Forecasting Continuous Non-Conservative Dynamical Systems in SO(3)

Lennart Bastian[1,2*] Mohammad Rashed[1,2*] Nassir Navab[1,2] Tolga Birdal[3]

[1] Technical University of Munich [2] Munich Center of Machine Learning [3] Imperial College London

Abstract

Modeling the rotation of moving objects is a fundamental task in computer vision, yet $SO(3)$ extrapolation still presents numerous challenges: (1) unknown quantities such as the moment of inertia complicate dynamics, (2) the presence of external forces and torques can lead to non-conservative kinematics, and (3) estimating evolving state trajectories under sparse, noisy observations requires robustness. We propose modeling trajectories of noisy pose estimates on the manifold of 3D rotations in a physically and geometrically meaningful way by leveraging Neural Controlled Differential Equations guided with $SO(3)$ Savitzky-Golay paths. Existing extrapolation methods often rely on energy conservation or constant velocity assumptions, limiting their applicability in real-world scenarios involving non-conservative forces. In contrast, our approach is agnostic to energy and momentum conservation while being robust to input noise, making it applicable to complex, non-inertial systems. Our approach is easily integrated as a module in existing pipelines and generalizes well to trajectories with unknown physical parameters. By learning to approximate object dynamics from noisy states during training, our model attains robust extrapolation capabilities in simulation and various real-world settings. Code is available here.

1. Introduction

Modeling the dynamics of rigid bodies in 3D is fundamental to numerous domains such as tracking, robotic control, trajectory planning, and sensor fusion [9, 12, 14, 41]. While the mathematical foundations of rigid body mechanics are well understood, many modern applications require predicting rotational motion from noisy sensor data when key physical parameters are unknown. This presents significant challenges at the intersection of data-driven and physical modeling.

The core difficulty stems from the inherent nonlinearity of rotational motion in $\mathrm{SO}(3)$, where even simple rigid bodies can exhibit complex behaviors ranging from regular precession[2] to chaos [1]. Unlike translational motion in Euclidean space, rotational dynamics are governed by the moment of inertia tensor - a quantity that couples the body's mass distribution with its angular momentum in non-trivial ways. When this inertial property is unknown, direct estimation becomes an ill-posed inverse problem, particularly in the presence of external torques and measurement noise [18, 41, 46]. This problem is further complicated by the system's sensitivity to initial conditions - a characteristic of chaotic behavior that emerges in many rotating rigid body systems [1].

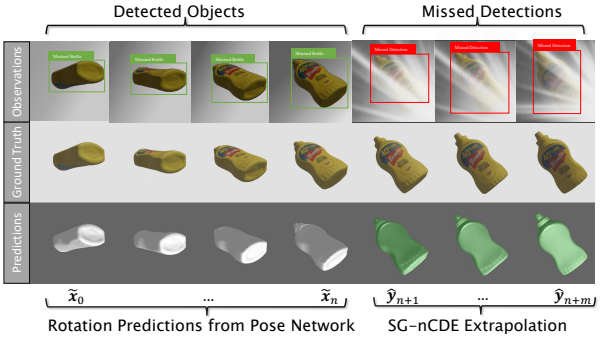

Figure 1. We propose Savitzky-Golay Neural Controlled Differential Equations (SG-nCDEs) for $\mathrm{SO}(3)$ forecasting, a fundamental task at the intersection of tracking and object pose estimation. Real-world rotational observations can be occluded or otherwise errant. Above, pose predictions (bottom left) are input into an SG-nCDE, which robustly compensates for missed detections of a rotational object trajectory (green predictions, bottom right).

This work proposes simultaneously addressing $\mathrm{SO}(3)$ state estimation and learning unknown physical parameters from noisy measurements. In contrast to previous approaches that restrict themselves to conservative systems, our method can predict rotational object trajectories in various physical scenarios, such as dissipative forces, or for objects under the influence of an external conservative force field. This versatility opens the door to a variety of applications, from improving the situational awareness of autonomous systems [13, 41], compensating for missing sensor measurements in tracking and localization [2, 14], to dynamic grasping [40, 63].

*Equal Contribution

✉ : lennart.bastian (at) tum.de

[2] consider a spinning top balancing due to the gyroscopic effect.

However, forecasting rotation presents several challenges: learned methods neglecting the geometrical nuances of $\mathrm{SO}(3)$ typically struggle due to numerical errors or non-linearities [3, 6, 15, 65]. Analogously, naive learning often fails to respect fundamental physical invariants motivating physics-guided machine learning (PGML) where explicit physical priors are incorporated into neural architectures [7, 10, 57]. An effective inductive bias should address both paradigms, learning the underlying physical laws and respecting the manifold geometry of $\mathrm{SO}(3)$.

To this end, we contribute the following:

- We propose to model the kinematic evolution of rotating bodies via a latent dynamical system, posing neural controlled differential equations (neural CDEs) as an appealing modeling paradigm for this setting.
- We design an integration control signal on the Lie group $\mathrm{SO}(3)$ to simultaneously handle denoising input states and serve as a numerical integration path for a CDE; our model learns a solution to the underlying physics based on a path in $\mathrm{SO}(3)$. This path is demonstrably suitable for controlling a neural CDE.
- Extensive experimental validation confirms our hypothesis that we can approximate a dynamical system with unknown inertial properties and external forces; the proposed method surpasses existing methods in both extrapolation capabilities and robustness to noise. We demonstrate the practicality of our method in simulation and various real-world object-tracking settings.

2. Related Works

We first review the relevant literature on $\mathrm{SO}(3)$ filtering, representation learning, and PGML.

SO(3) Filtering. Robustly filtering measurements from visual or inertial sensors is essential for many applications from video stabilization [24], continuous-time state estimation in robotics [54, 55, 62], tracking tasks [14], sensor fusion [4, 44], and SLAM [2, 21, 38]. Filtering approaches using splines [19, 30, 38] must be constructed to preserve manifold properties of $\mathrm{SO}(3)$ to avoid accruing numerical errors [4, 58]. However, $\mathrm{SO}(3)$ spline derivatives cannot be solved in closed form, making their integration into robust interpolation schemes computationally intensive [53]. Without optimization, splines are unsuitable for extrapolation as they diverge near interpolation boundaries [49]; many online methods requiring state prediction (e.g., ORB-SLAM [43]) therefore rely on simplified constant-velocity assumptions.

Savitzky-Golay filtering interpolates $\mathrm{SO}(3)$ more efficiently by regressing a polynomial in the tangent space and mapping back to the manifold via the exponential map [25]. This approach allows for simultaneous angular velocity and acceleration estimation solvable in a linear system, providing a robust method for processing noisy rotation data in $\mathrm{SO}(3)$, recently exemplified for hand tracking [14]. However, extrapolation of rotations presents challenges beyond interpolation and is particularly sensitive to error accumulation. Existing methods for extrapolation on $\mathrm{SO}(3)$ have considered Gaussian Processes (GPs) for their ability to model uncertainty and handle continuous-time data [16, 32, 33, 36], or regress incremental changes in $\mathrm{SE}(3)$ directly from point cloud sequences [5]. However, they do not consider physical laws governing the underlying dynamics.

Learning Representations in SO(3). The literature on learning 3D rotations has explored various representations, each with distinct trade-offs. While classical approaches like Euler angles [31], axis-angle, or quaternions [20, 64] are intuitive, they can suffer from singularities (e.g., gimbal lock) or have topological constraints that make them challenging to regress directly [3, 15, 65] and backpropagate through [6, 56]. This is fundamentally due to $\mathrm{SO}(3)$ not being homeomorphic to any subset of 4D Euclidean space [65]. Recent work has demonstrated that these limitations impede deep learning performance [15, 35, 65], motivating the development of higher-dimensional continuous representations that enable stable optimization by mapping $\mathrm{SO}(3)$ to an overparameterized continuous space. Notable approaches include 6D [65], 9D [35], and 10D [48] representations.

Physics-Guided Extrapolation. Various paradigms have been devised to instill data-driven models with priors from known physical laws [61]. Among these are Physics-Informed Neural Networks [50], which construct regularizers to make network predictions physically plausible. Another line of work seeks to learn the Hamiltonian [57] or Lagrangian [10], ensuring predictions adhere to conservation laws like energy and momentum. While real-world settings require models to disentangle dynamics from sensor noise, these works focus on learning the physical dynamics from ground truth observations.

On the other hand, Neural Ordinary Differential Equations (neural ODEs) explicitly solve an initial value problem (IVP) parameterized by a latent representation [7]. Several works such as GRU- and RNN-ODEs [11, 51] have been developed to incorporate additional measurements beyond the initial value more effectively, conditioning neural ODEs with accumulated representations, making them more suitable for time series modeling. Neural controlled differential equations (Neural CDEs) [29] achieve this by bridging controlled differential equations with deep neural networks. Instead of merely solving an IVP, the hidden state is integrated with the derivative of a control signal.

While Hamiltonian formulations have been used for state prediction from noisy observations in $\mathrm{SO}(3)$ [41] and $\mathrm{SE}(3)$ [12], adapting these for non-conservative systems proves cumbersome. [41] introduce a learning-based approach for predicting 3D rotational dynamics from images of rigid bodies with unknown mass distribution. They design

a network that learns a latent representation homeomorphic to $\mathrm{SO}(3)$ directly from images to extrapolate an object trajectory. However, this approach has limitations when modeling noisy signals as it is sensitive to the initial momentum estimation and, by design, cannot handle non-conservative systems. Furthermore, this confounds object pose estimation with learned dynamics, requiring many annotated image sequences to capture complex dynamics. Instead, we observe that $\mathrm{SO}(3)$ dynamics models can be trained in simulation and later coupled with specialized pose detectors [8, 27, 45], generalizing to noisy real-world dynamics.

3. Preliminaries

Rigid-body Dynamics. While translation and rotation of a rigid body in $\mathbb{R}^3$ each have three degrees of freedom, the space of rotations forms a non-commutative Lie group. This makes modeling an evolving trajectory more challenging than the Euclidean translational counterpart.

Definition 1 (Special Orthogonal Group)**.** *The special orthogonal group* $\mathrm{SO}(3)$ *represents the set of all rotations about the origin in* $\mathbb{R}^3$*:*

$$\mathrm{SO}(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\right\} \tag{1}$$

Modeling kinematics also requires derivatives in $\mathrm{SO}(3)$. The Lie algebra $\mathfrak{so}(3)$, the tangent space at the identity of $\mathrm{SO}(3)$, consists of real 3×3 skew-symmetric matrices.

Definition 2 (Hat Operator)**.** *Any* $\boldsymbol{\omega} \in \mathbb{R}^3$ *can be associated with an element* $\hat{\boldsymbol{\omega}} \in \mathfrak{so}(3)$ *via the hat operator:*

$$\hat{\boldsymbol{\omega}} = S(\boldsymbol{\omega}) = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{2}$$

where the hat operator $\hat{\boldsymbol{\omega}} : \mathbb{R}^3 \to \mathfrak{so}(3)$ *maps vectors to skew-symmetric matrices.*

The time evolution of the rotation matrix $R(t)$ is given by $\dot{R} = R\hat{\boldsymbol{\omega}}$ for the angular velocity $\boldsymbol{\omega}$ [25]. Here, a differential geometry perspective is useful: $\mathfrak{so}(3)$ is isomorphic to the tangent space $T_{\mathbf{R}}\mathrm{SO}(3)$ at $\mathbf{R}$ [34]; we characterize rotation trajectories using these geometric definitions.

4. $SO(3)$ Savitzky-Golay Neural-CDEs

We proceed by describing a method designed to extrapolate noisy rotation estimates of a dynamical system where the moment of inertia and external forces are unknown. The core of the method consists of a neural CDE, which is integrated with respect to a control signal on the manifold $\mathrm{SO}(3)$. We construct this control signal using an $\mathrm{SO}(3)$ Savitzky-Golay filter [25], providing a robust de-noised prior for extrapolating the learned latent dynamical system. An overview can be seen in Fig. 2.

4.1. $SO(3)$ Neural CDEs

Definition 3 (Discrete Trajectories in $\mathrm{SO}(3)$)**.** *Let* $\mathcal{T} = \{(t_k, \tilde{\mathbf{x}}_k)\}_{k=0}^N$ *be a sequence of observations of an object* $\mathbf{x}$ *at discrete time points* t_k*, where* $\tilde{\mathbf{x}}_k \in \mathrm{SO}(3)$ *represents a noisy rotational measurement at time* t_k*:*

$$\mathcal{T} = \{(t_0, \tilde{\mathbf{x}}_0), (t_1, \tilde{\mathbf{x}}_1), \ldots, (t_N, \tilde{\mathbf{x}}_N)\} \subset \mathbb{R} \times \mathrm{SO}(3)$$

Such a sequence could come from a rotating (non-zero angular velocity) object trajectory obtained directly from images through a 6D pose estimator [59] or in combination with other sensors. We seek to estimate the underlying dynamics of the object in a data-driven manner to extrapolate unseen rotations for $\hat{t} > t_N$.

