Which priors matter? Benchmarking models for learning latent dynamics. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1), 2021.
- L. Mars Gao and J. Nathan Kutz, *Bayesian autoencoders for data-driven discovery of coordinates, governing equations and fundamental constants*, Prooceedings of the Royal Society A, 2024.
- Stölzle, M., and Della Santina, C. (2024). Input-to-state stable coupled oscillator networks for closed-form model-based control in latent space. Advances in Neural Information Processing Systems, 37, 82010-82059.
- Yaofeng Desmond Zhong, Biswadip Dey, Amit Chakraborty, Dissipative SymODEN: Encoding Hamiltonian Dynamics with Dissipation and Control into Deep Learning, arXiv:2002.08860.
- Yuji Okamoto, Ryosuke Kojima, *Learning Deep Dissipative Dynamics*, arXiv:2408.11479
- Domenico Campolo, Jianyu Hu, Juan-Pablo Ortega, and Daiying Yin *A kernel-based global method for the learning of elastic potentials on Lie groups*, Proceedings of GSI 2025.

Thank you for listening!

Temporary page!

LATEX was unable to guess the total number of pages correctly. A
there was some unprocessed data that should have been added t
the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page
will go away, because LATEX now knows how many pages to expe
for this document.