Definition 4 (Neural CDE [29])**.** *Let* $X : [t_0, t_n] \to \mathbb{R}^{v+1}$ *be a control path, where* $X_{t_i} = (t_i, \mathbf{x}_i)$*. A Neural CDE is defined as:*

$$\mathbf{z}_t = \mathbf{z}_{t_0} + \int_{t_0}^{t} f_\theta(\mathbf{z}_s)\mathrm{d}X_s \quad \textit{for } t \in (t_0, t_n] \tag{3}$$

where:

- $\mathbf{z}_t \in \mathbb{R}^w$ *is the hidden state at time* t*,*
- $f_\theta : \mathbb{R}^w \to \mathbb{R}^{w\times(v+1)}$ *is a neural network with parameters* θ*,*
- $\mathbf{z}_{t_0} = \zeta_\theta(t_0, \mathbf{x}_0)$ *is the initial condition, where* $\zeta_\theta : \mathbb{R}^{v+1} \to \mathbb{R}^w$ *is a network to encode the initial condition* $(t_0, \mathbf{x}_0)$*.*

Lemma 5 (Constructing Neural CDEs)**.** *The solution* $\mathbf{z}$ *to Eq. (3) is the response of a Neural CDE controlled by the path* X*. The control path* X *can be constructed arbitrarily as long as it is continuous and differentiable.*

Kidger et al. [29] construct the path $X \in \mathcal{C}^2$ as a natural cubic spline over $\mathcal{T}$, while [42] use Hermite splines for their local support. Others have proposed learning integration paths [23]. We observe that these choices neither obey the geometric structure of $\mathrm{SO}(3)$, nor are they robust to sensor noise (see App. B.3 for details).

4.2. Robust $SO(3)$ Savitzky-Golay Regression

To address sensor noise from real-world trajectories, we propose constructing an integration path with a robust regression signal to integrate Eq. (3). However, robust regression on $\mathrm{SO}(3)$ is non-trivial due to the non-Euclidean nature of the Lie group. Our control signal should avoid expensive optimizations or polynomial basis functions unsuited for extrapolation [53]. Savitzky-Golay filters achieve this by constructing a polynomial with global support in the Lie algebra, solvable as a single least squares problem [25].

Definition 6 (Lie Algebra Polynomial)**.** *We define a second-order polynomial in the Lie algebra* $\mathfrak{so}(3)$ *as:*

$$p(t; \boldsymbol{\rho}) := \boldsymbol{\rho}_0 + \boldsymbol{\rho}_1 t + \frac{1}{2}\boldsymbol{\rho}_2 t^2 \in \mathfrak{so}(3) \tag{4}$$

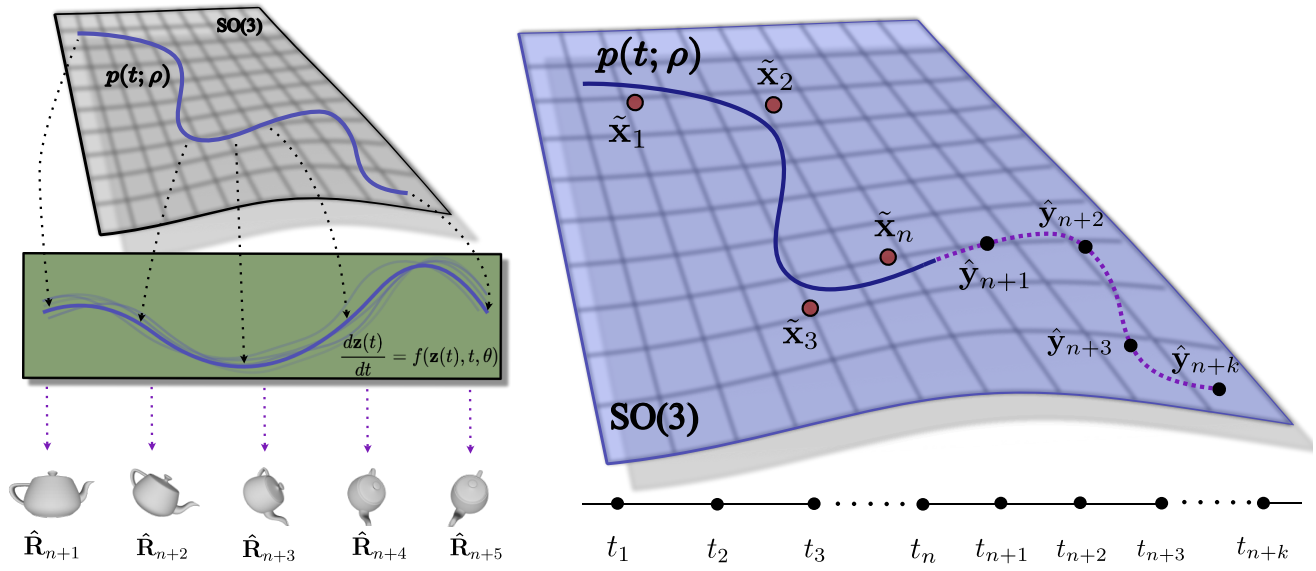

Figure 2. We consider the problem of forecasting the rotational motion of a rigid body with unknown physical properties from noisy sensor estimates $\tilde{\mathbf{x}}_k$ in $\mathrm{SO}(3)$. The core of our proposed method is a neural controlled differential equation, which learns a latent representation of the underlying dynamical system with respect to a control path $\mathbf{p}(t;\cdot)$ defined by coefficients ρ (left). We construct this control path directly on the manifold $\mathrm{SO}(3)$ by filtering noisy input observations with a Savitzky-Golay filter (right). This allows the CDE to learn a more robust parameterization for extrapolating rigid-body trajectory estimates $\hat{\mathbf{y}}_k$ than existing methods.

where $\boldsymbol{\rho} = [\boldsymbol{\rho}_0; \boldsymbol{\rho}_1; \boldsymbol{\rho}_2] \in \mathbb{R}^9$ represents the vector of polynomial coefficients.

Definition 7 (Control Path on $\mathrm{SO}(3)$)**.** *Given the sequence of noisy observations $\mathcal{T}$ from Def. (3), we define a smooth control path $\varphi(t) \in \mathrm{SO}(3)$ as:*

$$\varphi(t) = \mathrm{Exp}(p(t - t_k; \boldsymbol{\rho}_k))\tilde{\mathbf{x}}_k \tag{5}$$

where t_k is the anchor point time, $\tilde{\mathbf{x}}_k \in \mathrm{SO}(3)$ is the corresponding noisy measurement, $\mathrm{Exp} : \mathfrak{so}(3) \to \mathrm{SO}(3)$ is the exponential map, and $\boldsymbol{\rho}_k$ represents the polynomial coefficients specific to time t_k.

The control path $\varphi(t)$ provides a geometric solution to the path construction problem in Lem. (5). By designing $\varphi(t)$ to be smooth and respect the structure of $\mathrm{SO}(3)$, we can construct a more appropriate control signal for the Neural CDE than traditional cubic or Hermite splines.

Theorem 8 (Savitzky-Golay Filtering on $\mathrm{SO}(3)$ [25])**.** *The optimal polynomial coefficients $\boldsymbol{\rho}_k$ at time t_k can be obtained by solving the following least-squares problem:*

$$\boldsymbol{\rho}_k = \arg\min_{\boldsymbol{\rho}} \sum_{m=-n}^{n} \| (\mathrm{Log}(\tilde{\mathbf{x}}_{k+m}\tilde{\mathbf{x}}_k^{-1}) - p(t_{k+m} - t_k; \boldsymbol{\rho})) \|^2 \tag{6}$$

where $2n+1$ is the window size, and $\mathrm{Log} : \mathrm{SO}(3) \to \mathfrak{so}(3)$ maps the rotation difference to its corresponding element in the Lie algebra.

The optimization problem in Eq. (6) reduces to:

$$\boldsymbol{\rho}_k = \arg\min_{\boldsymbol{\rho}} \|\mathbf{A}\boldsymbol{\rho} - \mathbf{b}\|^2 \tag{7}$$

with solution:

$$\boldsymbol{\rho}_k = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b} \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{3(2n+1)\times 9}$ and $\mathbf{b} \in \mathbb{R}^{3(2n+1)}$ are constructed from time and rotation differences, respectively.

The constructions of $\mathbf{A}$ and $\mathbf{b}$ are detailed in App. C.2.

The filter described in Def. (7) can be used to numerically integrate the Neural CDE in Eq. (3) directly. However, Savitzky-Golay filters exhibit artifacts near the boundary t_N, which can hinder extrapolating the learned system dynamics. Therefore, we propose weighting the Savitzky-Golay trajectory with a weight matrix as proposed for the Euclidean analog [52]. Unlike existing works, we construct this out of learnable parameters $\mathbf{W}$, leading to the following weighted formulation:

Lemma 9 (Weighted Savitzky-Golay Optimization)**.** *The polynomial coefficients of the weighted Savitzky-Golay filter can be computed in closed form as:*

$$\boldsymbol{\rho} = (\mathbf{A}^\top \mathbf{W}\mathbf{A})^{-1}\mathbf{A}^\top \mathbf{W}\mathbf{b} \tag{9}$$

where $\mathbf{W}$ consists of learnable parameters that can be updated through stochastic gradient descent.

Our model learns to appropriately weight this denoising signal in a manner well-suited for extrapolation while evolving the dynamical system in Eq. (3).

4.3. $SO(3)$ Savitzky-Golay Neural-CDEs

Control paths for Neural CDEs are discussed extensively in [42]; the authors show that a desirable control function should be sufficiently smooth and bounded to attain the universal approximation properties of Neural CDEs.

We show that $SO(3)$ Savitzky-Golay filters admit these properties in what follows. To mitigate discontinuities and non-bounded terms in the piece-wise polynomial regression, we regress $\varphi(t)$ by defining a window over the last $2n+1$ samples of $\{\tilde{\mathbf{x}}_k\}_{k=0}^{N}$. We can then observe that this global Savitzky-Golay filter attains the desirable properties of a control path for Eq. (3) in the following informal result, as long as the rotational difference is bounded within this interval. This last assumption is required to avoid discontinuities of $\mathrm{Log} : \mathrm{SO}(3) \to \mathfrak{so}(3)$ at π, which we can assure with a sufficiently high sampling rate or small enough window size.

Proposition 10. *Let $\varphi(t)$ be the control path constructed in Def. (7). If the maximum rotational difference $\delta_n = \max_k \|\mathrm{Log}(\tilde{\mathbf{x}}_k \tilde{\mathbf{x}}_0^{-1})\|$ within the window is bounded, then:*
1. $\varphi(t)$ is analytic and twice differentiable,
2. The derivatives $\varphi'(t)$ and $\varphi''(t)$ are bounded,
3. $\varphi(t)$ uniquely minimizes the problem in Thm. (8).

Proof sketch. As $\varphi(t)$ is defined as a composition of polynomial and analytic functions on $\mathrm{SO}(3)$ (see Def. (7)), we observe that the first and second derivatives also take this form. Boundedness can be shown by analyzing the construction of the polynomial coefficients $\boldsymbol{\rho}$ with respect to δ_n and considering the derivatives of $\varphi(t)$. Uniqueness follows from strict convexity of the optimization problem in Thm. (8). □

We treat this thoroughly in App. D. This result assures that the approximated evolving trajectory $\varphi(t)$ is well behaved as a control path for Eq. (3). In addition to fulfilling the properties of a suitable control path for a CDE, this approach is more robust for extrapolation than hermite splines and allows more stable numerical integration (see Sec. 5.1).

4.4. Higher-order Control Paths

Besides the encoded latent initial condition z_{t_0} and the first-order derivative $\mathrm{d}X$, our model has no input on how angular velocity changes over time. This can cause temporal phase shifts where trajectories with the correct predicted orientation lag behind ground truth trajectories.

Lemma 11 (Second-Order Neural CDE)**.** *We propose a second-order Neural CDE that incorporates both the first and second derivatives of the control path:*

$$\mathbf{z}_t = \mathbf{z}_{t_0} + \int_{t_0}^{t} f_\theta(\mathbf{z}_s)\mathrm{d}X_s + g_{\theta'}(\mathbf{z}_s)\mathrm{d}^2X_s \qquad (10)$$

for $t \in (t_0, t_n]$ where f_θ and $g_{\theta'}$ are neural networks with parameters θ and θ', and $\mathbf{z}_{t_0}$ is the initial condition.

The second-order term $g_{\theta'}(\mathbf{z}_s)\mathrm{d}^2X_s$ provides explicit acceleration information to the neural ODE, enabling more accurate modeling of systems with non-constant angular velocity. The Savitzky-Golay filter on $\mathrm{SO}(3)$ produces both $\mathrm{d}X$ and d^2X in closed form, making it particularly well-suited for our proposed Neural CDE formulations.

4.5. Learning Rotational Kinematics

The literature supports using 9D rotation matrices as a latent representation for learning on $\mathrm{SO}(3)$ [3, 15, 65]. We therefore use the 9D rotational derivatives $\dot{R} = R\hat{\boldsymbol{\omega}}$ obtained from Sec. 4.2 directly as an integration path for the latent state in Eq. (3). Subsequently, the time-evolved hidden state of the Neural CDE is projected into the 6D representation [65], which has been demonstrated suitable as a learning target of a neural network:

$$\mathbf{r_k} = (\nu_1, \nu_2) \in \mathbb{R}^{3\times 2} \qquad (11)$$

The prediction $\mathbf{y_k} \in \mathrm{SO}(3)$ can then be recovered via Gram-Schmidt orthonormalization (GSO) [15]. We now optimize to minimize the geodesic distance to learn the latent code θ of the network $\mathbf{f}_\theta$ in Eq. (3):

$$\arg\min_\theta \sum_{k=1}^{m} ||\mathrm{Log}(\mathbf{y_k}\mathbf{x_k^{-1}})||_F \qquad (12)$$

where $\hat{\mathbf{y}}_\mathbf{k}$ is the rotation matrix reconstructed via GSO obtained by solving the Neural CDE at time t_k, $\mathbf{x_k}$ is the ground-truth observation, and $||\cdot||_F$ denotes the Frobenius norm. During inference, we proceed by evolving the differential equation in Eq. (3) forward in time for each future point $k \in [t_{N+1}, t_{N+m}]$ (see Fig. 2).

5. Emperical Evaluation

We conduct extensive experiments on simulated rotational trajectories to understand the behavior of rotation extrapolation methods. Furthermore, we use models trained in simulation for rotational forecasting on real-world data from the Oxford Motion Dataset (MOD) [26] and rotational estimates from a modern 6D-pose estimator [59] to demonstrate generalization to unseen dynamics and sensor noise.

$\mathrm{SO}(3)$ Kinematics. The motion of rotating rigid bodies follows well-established physical principles that can be derived from classical mechanics; using these, we can simulate diverse rotational trajectories to study $\mathrm{SO}(3)$ forecasting in a principled manner.

Definition 12 (Moment of Inertia Tensor)**.** *The moment of inertia (MOI) tensor $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is symmetric positive-definite characterizing a rigid body's resistance to rotational*

Extrapolation Horizon	Method	Freely Rotating	Linear Control	Velocity Damping	Configuration-Dependent Torque	Variable Dynamics
$t = 0.8$ *seconds*	LEAP ([23])	7.67 ± 4.76	3.98 ± 3.76	4.12 ± 1.91	5.45 ± 2.58	11.45 ± 7.67
	Conservational ([41])	2.63 ± 0.22	2.16 ± 0.03	2.14 ± 0.02	2.17 ± 0.03	2.72 ± 0.24
	SO(3)-GRU	1.65 ± 0.06	0.82 ± 0.03	0.78 ± 0.02	0.88 ± 0.02	1.81 ± 0.10
	SO(3)-nCDE	2.25 ± 0.18	1.11 ± 0.07	1.07 ± 0.05	1.23 ± 0.07	2.30 ± 0.22
	SG-nCDE (Ours)	$\mathbf{0.87 \pm 0.05}$	$\mathbf{0.49 \pm 0.03}$	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.58 \pm 0.03}$	$\mathbf{0.89 \pm 0.07}$
$t = 1.2$ *seconds*	LEAP ([23])	11.78 ± 1.56	6.51 ± 2.73	4.76 ± 0.79	7.28 ± 1.51	15.12 ± 5.95
	Conservational ([41])	4.09 ± 0.34	3.18 ± 0.05	3.14 ± 0.04	3.20 ± 0.05	4.28 ± 0.44
	SO(3)-GRU	2.67 ± 0.11	1.34 ± 0.05	1.12 ± 0.04	1.43 ± 0.05	2.71 ± 0.19
	SO(3)-nCDE	3.77 ± 0.26	1.76 ± 0.10	1.68 ± 0.09	1.83 ± 0.10	3.89 ± 0.42
	SG-nCDE (Ours)	$\mathbf{1.28 \pm 0.08}$	$\mathbf{0.64 \pm 0.03}$	$\mathbf{0.50 \pm 0.03}$	$\mathbf{0.74 \pm 0.04}$	$\mathbf{1.31 \pm 0.14}$

Table 1. Trajectory prediction accuracy on 0.8 and 1.2 second extrapolation in rotational error (degrees). We evaluate several recent methods for forecasting temporal dynamics on scenarios encompassing known and unknown moments of inertia and various external forces. **Best** and second-best results are emphasized.

motion. There exists a coordinate system (principal axes) where $\mathbf{J}$ *is diagonal [17] (see App. C.1).*

In real-world applications with dissipative forces and measurement noise, strict conservation laws may not always hold. We, therefore, simulate both conservational (as in [41]) and non-conservational ($\tau_{\text{ext}} \neq 0$) trajectories.

Definition 13 (Rotational Rigid Body Dynamics)**.** *The following equations govern rigid rotational dynamics in 3D:*

$$\dot{R} = R\hat{\omega}, \tag{13}$$

$$\dot{\omega} = \mathbf{J}^{-1}(\tau_{ext} - \omega \times \mathbf{J}\omega), \tag{14}$$

where:

- $R(t) \in \mathrm{SO}(3)$ *is the rotation matrix representing orientation,*
- $\omega(t) \in \mathbb{R}^3$ *is the body-frame angular velocity,*
- $\mathbf{J} \in \mathrm{R}^{3\times3}$ *is the (diagonal) inertia matrix,*
- $\tau_{ext} \in \mathbb{R}^3$ *represents external torques.*

Experimental Design. We evaluate our method on five simulated scenarios representing diverse, realistic dynamics by integrating the equations in Def. (13) forward in time with $dt = 10^{-3}$. To understand the behavior of non-conservative systems, we explicitly define different representations for external forces τ_{ext}:

- *Freely Rotating Body:* No external torques ($\tau_{\text{ext}} = 0$). Angular momentum is conserved.
- *Linear Torque Control:* A linear control law $\tau_{\text{ext}} = \mathbf{J}(\mathbf{A}\omega + \mathbf{b})$ where $\mathbf{A} \in \mathbb{R}^{3\times3}$.
- *Velocity Damping:* Direct damping as a function of angular velocity through $\tau_{\text{ext}} = \mathbf{J}(\mathbf{D}\omega)$ with negative definite damping matrix $\mathbf{D}$.
- *Combined External Torque:* Superposition of damped rotation and an external torque computed in the world frame, representing conservative force fields (e.g., magnetic, gravitational, or torsional forces) where the torque depends explicitly on orientation with contributions weighted by $\mathbf{J}$.
- *Variable Dynamics:* Trajectories sampled uniformly from the four scenarios above.

Data Simulation. We generate 8000 10-second trajectories for each simulation above, separated into four splits. Each split is simulated based on a distinct MOI distribution to assess model generalization. We use 2/1/1 splits for training, validation (model selection and hyperparameter tuning), and testing. During each training iteration, a model sees a 1.2 second trajectory segment sampled uniformly at $dt = 10^{-1}$ and is regularized on predictions for the next 0.8 seconds. Batches are sampled randomly from all trajectories and their segments within a given split. Additional details are provided in App. B. 1.

Evaluation Metrics. During evaluation, we measure the rotational geodesic error (RGE) [22] each prediction:

$$\mathrm{RGE}(R_1, R_2) = 2 \arcsin\left(\frac{\|R_2 - R_1\|_F}{2\sqrt{2}}\right) \tag{15}$$

Noise for the input sequences is sampled from $\mathcal{N}(0, \delta)$ with $\delta \in [0.01, 0.05]\pi$. 0.05π corresponds to a rotation of $\approx 9°$, which is already considerable noise (for instance, in 6D pose estimation $2°$ is used as a threshold [59]). Results for $\delta = 0.05\pi$ are reported in Tab. 1.

Baselines. We compare our method with the Conservation of Energy approach for SO(3) prediction [41], SO(3)-GRU and spline SO(3)-nCDE baselines we introduce, and the learnable path CDE (LEAP) [23] on the 10 scenarios described in Sec. 5. As the conservation of energy method [41] solves an initial value problem, we provide the additionally required angular momentum estimate to integrate forward in time from the $\tilde{\mathbf{x}}_n$. Momentum estimates $\tilde{\mathbf{L}}$ are sampled from a noise distribution proportional to the noisy state estimates $\tilde{\mathbf{x}}$. Details for baseline models can be found in App. B.2.

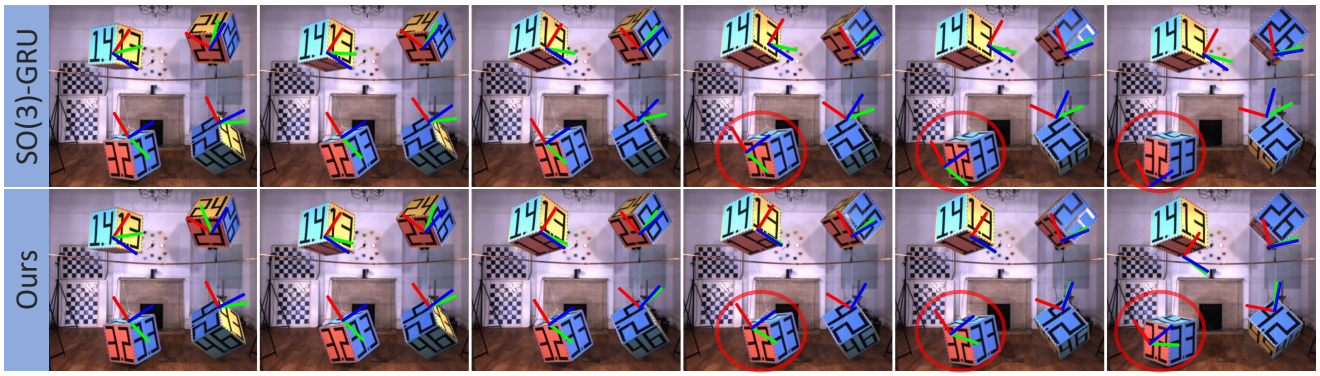

Figure 3. Qualitative comparison of the extrapolation results (left to right) produced by our method (top row) and the $\mathrm{SO}(3)$-GRU baseline (bottom row) on the OMD dataset. Methods are conditioned on the preceding timesteps (not visualized). Rotation is represented using 3D axis coordinates. Ground truth observations relative to the world coordinate system are used for the translation component.

Implementation Details. We use Python version 3.12, PyTorch [47] for data simulation, and diffrax [28] for numerical integration of our method using Dormand-Prince 4/5 to integrate the dynamical systems. Experiments are run on an RTX 5000 GPU. Adam is employed as an optimizer with a learning rate of 5×10^{-3}.

$\mathrm{SO}(3)$ Trajectory Visualizations. Quaternions offer a convenient visualization method as they can be reduced to three dimensions while preserving both the angle and axis of rotation. Let $\mathbf{q} = [cos(\theta/2), sin(\theta/2)\mathbf{u}]$ be a quaternion where θ is the rotation angle, and $\mathbf{u}$ is the unit vector representing the axis of rotation (the axis-angle representation). By normalizing the last three coordinates, we project trajectories onto the unit sphere $\mathbb{S}^2$. Visualizations of longer forecasts of 1.2 seconds (during training, the model only sees 0.8s) are depicted on $\mathbb{S}^2$ in Fig. 4 for the *velocity damping* scenario.

5.1. Analysis & Applications

The results for simulated experiments are depicted in Tab. 1. LEAP [23] generally does not successfully reconstruct or forecast $\mathrm{SO}(3)$ trajectories. Nevertheless, this motivates using a robust filtering approach for the CDE instead of learning the path in an unstructured way. On the other hand, the baseline $\mathrm{SO}(3)$ spline neural CDE predicts plausible trajectories, particularly within the training horizon of $t = 0.8s$. The $\mathrm{SO}(3)$-GRU performs competitively in several cases, particularly with dissipative forces. Our proposed method tends to perform the most consistently.

We observe the GRU and our method predictions are competitive beyond the training horizon (see Fig. 4). At the same time, the $\mathrm{SO}(3)$-nCDE baseline and LEAP tend to diverge sharply after several predicted timesteps. The conservation approach [41] predicts plausible trajectories, yet visualized trajectories in non-conservational cases confirm that external forces are not accounted for. In App. B.4 we show that even with ground truth momentum estimates, the trajectores diverge for non-conservational cases.

Sim2Real: Unseen Dynamics and Noise. We evaluate the capabilities of models trained on the *Variable Dynamics* scenario in Sec. 5.1 on trajectories in the Oxford Motion Dataset (OMD) [26] consisting of observations of four swinging boxes for three different camera motions (translational, unconstrained, and static). Vicon tracking data is downsampled to 40Hz, and models are evaluated on predicting 0.3 seconds. Statistics for results aggregated over the four boxes in each scene are depicted in Tab. 2. We observe that SG-nCDEs consistently outperform despite the diverse motions present.

Method	Translation Motion	Unconstrained Motion	Static Motion
$\mathrm{SO}(3)$-nCDE	6.49 ± 9.42	5.43 ± 7.09	7.82 ± 12.19
$\mathrm{SO}(3)$-GRU	3.04 ± 1.29	2.90 ± 1.25	2.83 ± 1.26
SG-nCDE (Ours)	$\mathbf{2.32 \pm 1.03}$	$\mathbf{2.30 \pm 1.02}$	$\mathbf{2.18 \pm 1.00}$

Table 2. Trajectory prediction error (degrees) across different motions in the OMD dataset [26]. We evaluate methods in 3 different scenarios (translational, unconstrained, and static). **Best** and second-best results are emphasized.

We additionally present a qualitative comparison of extrapolation performance in Fig. 3, highlighting the differences in extrapolation quality between our method and the $\mathrm{SO}(3)$-GRU baseline. As we represent translation with respect to the world coordinate system, errors accrued in rotation estimation manifest as offsets in pixel space. This is apparent later in the image sequence, where the predicted rotations from the $\mathrm{SO}(3)$-GRU baseline cause the projected coordinate system to drift significantly. The boxes have different translational velocities, with the two right-most boxes (#2 and #4) yielding the most stable overall predictions.

Dynamic Pose Estimation. To gauge the performance in conjunction with 6D object pose estimators, we apply our method to the outputs of the widely used GDR-NPP [37, 59]. We fine-tune the network weights published by the authors on a single object and apply the pose estimation to a sequence of (unseen) simulated images depicted in Fig. 1, with detailed comparisons in App. A.1. In this way, dynamics estimation can be decoupled from object pose estimation,

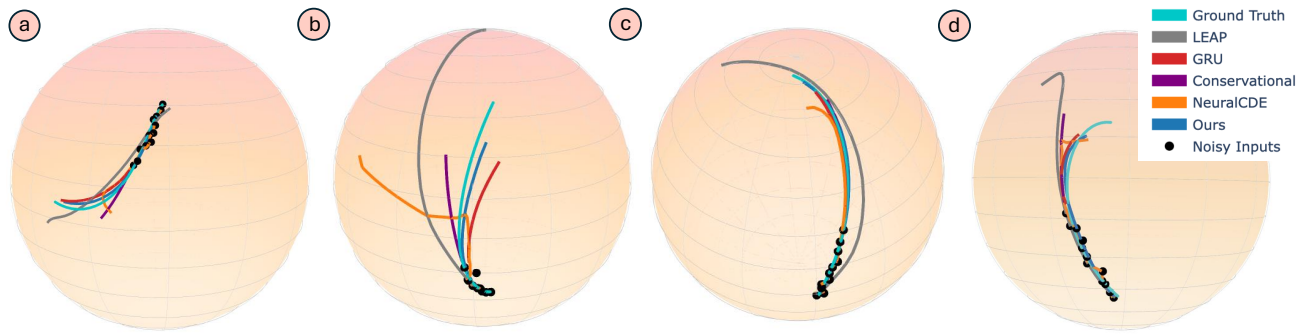

Figure 4. Trajectory extrapolations visualized on the unit sphere $\mathbb{S}^2$ for the proposed method (blue) and the relevant baselines on the *damped* scenario. We additionally depict the ground truth (teal) and the noisy state estimates (black) provided as input for reference. Trajectories are visualized for $t = 1.2s$ future timesteps.

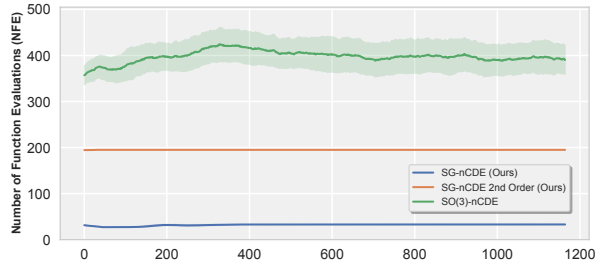

Figure 5. Comparison of the number of function evaluations (NFE) per training step between the baseline model and our model. The shaded areas represent one standard deviation.

mitigating the cost of annotating dynamic objects.

Ablation Study. We analyze the performance of the baseline $\mathrm{SO}(3)$-nCDE with our proposed modifications in Tab. 3 in the *freely rotating* scenario. This includes Savitzky-Golay filtering as a control signal for the CDE, iterative denoising, the second order control signal, and learning the weight matrix $\mathbf{W}$, which all notably increase performance over the baseline model. Notably, a naive adaptation of the Savitzky-Golay filter does not initially improve performance due to boundary artifacts [52]; the strong performance of our method is a culmination of the possibilities opened up by using robust control signals together with neural CDEs.

Number of Function Evaluations (NFE). NFE assesses computational efficiency in differential equation solvers correlating with the complexity of the numerical integration. We observe that SG-nCDEs have lower and more consistent NFE counts throughout the training process; our 2nd order model requires around half as many function evaluations, while the first-order filter model requires more than an order of magnitude less than the baseline $\mathrm{SO}(3)$-nCDE. This implies the that Savitzky-Golay filtering yields a more easily numerically integrable function in Def. (4).

Runtime. As evident in Tab. 3, SG-nCDEs introduce marginal additional computational overhead. While these techniques improve accuracy, practitioners should weigh the performance tradeoffs against their application requirements.

	Robust SG Regression	2nd-Order Control Signal	Adaptive $\mathbf{W}$	Freely Rotating (RGE)	Runtime (ms)
i				3.62 ± 0.24	74.6 ± 0.7
ii	✓			1.91 ± 0.12	60.9 ± 1.4
iii	✓		✓	1.98 ± 0.13	60.9 ± 1.2
iv	✓	✓		1.31 ± 0.08	74.5 ± 1.1
v	✓	✓	✓	1.28 ± 0.08	76.8 ± 1.2

Table 3. Ablating our modeling components and runtime. We study the robust Savitzky-Golay (SG) regression, adaptive weighting $\mathbf{W}$, and the 2nd-order CDE. (i) represents the $\mathrm{SO}(3)$-nCDE baseline with no other modifications. Extrapolation is evaluated in rotational geodesic error at 1.2s on the freely rotating scenario.

6. Conclusion

We propose a method for extrapolating rotational motion for dynamical systems in $\mathrm{SO}(3)$ with unknown physical parameters. This problem is challenging and understudied; previous works either consider pairwise predictions or assume energy conservation, neglecting long-term forecasting in $\mathrm{SO}(3)$. Our method achieves more accurate predictions than existing works by regressing a robust control path directly on the manifold of rotations $\mathrm{SO}(3)$. This path is used as a control signal for a neural CDE modeling the evolving dynamical system. Our approach outperforms existing methods in accuracy and extrapolation robustness across several settings.

Limitations and future work. Our approach is limited to the special orthogonal group $\mathrm{SO}(3)$, whereas the group of rigid motions $\mathrm{SE}(3)$ also involves translational components. As the extrapolation of Euclidean quantities has been extensively studied, extending our framework to SE(3) presents an interesting opportunity for future work. Particularly exciting is the study of coupled motion between rotation and translation (e.g., centripetal and Coriolis forces) and filtering in the Lie algebra of twists. Nevertheless, robust filtering on $\mathrm{SE}(3)$ risks inadvertently decoupling rotation and translation [39, 53, 60], posing problems when studying gyroscopic coupling effects. In the future, we will also explore different path parameterizations.

Acknowledgements. The authors would like to thank Junwen Huang and Benjamin Busam for fruitful early discussions. The authors are grateful for the support of the Excellence Strategy of German local and state governments, including computational resources of the LRZ AI service infrastructure provided by the Leibniz Supercomputing Center (LRZ), the German Federal Ministry of Education and Research (BMBF), and the Bavarian State Ministry of Science and the Arts (StMWK). T. Birdal acknowledges support from the Engineering and Physical Sciences Research Council [grant EP/X011364/1]. T. Birdal was supported by a UKRI Future Leaders Fellowship [grant MR/Y018818/1].

References

[1] A. Borisov and I.S. Mamaev. *Rigid Body Dynamics*. De Gruyter, 2018. 1

[2] Michael Bosse and Robert Zlot. Continuous 3d scan-matching with a spinning 2d laser. In *Int. Conf. Robot. Automation.*, pages 4312–4319. IEEE, 2009. 1, 2

[3] Romain Brégier. Deep regression on manifolds: a 3d rotation case study. In *Int. Conf. 3D Vision*, pages 166–174. IEEE, 2021. 2, 5

[4] Benjamin Busam, Tolga Birdal, and Nassir Navab. Camera pose filtering with local regression geodesics on the riemannian manifold of dual quaternions. In *ICCVW*, pages 2436–2445, 2017. 2

[5] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *Int. Conf. Robot. Automation.*, pages 173–180. IEEE, 2017. 2

[6] Jiayi Chen, Yingda Yin, Tolga Birdal, Baoquan Chen, Leonidas J Guibas, and He Wang. Projective manifold gradient layer for deep rotation regression. In *CVPR*, pages 6646–6655, 2022. 2

[7] Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. 31, 2018. 2

[8] Yamei Chen, Yan Di, Guangyao Zhai, Fabian Manhardt, Chenyangguang Zhang, Ruida Zhang, Federico Tombari, Nassir Navab, and Benjamin Busam. Secondpose: Se (3)-consistent dual-stream feature fusion for category-level pose estimation. In *CVPR*, pages 9959–9969, 2024. 3

[9] Hyunggi Cho, Young-Woo Seo, Vijaya Kumar, and Ragunathan R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *Int. Conf. Robot. Automation.* IEEE, 2014. 1

[10] Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLRW*, 2020. 2

[11] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. 32, 2019. 2

[12] Thai Duong and Nikolay Atanasov. Hamiltonian-based neural ode networks on the se (3) manifold for dynamics learning and control. In *Robotics: Science and Systems (RSS)*, 2021. 1, 2

[13] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *CVPR*, pages 9710–9719, 2021. 1

[14] Zicong Fan, Takehiko Ohkawa, Linlin Yang, Nie Lin, Zhishan Zhou, Shihao Zhou, Jiajun Liang, Zhong Gao, Xuanyang Zhang, Xue Zhang, et al. Benchmarks and challenges in pose estimation for egocentric hand interactions with objects. In *ECCV*, pages 428–448. Springer, 2025. 1, 2

[15] Andreas René Geist, Jonas Frey, Mikel Zhobro, Anna Levina, and Georg Martius. Learning with 3d rotations, a hitchhiker's guide to so (3). In *Int. Conf. Mach. Lear.*, 2024. 2, 5

[16] Giulio Giacomuzzo, Ruggero Carli, Diego Romeres, and Alberto Dalla Libera. A black-box physics-informed estimator based on gaussian process regression for robot inverse dynamics identification. *IEEE Transactions on Robotics*, 2024. 2

[17] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison Wesley, San Francisco, 3 edition, 2002. 6

[18] Shota Gugushvili and Chris AJ Klaassen. Consistent parameter estimation for systems of ordinary differential equations: bypassing numerical integration via smoothing. *Bernoulli*, pages 1061–1098, 2012. 1

[19] Adrian Haarbach, Tolga Birdal, and Slobodan Ilic. Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation. In *Int. Conf. 3D Vision*. IEEE, 2018. 2

[20] Yannan He, Garvita Tiwari, Tolga Birdal, Jan Eric Lenssen, and Gerard Pons-Moll. Nrdf: Neural riemannian distance fields for learning articulated pose priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1661–1671, 2024. 2

[21] David Hug and Margarita Chli. Hyperslam: A generic and modular approach to sensor fusion and simultaneous localization and mapping in continuous-time. In *Int. Conf. 3D Vision*, pages 978–986. IEEE, 2020. 2

[22] Du Q Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35: 155–164, 2009. 6

[23] Sheo Yon Jhin, Minju Jo, Seungji Kook, and Noseong Park. Learnable path in neural controlled differential equations. In *AAAI*, pages 8014–8022, 2023. 3, 6, 7

[24] Chao Jia and Brian L Evans. Constrained 3d rotation smoothing via global manifold regression for video stabilization. *IEEE Trans. on Sig. Proc.*, 62(13):3293–3304, 2014. 2

[25] Maarten Jongeneel and Alessandro Saccon. Geometric savitzky-golay filtering of noisy rotations on so (3) with simultaneous angular velocity and acceleration estimation. In *Int. Conf. Intell. Robot. and Syst.*, pages 2962–2968. IEEE, 2022. 2, 3, 4

[26] Kevin Michael Judd and Jonathan D Gammell. The oxford multimotion dataset: Multiple se (3) motions with ground truth. *Robo. Auto. Let.*, 4(2):800–807, 2019. 5, 7

[27] HyunJun Jung, Shun-Cheng Wu, Patrick Ruhkamp, Guangyao Zhai, Hannah Schieber, Giulia Rizzoli, Pengyuan

Wang, Hongcheng Zhao, Lorenzo Garattoni, Sven Meier, et al. Housecat6d-a large-scale multi-modal category level 6d object perception dataset with household objects in realistic scenarios. In *CVPR*, pages 22498–22508, 2024. 3

[28] Patrick Kidger. *On Neural Differential Equations*. PhD thesis, University of Oxford, 2021. 7

[29] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. 33:6696–6707, 2020. 2, 3

[30] Myoung-Jun Kim, Myung-Soo Kim, and Sung Yong Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *Computer graphics and interactive techniques*, pages 369–376, 1995. 2

[31] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 2

[32] Muriel Lang and Sandra Hirche. Computationally efficient rigid-body gaussian process for motion dynamics. *Robo. Auto. Let.*, 2, 2017. 2

[33] Muriel Lang, Oliver Dunkley, and Sandra Hirche. Gaussian process kernels for rotations and 6d rigid body motions. In *Int. Conf. Robot. Automation.*, pages 5165–5170. IEEE, 2014. 2

[34] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Global formulations of lagrangian and hamiltonian dynamics on manifolds. *Springer*, 13:31, 2017. 3

[35] Jake Levinson, Carlos Esteves, Kefan Chen, Noah Snavely, Angjoo Kanazawa, Afshin Rostamizadeh, and Ameesh Makadia. An analysis of svd for deep rotation estimation. 33: 22554–22565, 2020. 2

[36] Sven Lilge and Timothy D Barfoot. Incorporating control inputs in the estimation of continuous mobile robot trajectories and continuum robot shapes. *arXiv preprint arXiv:2408.01333*, 2024. 2

[37] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. Gdrnpp. https://github.com/shanice-l/gdrnpp_bop2022, 2022. 7

[38] Steven Lovegrove, Alonso Patron-Perez, and Gabe Sibley. Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras. In *BMVC*, page 8, 2013. 2

[39] Nathan Mankovich and Tolga Birdal. Chordal averaging on flag manifolds and its applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3881–3890, 2023. 8

[40] Naresh Marturi, Marek Kopicki, Alireza Rastegarpanah, Vijaykumar Rajasekaran, Maxime Adjigble, Rustam Stolkin, Aleš Leonardis, and Yasemin Bekiroglu. Dynamic grasp and trajectory planning for moving objects. *Autonomous Robots*, 43:1241–1256, 2019. 1

[41] Justice J Mason, Christine Allen-Blanchette, Nicholas Zolman, Elizabeth Davison, and Naomi Ehrich Leonard. Learning to predict 3d rotational dynamics from images of a rigid body with unknown mass distribution. *Aerospace*, 10(11): 921, 2023. 1, 2, 6, 7

[42] James Morrill, Patrick Kidger, Lingyi Yang, and Terry Lyons. On the choice of interpolation scheme for neural cdes. *Trans. Mach. Lear. Resea.*, 2022(9), 2022. 3, 5

[43] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 2

[44] K Nirmal, AG Sreejith, Joice Mathew, Mayuresh Sarpotdar, Ambily Suresh, Ajin Prakash, Margarita Safonova, and Jayant Murthy. Noise modeling and analysis of an imu-based attitude sensor: improvement of performance by filtering and sensor fusion. In *Advances in optical and mechanical technologies for telescopes and instrumentation II*, pages 2138–2147. SPIE, 2016. 2

[45] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. In *ECCV*, pages 163–182. Springer, 2025. 3

[46] Wooram Park, Yan Liu, Yu Zhou, Matthew Moses, and Gregory S Chirikjian. Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map. *Robotica*, 26(4):419–434, 2008. 1

[47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. 32, 2019. 7

[48] Valentin Peretroukhin, Matthew Giamou, W. Nicholas Greene, David Rosen, Jonathan Kelly, and Nicholas Roy. A smooth representation of belief over so(3) for deep rotation learning with uncertainty. *Robotics: Science and Systems*, 2020. 2

[49] Mikael Persson, Gustav Häger, Hannes Ovrén, and Per-Erik Forssén. Practical pose trajectory splines with explicit regularization. In *Int. Conf. 3D Vision*, pages 156–165. IEEE, 2021. 2

[50] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019. 2

[51] Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. 32, 2019. 2

[52] Michael Schmid, David Rath, and Ulrike Diebold. Why and how savitzky–golay filters should be replaced. *ACS Measurement Science Au*, 2(2):185–196, 2022. 4, 8

[53] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative b-splines on lie groups. In *CVPR*, 2020. 2, 3, 8

[54] William Talbot, Julian Nubert, Turcan Tuna, Cesar Cadena, Frederike Dümbgen, Jesus Tordesillas, Timothy D Barfoot, and Marco Hutter. Continuous-time state estimation methods in robotics: A survey. *arXiv preprint arXiv:2411.03951*, 2024. 2

[55] Tim Yuqing Tang, David Juny Yoon, and Timothy D Barfoot. A white-noise-on-jerk motion prior for continuous-time trajectory estimation on se (3). *Robo. Auto. Let.*, 4, 2019. 2

[56] Zachary Teed and Jia Deng. Tangent space backpropagation for 3d transformation groups. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 10338–10347, 2021. 2

[57] Peter Toth, Danilo J Rezende, Andrew Jaegle, Sébastien Racanière, Aleksandar Botev, and Irina Higgins. Hamiltonian generative networks. In *ICLR*, 2020. 2

[58] Chen Wang, Dasong Gao, Kuan Xu, Junyi Geng, Yaoyu Hu, Yuheng Qiu, Bowen Li, Fan Yang, Brady Moon, Abhinav Pandey, et al. Pypose: A library for robot learning with physics-based optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22024–22034, 2023. 2

[59] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, pages 16611–16621, 2021. 3, 5, 6, 7

[60] Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. Computation of rotation minimizing frames. *ACM Transactions on Graphics (TOG)*, 27(1):1–18, 2008. 8

[61] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1):1–34, 2020. 2

[62] Jeremy N Wong, David J Yoon, Angela P Schoellig, and Timothy D Barfoot. A data-driven motion prior for continuous-time trajectory estimation on se(3). *Robo. Auto. Let.*, 5, 2020. 2

[63] Tianhao Wu, Fangwei Zhong, Yiran Geng, Hongchen Wang, Yongjian Zhu, Yizhou Wang, and Hao Dong. Grasparl: Dynamic grasping via adversarial reinforcement learning. *arXiv preprint arXiv:2203.02119*, 2022. 1

[64] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas Guibas, and Federico Tombari. Quaternion equivariant capsule networks for 3d point clouds. In *ECCV*, pages 1–19. Springer, 2020. 2

[65] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019. 2, 5

Forecasting Continuous Non-Conservative Dynamical Systems in SO(3)

Appendix

Lennart Bastian[1,2*] **Mohammad Rashed**[1,2*] **Nassir Navab**[1,2] **Tolga Birdal**[3]

[1] **Technical University of Munich** [2] **Munich Center of Machine Learning** [3] **Imperial College London**

Contents

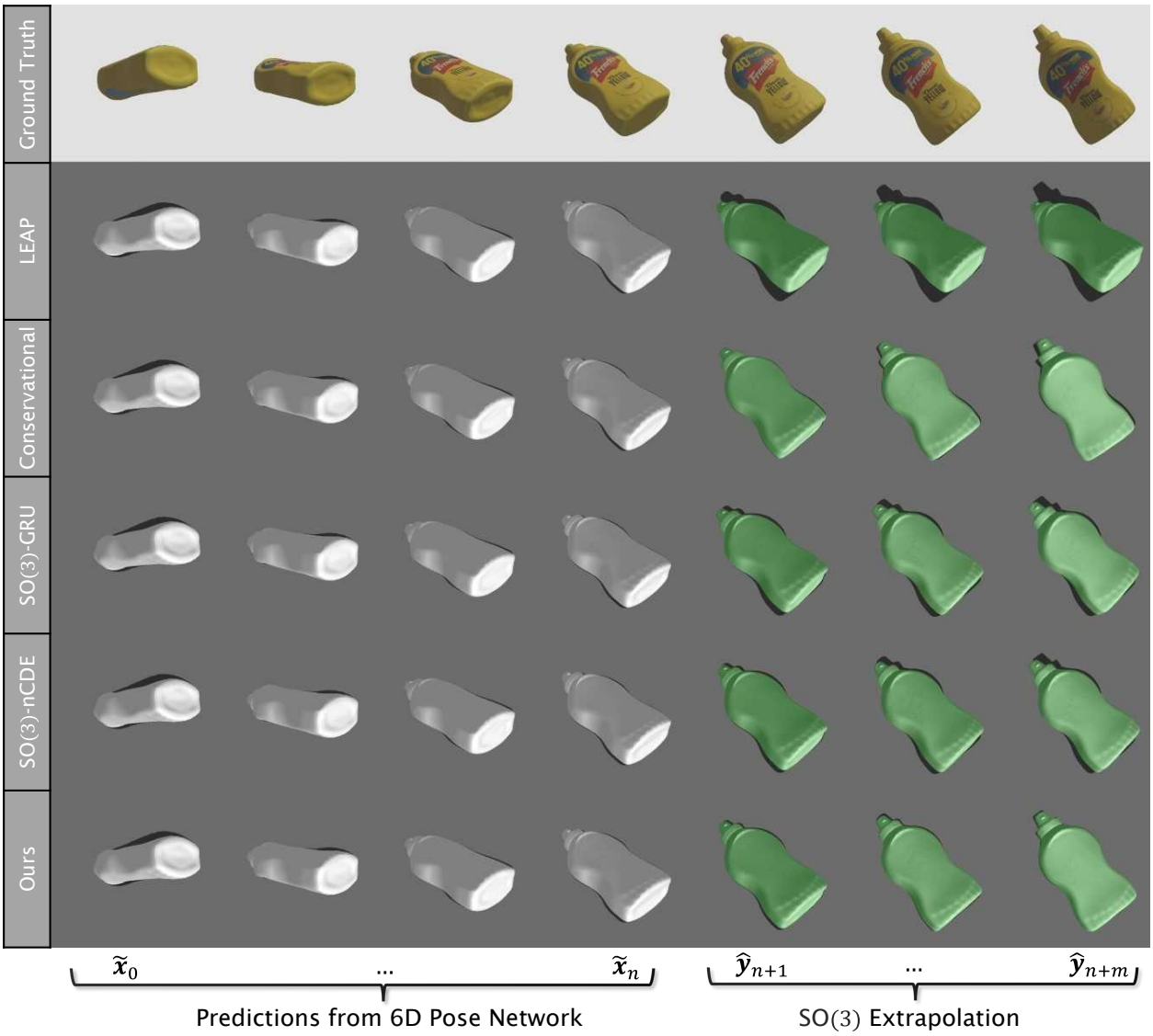

Figure 1. Application: 6D object pose estimation. We compare our method with the baselines by applying all models on the pose prediction outputs of GDRNet [17]. 6D pose predictions (bottom left) are used as input to the $\mathrm{SO}(3)$extrapolation methods. Extrapolated outputs are depicted in green. Our method is able to predict more robust future states based on noisy pose estimates.

A. Experiments and Ablations

A.1. 6D Pose Estimation

In this section, we provide additional details regarding the 6D Pose estimation application introduced in the main paper. A total of 25k images are generated using BlenderProc2 [4]. The synthetic scenes featured a single object (a mustard bottle) from the YCB-Video dataset [18], with the object's orientation randomly sampled. We then train GDRNet [17] on individual (non-sequential) RGB images. The publicly available pre-trained weights are fine-tuned for one training epoch, without data augmentations.

Figure 1 illustrates simulated trajectories of the object for the configuration-dependent torque scenario. In this experiment, image sequences are given as input to the pose estimator, producing noisy $\mathrm{SO}(3)$ rotation trajectories. These are then used as input for the $\mathrm{SO}(3)$ extrapolation methods. We compare our method to the baselines LEAP, Conservational, $\mathrm{SO}(3)$-GRU, and $\mathrm{SO}(3)$-nCDE.

As shown in Figure 1, the LEAP baseline struggles to handle the rotational behavior effectively. The conservational

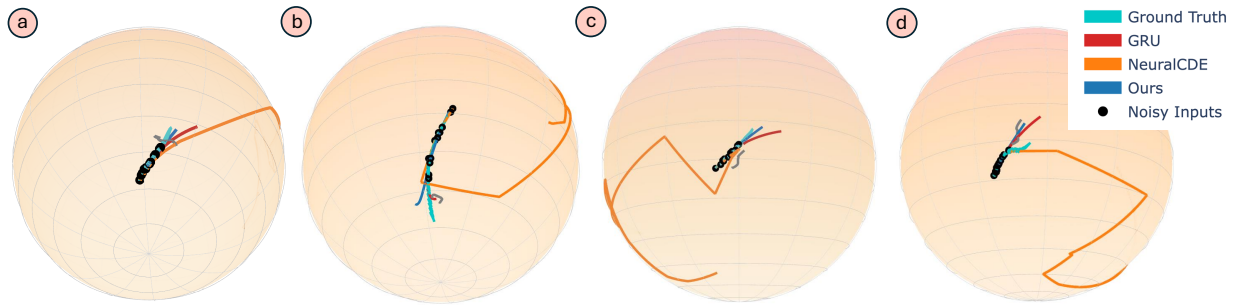

Figure 3. We evaluate the best-performing models' extrapolation capabilities under irregular sampling from noisy samples fused from multiple sensors. The trajectories are visualized on the unit sphere $\mathbb{S}^2$.

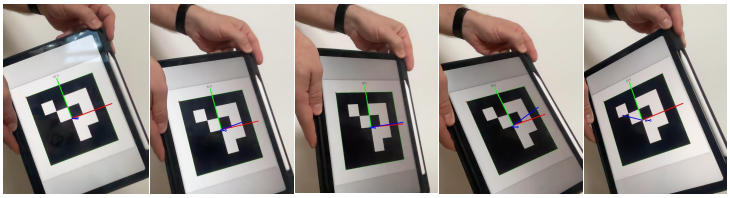

Figure 4. Evaluating the robustness of the proposed extrapolation methods for irregularly sampled extrapolation in the wild. The IMU signal from a tablet is synchronized with an external camera which captures the pose of a displayed ArUco marker. The resulting noisy and irregular trajectories are used to evaluate the models' extrapolation capabilities.

Table 1. Comparison of prediction errors across different models from irregularly sampled noisy sensor measurements. Results show mean ± standard deviation in rotational geodesic error (RGE).

Model	Prediction Error
LEAP	18.30 ± 5.45
$\mathrm{SO}(3)$-GRU	11.39 ± 0.73
$\mathrm{SO}(3)$-nCDE	75.48 ± 9.39
Ours	$\mathbf{8.11 \pm 1.59}$

approach effectively predicts rotation within this time horizon but does not capture the damping behavior and overshoots slightly. While the $\mathrm{SO}(3)$-GRU and $\mathrm{SO}(3)$-nCDE baselines perform better, managing to account for the damped patterns, the extrapolated poses produced by these methods exhibited higher errors than our approach.

A.2. Application: Irregularly Sampled Sensor Fusion

To evaluate the capabilities of the proposed method in real-world scenarios, we study the case of extrapolation in a sensor fusion application. The data rig consists of a tablet depicting an ArUco [5] marker that streams the onboard inertial measurement unit (IMU) sampled at 100Hz sensor signal to a system that simultaneously captures the 6D pose of the tablet with a camera (sampled at 30Hz) as shown in figure Fig. 4. The sensor is then moved manually and tracked, a particularly challenging (and potentially ill-defined) scenario as stochastic external torques can influence the device during the extrapolation period. Sensors are calibrated spatiotemporally using a Levenberg-Marquardt optimization, after which we perform outlier rejection based on a simple median-absolute-deviation heuristic. We then capture 24 trajectories with the sensor rig to evaluate the models trained on the simulated *combined external torque* scenarios. The trajectories from the two sensors are naively merged according to the registered timestamps and then used as input to the models.

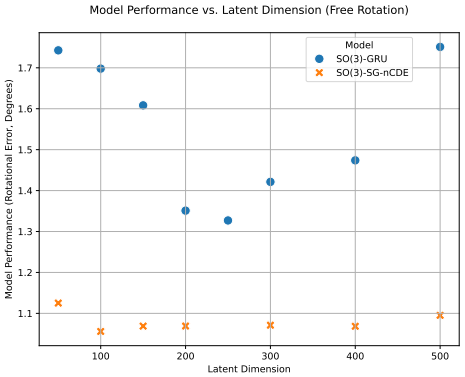

Figure 2. Ablating latent dimension vs. prediction performance in GRU vs. neural CDEs on the *freely rotating* scenario with $\delta = 0.05$ and a prediction horizon of $t = 0.8s$.

Results The results are depicted in Tab. 1, with visualizations in Fig. 3. Our method outperforms the baselines LEAP and $\mathrm{SO}(3)$-nCDE by a large margin. In contrast to the simulated scenarios, the latter cannot accommodate the irregular, noisy signals and creates errant extrapolations. The GRU performs competitively, despite the irregular sampling, and produces reasonable estimates. Fig. 3 (d) depicts an example where the applied torque changes in a stochastic manner during the extrapolation horizon, which the models are not

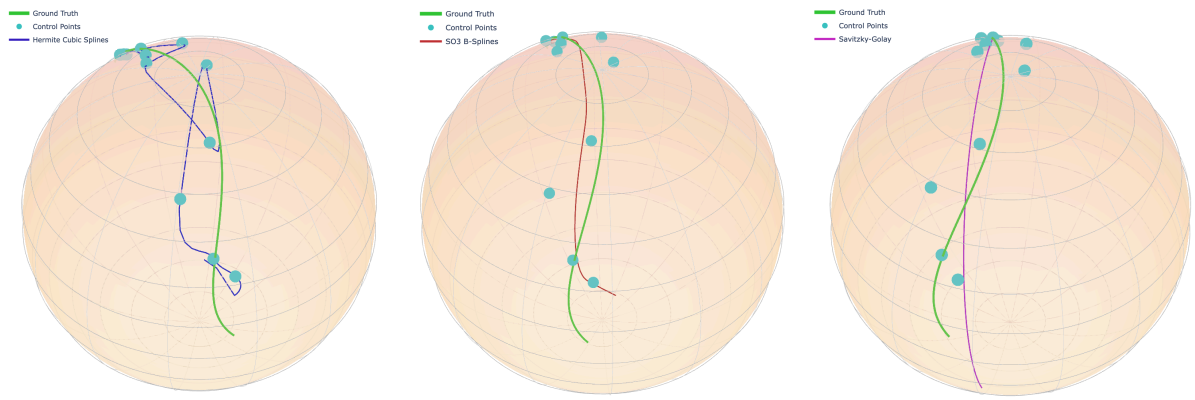

Figure 5. Comparison of interpolation methods on a particularly complex and noisy trajectory from the *combined external force* scenario. From left to right: Hermite cubic splines with backward differences [10], $\mathrm{SO}(3)$ B-splines [14], and $\mathrm{SO}(3)$ Savitzky Golay Filtering [7]. Trajectories represent interpolation and $t = 0.2s$ extrapolation into the future.

able to capture as they have no knowledge of such events.

A.3. Hyperparameters

As the latent dimension significantly impacts model performance for both the GRU and nCDE-based methods, we provide an additional ablation over the latent dimension (see Fig. 2).

While increasing the latent dimension of the GRU leads to overfitting, $\mathrm{SO}(3)$-SG-nCDEs are robust to changes in latent dimension and overfitting. $\mathrm{SO}(3)$-nCDEs and $\mathrm{SO}(3)$-SG-nCDEs are trained with a latent dimension of 125, while GRUs are trained with a latent dimension of 250 and 3 hidden units. LEAP uses a latent dimension of 50; increasing the latent dimension for this method significantly increases runtime and VRAM but not performance.

A.4. Choosing Integration Paths

Various control signals for neural CDEs have been proposed; most interpolation methods use cubic or higher order splines [8, 10]. Kidger et al. [8] construct the integration path $X \in \mathcal{C}^2$ as a natural cubic spline over the input sequence. While this guarantees smoothness of the first derivative, Hermite cubic splines with backward differences are preferred for CDE integration due to local control [10]. We observe that these choices do not obey the geometric structure of $\mathrm{SO}(3)$; furthermore, they are not robust with respect to sensor noise interpolating the points erratically (see Fig. 5, left). In general, such higher-order polynomials tend to diverge near the interpolation endpoints and must be extrapolated with some assumption, for instance, constant velocity [11]. However, when dynamics are complicated, this assumption is overly simplifying, even more so than assuming conservation of energy as in [9].

Efficient derivative calculations have been proposed for robust $\mathrm{SO}(3)$ B-spline interpolation of trajectories in $\mathrm{SO}(3)$ [14]. These can be made suitable for extrapolation with, e.g., constant velocity extrapolation at the endpoint by constructing a nonlinear optimization to minimize residuals to the measurements, with additional smoothness near the endpoint. We implement this in Ceres [1], Basalt [16], and Sophus [15] to integrate Lie group constraints into the manifold optimization. We use the efficient derivative computation of [14] and an additional first-order Taylor expansion using the computed Jacobians to include a constant velocity endpoint constraint defining the splines beyond their support interval. While the splines nicely interpolate the noisy trajectory and can be used to extrapolate (see Fig. 5, center), they still exhibit bias towards endpoints. Moreover, this approach is too computationally expensive to use during network training. On a 24-Core Intel(R) Xeon(R) CPU, a single batch of 1000 trajectories takes ≈ 22 seconds when maximally parallelized, a fraction of the trajectories we use for training.

On the other hand, the Savitzky-Golay filter provides a reasonable approximation at a low cost. As it merely requires solving a modest linear system, this means it can be differentiated through to learn a robust weighting suitable for extrapolation (see Fig. 5, right).

A.5. Model performance vs Input Noise

In Fig. 6, we further evaluate the quality of the predictions over varying time horizons (1, 4, 6, 8) and level of noise $[0.02, 0.03, 0.04, 0.05]\pi$ on the *velocity damping* scenario. The results are depicted in Fig. 6. Notably, all methods

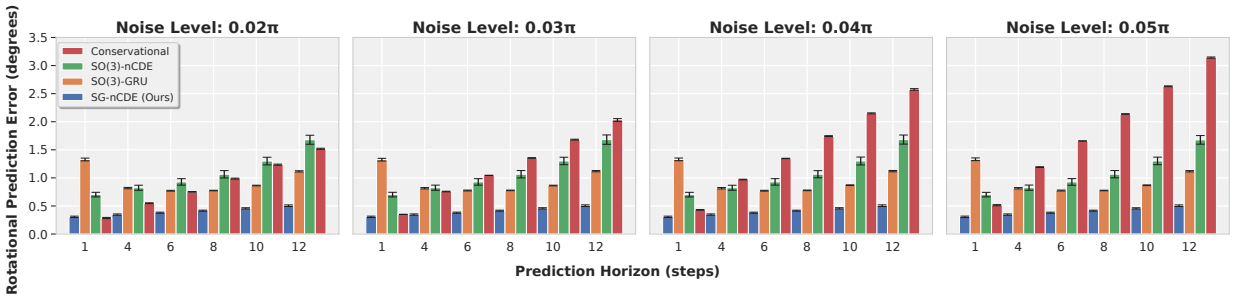

Figure 6. Rotational geodesic error for different input noise levels on the velocity damping scenario. We compare the performance of the baseline neural CDE and our best performing method (weighted, 2nd-order) with different amounts of simulated noise added to the simulated dynamical trajectories. Noise is sampled from $\mathcal{N}(0, \delta)$ with $\delta \in [0.02, 0.05]\pi$.

suffer from increased prediction errors as noise increases, but our method consistently outperforms the baselines.

B. Implementation Details

We provide further details regarding the five simulated experimental scenarios described in the main paper.

B.1. Experimental Design

For each experiment, we sample an initial orientation according to the strategy in [13], and an initial angular velocity is sampled according to a truncated normal distribution to avoid degenerate cases with a small angular velocity.

We sample via rejection sampling:

1. Sample $X \sim \mathcal{N}(0, \sigma)$
2. Accept if $|X| > \eta$, otherwise repeat step 1,

with $\sigma = 0.3$ and $\eta = 0.1$

B.2. Moment of Inertia Distributions for Simulation

To evaluate model generalization across different rigid body properties, we simulate trajectories using four distinct moment of inertia (MOI) distributions. Each distribution is characterized by a base diagonal MOI tensor and additive noise:

$$\mathbf{J} = \text{diag}(\mathbf{J}_{\text{base}}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_3) \tag{1}$$

where $\mathbf{J}_{\text{base}}$ is the base MOI vector and $\sigma = 0.2$ is the noise standard deviation. The four distributions use the following base MOI configurations:

$$\mathbf{J}_{\text{base}}^{(1)} = [1.0, 2.0, 3.0] \tag{2}$$
$$\mathbf{J}_{\text{base}}^{(2)} = [3.0, 1.0, 2.0] \tag{3}$$
$$\mathbf{J}_{\text{base}}^{(3)} = [3.0, 2.0, 1.0] \tag{4}$$
$$\mathbf{J}_{\text{base}}^{(4)} = [2.0, 3.0, 1.0] \tag{5}$$

These distributions represent different permutations of the same eigenvalues, creating rigid bodies with varied principal axis orientations. This approach challenges models to generalize across different rotational behaviors resulting from the same underlying physics but with different inertial configurations.

B.3. Simulation Scenarios

Freely Rotating. Freely rotating objects are governed by kinematic equations. Without any external torques, they reduce to the following:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \tag{6}$$

While this equation is simplified and maintains energy conservation, the gyroscopic torque defined through the cross-product is highly non-linear and can lead to chaotic motion depending on the mass distribution.

Linear Control. In the case of having an identity moment of inertia matrix, the gyroscopic torque can be neglected, and the kinematic equations

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \tau_{\text{ext}} \tag{7}$$

reduce to:

$$\mathbf{J}\dot{\boldsymbol{\omega}} = \tau_{\text{ext}} \tag{8}$$

To this end, we simulate a linear controller where all acceleration is *driven* by an external torque which changes according to the linear control law $\tau_{\text{ext}} = \mathbf{J}(\mathbf{A}\boldsymbol{\omega} + \mathbf{b})$ where $A \in \mathbb{R}^{3\times3}$ and $b \in \mathbb{R}^3$.

By defining τ_{ext} in this manner, we see that the angular velocity changes according to the external control signal directly as a function of orientation: $\dot{\boldsymbol{\omega}} = (\mathbf{A}\boldsymbol{\omega} + \mathbf{b})$, which is a useful simplified external torque without highly non-linear gyroscopic effects. Notably, the baseline $\mathrm{SO}(3)$-nCDE and $\mathrm{SO}(3)$-GRU also perform competitively in this scenario due to it's simplicity.

Configuration Dependent Torque. We further consider scenarios where the external torque depends explicitly on the current orientation, leading to

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} = \tau_{\text{ext}}(R) \tag{9}$$

where $R \in \mathrm{SO}(3)$ is the current orientation. This class of torques arises in various physical systems where the interaction with external fields generates configuration-dependent forcing, such as magnetic dipoles in uniform fields or gravitational gradients.

Unlike the linear control case, the rotational motion is coupled directly to the orientation state in a global reference frame, potentially leading to multiple equilibria and complex trajectories. The external torque takes the general form $\tau_{\text{ext}}(R) = f(R\mathbf{v})$ where $\mathbf{v} \in \mathbb{R}^3$ represents some body-fixed vector and $f : \mathbb{R}^3 \to \mathbb{R}^3$ is a nonlinear function that describes the physical interaction in the world frame. We observe this scenario to be most challenging for the models as the trajectories can exhibit erratic motion.

To limit chaotic behavior, we additionally apply a damping signal (e.g., simulating friction or other dissipative forces). We initialize the damping matrix $\mathbf{D}$ to be negative definite such that $\lim_{t\to\infty} \|\boldsymbol{\omega}(t)\| = 0$ while the object is also under the effect of the internal gyroscopic motion:

$$\mathbf{D} = \begin{bmatrix} -0.2 & 0 & 0 \\ 0 & -0.2 & 0 \\ 0 & 0 & -0.2 \end{bmatrix} \tag{10}$$

These are combined for the following total external torque:

$$\tau_{\text{ext}} = w_1\tau_{\text{config}}(R) + w_2\tau_{\text{damp}}(\boldsymbol{\omega}) \tag{11}$$

where the weights w_i allow for scaling individual contributions. Despite combining multiple effects, these scenarios often exhibit more predictable behavior than pure configuration-dependent cases, as damping terms provide a stabilizing effect by dissipating energy from the system.

B.4. Baseline Model Design

We provide additional details regarding the baseline models. Specifically, how we adapted the $\mathrm{SO}(3)$-GRU and $\mathrm{SO}(3)$-nCDE baselines to predict rotations in $\mathrm{SO}(3)$. We also detail the essential components of the conservation approach [9] that are used to predict rotations directly without confounding pose estimation from images with object dynamics.

$\mathrm{SO}(3)$**-nCDE**. We adapt the neural CDE baseline from [8] to predict rotations in $\mathrm{SO}(3)$. Hermit cubic coefficients are directly constructed on the input 9D rotation representation [19] via backward differences. Following [8], the method encodes an initial value z_0, which is then integrated forward in time using *torchdiffeq* [3] and Dormand-Prince 4/5 with respect to the constructed spline. The latent representation is then decoded into the 6D rotation representation [6, 19], and transformed to a rotation matrix with Gram-Schmidt orthonormalization (GSO).

Table 2. Comparison of prediction errors with the conservational approach [9] using a ground truth momentum estimate across different experimental conditions. Results show mean ± standard deviation in rotational geodesic error (RGE). First and second best results are emphasized.

Extrapolation Horizon	Method	Freely Rotating	Linear Control	Velocity Damping	Configuration-Dependent Torque	Variable Dynamics
$t = 0.8s$	Conservational ([9])	1.15 ± 0.00	1.25 ± 0.00	1.14 ± 0.00	1.26 ± 0.00	1.43 ± 0.00
	SG-nCDE (Ours)	$\mathbf{0.87 \pm 0.05}$	$\mathbf{0.49 \pm 0.03}$	$\mathbf{0.42 \pm 0.02}$	$\mathbf{0.58 \pm 0.03}$	$\mathbf{0.89 \pm 0.07}$
$t = 1.2s$	Conservational ([9])	$\mathbf{1.15 \pm 0.00}$	2.05 ± 0.00	1.87 ± 0.00	2.05 ± 0.00	2.44 ± 0.01
	SG-nCDE (Ours)	1.28 ± 0.08	$\mathbf{0.64 \pm 0.03}$	$\mathbf{0.50 \pm 0.03}$	$\mathbf{0.74 \pm 0.04}$	$\mathbf{1.31 \pm 0.14}$

$\mathrm{SO}(3)$**-GRU**. The GRU baseline consists of a multi-layer gated recurrent unit (GRU) RNN with three recurrent units. The GRU is similarly applied sequentially on each 9D rotation representation, yielding a 6D prediction converted to an element in $\mathrm{SO}(3)$ via GSO.

Conservational [9]. The conservation of energy approach of [9] is designed to learn a rotation representation directly from images. A sequence of images is used to estimate the object's momentum, upon which they integrate the kinematics equations (the authors equivalently use the Lie–Poisson formulation) forward in time and obtain estimates of the $\mathrm{SO}(3)$ state change under energy conservation. They then reconstruct the predicted image sequence, and estimate predicted rotation accuracy based on image reconstruction.

We observe that the dynamics component of this pipeline is identical to how we simulate the *Free Rotation* scenario in Appendix B.3. However, for forecasting rotations from a trajectory, this approach has several limitations:

1. by solving an initial value problem, they can only condition the solve with a single value
2. high sensitivity to the initial momentum estimate
3. cannot handle non-conservative systems ($\tau_{ext} \neq 0$)

To study this method more generally, we provide a momentum estimate and similarly integrate the equations of motion forward in time, evaluating the accuracy directly via rotational geodesic error (RGE) instead of for a downstream image reconstruction as in [9]. In this way, we decouple the image reconstruction task and the associated image quality metrics (the authors use pixel mean-squared error) from dynamics estimation. We emphasize that our problem setting encompasses this task; any dynamics module can be plugged directly into an object pose estimator to forecast rotation.

For a fair comparison, we additionally provide results with a ground-truth momentum estimate, observing that our method outperforms [9] in all cases but conservational, where the trajectories are parallel to the ground-truth trajectories (they are offset by the noise of the last observation). The results can be seen in table Tab. 2.

In practice, obtaining an accurate momentum estimate of an unknown object would be challenging to obtain from a single trajectory as it essentially requires estimating the velocity and mass distribution. The authors demonstrate this on a limited set of synthetically generated and well-cropped images; however, such an approach does not generalize to unknown mass distributions, let alone objects, limiting applicability in practice.

C. Additional Background

C.1. Moment of Inertia and Diagonalization

The moment of inertia (MOI) tensor $\mathbf{J}$ is defined as the integral of the mass distribution that characterizes a body's resistance to rotational acceleration about any axis. For a continuous mass distribution with density $\eta(\mathbf{r})$:

$$J_{ij} = \begin{bmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{xy} & J_{yy} & J_{yz} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} \tag{12}$$

with components:

$$
\begin{aligned}
J_{xx} &= \int \eta(\mathbf{r})(y^2 + z^2)\, dV \\
J_{yy} &= \int \eta(\mathbf{r})(x^2 + z^2)\, dV \\
J_{zz} &= \int \eta(\mathbf{r})(x^2 + y^2)\, dV \\
J_{xy} = J_{yx} &= -\int \eta(\mathbf{r}) xy\, dV \\
J_{xz} = J_{zx} &= -\int \eta(\mathbf{r}) xz\, dV \\
J_{yz} = J_{zy} &= -\int \eta(\mathbf{r}) yz\, dV
\end{aligned}
$$

where $\mathbf{r} = (x, y, z)$ and $\mathbf{r}^2 = x^2 + y^2 + z^2$. Diagonalization yields the principal moments:

$$
\mathbf{P}^{-1}\mathbf{J}\mathbf{P} = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \tag{13}
$$

where I_k are eigenvalues of $\mathbf{J}$ and $\mathbf{P}$ contains the corresponding eigenvectors as columns.

As any MOI tensor can be diagonalized, we consider only diagonal uniform and non-uniform MOI in our numerical simulations, with objects rotating in the canonicalized coordinate system. We note that [9] also consider several fixed non-diagonal MOIs, but these reduce to diagonal cases [2]. Instead, we sample from distributions around non-uniform MOI tensors, which are varied during training, validation, and testing, forcing models to generalize.

C.2. Exponential and Logarithmic Maps

Definition 1 (Exponential Map on SO(3))**.** *The exponential map* $\mathrm{Exp} : \mathfrak{so}(3) \to \mathrm{SO}(3)$ *is a surjective mapping from the Lie algebra* $\mathfrak{so}(3)$ *to the Lie group* $\mathrm{SO}(3)$ *defined as:*

$$
\mathrm{Exp}(\boldsymbol{\xi}) = \sum_{n=0}^{\infty} \frac{1}{n!}\boldsymbol{\xi}^n = \mathbf{I} + \boldsymbol{\xi} + \frac{1}{2!}\boldsymbol{\xi}^2 + \frac{1}{3!}\boldsymbol{\xi}^3 + \cdots \tag{14}
$$

where $\boldsymbol{\xi} \in \mathfrak{so}(3)$ *is a skew-symmetric matrix.*

Geometrically, if $\boldsymbol{\xi} = \hat{\mathbf{v}}$ *for some* $\mathbf{v} \in \mathbb{R}^3$ *with* $\|\mathbf{v}\| = \theta$*, then* $\mathrm{Exp}(\boldsymbol{\xi})$ *represents a rotation by angle* θ *about the axis* $\mathbf{v}/\|\mathbf{v}\|$.

Definition 2 (Logarithmic Map on SO(3))**.** *The logarithmic map* $\mathrm{Log} : \mathrm{SO}(3) \to \mathfrak{so}(3)$ *is the local inverse of the exponential map, retrieving the corresponding Lie algebra element from a rotation matrix:*

$$
\mathrm{Log}(\mathbf{R}) = \boldsymbol{\xi} \in \mathfrak{so}(3) \tag{15}
$$

such that $\mathrm{Exp}(\boldsymbol{\xi}) = \mathbf{R}$*. This mapping is well-defined for all* $\mathbf{R} \in \mathrm{SO}(3)$ *where the rotation angle* θ *satisfies* $0 \leq \theta < \pi$.

The logarithmic map Log is not uniquely defined for rotations of angle $\theta = \pi$, as rotations by π radians about antipodal axes $\mathbf{n}$ and $-\mathbf{n}$ yield identical rotation matrices, yielding a singularity.

C.3. SO(3) Savitzky-Golay Filtering [7]

The Savitzky-Golay filter on SO(3) solves a least-squares problem to estimate polynomial coefficients that best fit the noisy rotational data in the Lie algebra, as described in Theorem 10.

The design matrix $\mathbf{A} \in \mathbb{R}^{3(2n+1)\times 3(p+1)}$ takes the form of a Vandermonde matrix with time-shifted polynomial coefficients. It is expanded with the Kroeneker product $\otimes$ and $\mathbf{I}_3$, the identity matrix in $\mathbb{R}^3$ such that $\mathbf{A} = \hat{A} \otimes \mathbf{I}_3$. $\hat{A} \in \mathbb{R}^{(2n+1)\times(p+1)}$ is defined as:

$$\hat{A} = \begin{bmatrix} 1 & (t_{-n} - t_k) & \cdots & \frac{1}{p}(t_{-n} - t_k)^p \\ 1 & (t_{-n+1} - t_k) & \cdots & \frac{1}{p}(t_{-n+1} - t_k)^p \\ \vdots & \vdots & \vdots & \vdots \\ 1 & (t_n - t_k) & \cdots & \frac{1}{p}(t_n - t_k)^p \end{bmatrix}$$

$\mathbf{b}$ is constructed by rotational differences in the lie algebra and has the form:

$$\mathbf{b} = \begin{bmatrix} \mathrm{Log}(\tilde{\mathbf{x}}_{k-n}\tilde{\mathbf{x}}_k^{-1})^\vee \\ \vdots \\ \mathrm{Log}(\tilde{\mathbf{x}}_{k+n}\tilde{\mathbf{x}}_k^{-1})^\vee \end{bmatrix}$$

$(\cdot)^\vee : \mathfrak{so}(3) \to \mathbb{R}^3$ is the inverse of the hat operator that maps skew-symmetric matrices to vectors.

This formulation brings several advantages over conventional interpolation with splines. It filters robustly in the region of a point (with an arbitrary support window) and can be solved without iterative optimization. Moreover, the derivatives are smooth up to the order of the polynomial, which we leverage in the next section to demonstrate universal approximation based on the results from [10].

D. Proof of Proposition 10

We demonstrate our modified Savitzky-Golay filter satisfies the three properties of a suitable control path for the neural CDE as described by Morrill et al. [10]. We first restate the proposition regarding the suitability of the control path.

Proposition (10)**.** *Let $\varphi(t)$ be the control path constructed in Def. (7), with polynomial coefficients $\boldsymbol{\rho}$ based on solving the optimization problem in Thm. (8). If the maximum rotational difference $\delta_n = \max_k \|\mathrm{Log}(\tilde{\mathbf{x}}_k\tilde{\mathbf{x}}_0^{-1})\|$ within the window is bounded, then:*

1. *$\varphi(t)$ is analytic and twice differentiable,*
2. *The derivatives $\varphi'(t)$ and $\varphi''(t)$ are bounded,*
3. *$\varphi(t)$ uniquely minimizes the problem in Thm. (8).*

Proof. **Smoothness**. We begin by demonstrating sufficient smoothness of the map $\varphi(t)$. We observe that $\tilde{\mathbf{x}}_k$ is constant, and the term $\mathrm{Exp}\left(\mathbf{p}\left(t - t_k; \boldsymbol{\rho}_k\right)\right)$ is a composition of a polynomial and the exponential map.

We note that the logarithmic map $\mathrm{Log} : \mathrm{SO}(3) \to \mathfrak{so}(3)$ is not uniquely defined for rotations of angle π. Specifically, for $\|\xi\| = \pi$ the axis of rotation becomes non-unique, as rotations by π about $\mathbf{n}$ and $-\mathbf{n}$ yield the same rotation matrix. This requires constraints or assumptions on the magnitude of rotations within any given filter window to ensure smoothness and injectivity of the maps.

Since the exponential map $\mathrm{Exp} : \mathfrak{so}(3) \to \mathrm{SO}(3)$ is smooth and analytic on $\mathfrak{so}(3)$ [12], and the polynomial $\mathbf{p}\left(t - t_k; \boldsymbol{\rho}_k\right) \in C^\infty$ by definition, we have that $\varphi(t) \in C^\infty$ provided that $\|\mathbf{p}(t - t_k; \boldsymbol{\rho}_k)\| < \pi$ for all t.

In practice, the magnitude of rotational differences encountered in the filtering process is small, especially when dealing with high-frequency sampling and smooth motion trajectories. Therefore, the norm $\|\mathbf{p}(t - t_k; \boldsymbol{\rho}_k)\| < \pi$ holds in most cases. However, to rigorously ensure the smoothness and differentiability of $\varphi(t)$, we impose a constraint that $\|\mathbf{p}(t - t_k; \boldsymbol{\rho}_k)\| < \pi$ for all t within the filter window as the exponential map is injective on this domain. This can be achieved by appropriately selecting the filter window size and ensuring the sampling rate is sufficiently high with respect to the angular velocity.

Next, we consider smoothness of the derivatives $\frac{d}{dt}\varphi(t)$ and $\frac{d^2}{dt^2}\varphi(t)$ of $\varphi(t)$. We observe that they coincide with the angular velocity and acceleration, which are defined by [7] as:

$$\begin{aligned} \widehat{\boldsymbol{\omega}}(t) &= D\mathrm{Exp}\left(\boldsymbol{\varphi}(t)^\wedge\right)\dot{\boldsymbol{\varphi}}(t), \\ \widehat{\dot{\boldsymbol{\omega}}}(t) &= D^2\mathrm{Exp}\left(\boldsymbol{\varphi}(t)^\wedge\right)[\dot{\boldsymbol{\varphi}}(t), \dot{\boldsymbol{\varphi}}(t)] + \\ &\quad D\mathrm{Exp}\left(\boldsymbol{\varphi}(t)^\wedge\right)\ddot{\boldsymbol{\varphi}}(t) \end{aligned}$$

where $D^k f(\cdot)$ denotes the k-th order directional derivative with respect to f. These are, again, compositions of analytic functions and the polynomial derivatives, which are bounded and continuous and, therefore, continuously differentiable. We refer to [7] for a complete derivation of the derivatives.

Boundedness. To show the polynomial coefficients $\boldsymbol{\rho}$ are bounded, we observe that they are calculated via least squares with $\mathbf{A}$ and $\mathbf{b}$ defined as in Appendix C.3. We can see that $\hat{A}$ has full column rank as long as $2n+1 \geq p+1$. Therefore the rank of $\mathbf{A}$ can be obtained via: rank($\mathbf{A}$) = rank($\hat{A}$) $\cdot$ rank($\mathbf{I}_3$). Since both these matrices have full column rank, so does $\mathbf{A}$, and therefore, the matrix $\mathbf{A}^T\mathbf{A}$ is symmetric and positive definite. The inverse $(\mathbf{A}^T\mathbf{A})^{-1}$ therefore exists and is bounded.

The data vector $\mathbf{b}$ is constructed from the residuals $\boldsymbol{\delta}_m$, which are bounded by $\|\boldsymbol{\delta}_m\| \leq \theta_{\max} < \pi$. The entries of $\mathbf{A}^\top\mathbf{b}$ are sums of the form:

$$[\mathbf{A}^\top\mathbf{b}]_i = \sum_{m=k-n}^{k+n} \frac{(t_m - t_k)^i}{i!}\boldsymbol{\delta}_m,$$

which are bounded as $\boldsymbol{\delta}_k$, and the time intervals are bounded.

Combining the above, the polynomial coefficients $\boldsymbol{\rho}$ satisfy:

$$\|\boldsymbol{\rho}\| \leq \left\|(\mathbf{A}^\top\mathbf{A})^{-1}\right\| \left\|\mathbf{A}^\top\mathbf{b}\right\|$$

Finally, we note that as long as our learned weight matrix $\mathbf{W}$ is bounded, this also holds for the weighted adaptation. $\mathbf{W} \in \mathbb{R}^{3(2n-1)\times 3(2n-1)}$ is a (similarly expanded with $\otimes\mathbf{I}_3$) diagonal matrix with a weight coinciding to each point in the trajectory window. Then, the bound can be adapted to:

$$\|\boldsymbol{\rho}\| \leq \left\|(\mathbf{A}^\top\mathbf{W}\mathbf{A})^{-1}\right\| \left\|\mathbf{A}^\top\mathbf{W}\mathbf{b}\right\|$$

In practice, we find that we do not need to impose any constraints on $\mathbf{W}$ and that the network learns to construct the regression weights that are well-behaved.

Uniqueness. [10] additionally requires the control signal X to be unique. For regular sampling, this follows as cubic splines pass directly through the control points.

In our case, the polynomial $\mathbf{p}\,(t - t_k; \boldsymbol{\rho}_k)$ does not pass through the control points. However, we observe that because the matrix $\mathbf{A}$ has full column rank, the reduced optimization problem in Thm. (8) is strictly convex, and thus the least squares problem attains a unique solution. In the case of irregular sampling, however, this does not hold [10, B.2.1]. As a simple counterexample, consider that a point placed on the estimated trajectory $\varphi(t)$ would not alter the solution due to the strict convexity of the cost function in Thm (8). In practice, we find that the method also behaves well for irregularly sampled measurements. □

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 2023. 4

[2] Lennart Bastian, Yizheng Xie, Nassir Navab, and Zorah Lähner. Hybrid functional maps for crease-aware non-isometric shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3313–3323, 2024. 8

[3] Ricky T.Q. Chen. torchdiffeq, 2018. 6

[4] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. Blenderproc2: A procedural pipeline for photorealistic rendering. *Journal of Open Source Software*, 8(82):4901, 2023. 2

[5] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. 3

[6] Andreas René Geist, Jonas Frey, Mikel Zhobro, Anna Levina, and Georg Martius. Learning with 3d rotations, a hitchhiker's guide to so (3). In *Int. Conf. Mach. Lear.*, 2024. 6

[7] Maarten Jongeneel and Alessandro Saccon. Geometric savitzky-golay filtering of noisy rotations on so (3) with simultaneous angular velocity and acceleration estimation. In *Int. Conf. Intell. Robot. and Syst.*, pages 2962–2968. IEEE, 2022. 1, 4, 8, 9

[8] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. 33:6696–6707, 2020. 4, 6

[9] Justice J Mason, Christine Allen-Blanchette, Nicholas Zolman, Elizabeth Davison, and Naomi Ehrich Leonard. Learning to predict 3d rotational dynamics from images of a rigid body with unknown mass distribution. *Aerospace*, 10(11):921, 2023. 4, 6, 7, 8

[10] James Morrill, Patrick Kidger, Lingyi Yang, and Terry Lyons. On the choice of interpolation scheme for neural cdes. *Trans. Mach. Lear. Resea.*, 2022(9), 2022. 4, 9, 10

[11] Mikael Persson, Gustav Häger, Hannes Ovrén, and Per-Erik Forssén. Practical pose trajectory splines with explicit regularization. In *Int. Conf. 3D Vision*, pages 156–165. IEEE, 2021. 4

[12] Ramona-Andreea Rohan. Some remarks on the exponential map on the groups so(n) and se(n). In *International Conference on Geometry, Integrability and Quantization*, pages 160–175. Avangard Prima, 2013. 9

[13] Ken Shoemake. Uniform random rotations. In *Graphics Gems III (IBM Version)*, pages 124–132. Elsevier, 1992. 5

[14] Christiane Sommer, Vladyslav Usenko, David Schubert, Nikolaus Demmel, and Daniel Cremers. Efficient derivative computation for cumulative b-splines on lie groups. In *CVPR*, 2020. 4

[15] Hauke Strasdat. Sophus: C++ implementation of lie groups using eigen. https://github.com/strasdat/Sophus, 2011. 4

[16] Vladyslav Usenko, Nikolaus Demmel, and Daniel Cremers. The double sphere camera model. In *2018 International Conference on 3D Vision (3DV)*, pages 552–560. IEEE, 2018. 4

[17] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *CVPR*, pages 16611–16621, 2021. 2

[18] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*. Robotics: Science and Systems Foundation, 2018. 2

[19] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, pages 5745–5753, 2019. 6