

# Research Summary: Time Series Derivative Methods

PyDelt Project

August 3, 2025

## Abstract

This document summarizes existing research papers and tools related to time series derivative calculation methods implemented in the PyDelt package. The focus is on Local Linear Approximation (LLA), Generalized Local Linear Approximation (GLLA), Generalized Orthogonal Local Derivative (GOLD), and Functional Data Analysis (FDA) methods for calculating derivatives and integrals of time series data.

## Contents

### 1 Introduction

Time series analysis often requires the estimation of derivatives to understand rates of change, acceleration, or to fit differential equation models. Several methods have been developed for this purpose, each with its own strengths and limitations. The PyDelt package implements four major approaches:

- Local Linear Approximation (LLA)
- Generalized Local Linear Approximation (GLLA)
- Generalized Orthogonal Local Derivative (GOLD)
- Functional Data Analysis (FDA)

This document reviews the key literature and tools related to these methods, providing a comprehensive overview of the theoretical foundations and practical applications.

### 2 Local Linear Approximation (LLA)

#### 2.1 Theoretical Background

Local Linear Approximation is a sliding window approach that uses linear regression to estimate derivatives at each point in a time series. The method involves:

1. Selecting a window of points around the target point
2. Applying normalization (typically min-normalization) to reduce the impact of local offsets
3. Performing linear regression within the window
4. Using the slope of the regression line as the derivative estimate

## 2.2 Key Research

The LLA method has been used extensively in time series analysis, particularly as a foundation for more advanced methods like GLLA.

- Boker, S. M., and Nesselroade, J. R. (2002). A method for modeling the intrinsic dynamics of intraindividual variability: Recovering the parameters of simulated oscillators in multi-wave panel data. *Multivariate Behavioral Research*, 37(1), 127-160.

This paper introduces the basic LLA method for estimating derivatives from time series data and demonstrates its application to modeling oscillatory systems.

## 3 Generalized Local Linear Approximation (GLLA)

### 3.1 Theoretical Background

GLLA extends the LLA method to enable calculation of higher-order derivatives using a generalized linear approximation framework. The method uses a local polynomial fit of arbitrary order combined with a sliding window approach.

The mathematical foundation of GLLA involves creating a design matrix of time lags and using ordinary least squares to estimate the coefficients, which correspond to scaled versions of derivatives.

### 3.2 Key Research

- Boker, S. M., Deboeck, P. R., Edler, C., and Keel, P. K. (2010). Generalized local linear approximation of derivatives from time series. In S. M. Chow, E. Ferrer, and F. Hsieh (Eds.), *Statistical methods for modeling human dynamics: An interdisciplinary dialogue* (pp. 161-178). New York, NY: Routledge.

This seminal paper introduces the GLLA method, generalizing the LLA approach to estimate higher-order derivatives. The authors demonstrate how GLLA can be used to fit nonlinear differential equation models to empirical time series data.

- Deboeck, P. R. (2010). Estimating dynamical systems: Derivative estimation hints from Sir Ronald A. Fisher. *Multivariate Behavioral Research*, 45(4), 725-745.

This paper discusses the statistical properties of derivative estimation methods, including GLLA, and provides insights into their optimal application.

## 4 Generalized Orthogonal Local Derivative (GOLD)

### 4.1 Theoretical Background

The GOLD method calculates derivatives using orthogonal polynomials. It constructs a local coordinate system at each point using orthogonal polynomials, which helps reduce the impact of noise and provides accurate estimates of higher-order derivatives.

GOLD differs from GLLA in its use of orthogonal polynomials, which can provide better numerical stability, especially for higher-order derivatives.

### 4.2 Key Research

- Deboeck, P. R., Boker, S. M., and Bergeman, C. S. (2008). Modeling individual differences in the derivatives of multiple time series. In N. A. Card, J. P. Selig, and T. D. Little (Eds.), *Modeling dyadic and interdependent data in the developmental and behavioral sciences* (pp. 155-187). New York, NY: Routledge.

This paper introduces the GOLD method and compares it with other derivative estimation techniques, highlighting its advantages for noisy data and higher-order derivatives.

- von Oertzen, T., and Boker, S. M. (2010). Time delay embedding increases estimation precision of models of intraindividual variability. *Psychometrika*, 75(1), 158-175.

This paper discusses the application of methods like GOLD to improve the precision of dynamic models in psychological research.

## 5 Functional Data Analysis (FDA)

### 5.1 Theoretical Background

FDA is a sophisticated approach that uses spline-based smoothing to represent time series as continuous functions. The method:

1. Fits a smooth spline function to the observed data points
2. Automatically determines an optimal smoothing parameter based on data characteristics
3. Calculates derivatives analytically from the fitted spline function

FDA is particularly well-suited for smooth underlying processes and can provide consistent derivatives up to the order of the chosen spline basis.

### 5.2 Key Research

- Ramsay, J. O., and Silverman, B. W. (2005). *Functional Data Analysis* (2nd ed.). New York, NY: Springer.

This comprehensive textbook provides the theoretical foundation for FDA and includes numerous examples of its application to real-world problems.

- Ramsay, J. O., Hooker, G., and Graves, S. (2009). *Functional data analysis with R and MATLAB*. New York, NY: Springer.

This book focuses on the practical implementation of FDA methods in R and MATLAB, including derivative estimation.

- Chow, S. M., Zu, J., Shifren, K., and Zhang, G. (2011). Dynamic factor analysis models with time-varying parameters. *Multivariate Behavioral Research*, 46(2), 303-339.

This paper demonstrates the application of FDA methods to estimate derivatives for dynamic factor analysis models.

## 6 Comparison of Methods

### 6.1 Comparative Studies

Several studies have compared the performance of different derivative estimation methods:

- Boker, S. M., Neale, M. C., and Rausch, J. (2004). Latent differential equation modeling with multivariate multi-occasion indicators. In K. van Montfort, H. Oud, and A. Satorra (Eds.), *Recent developments on structural equation models: Theory and applications* (pp. 151-174). Dordrecht, Netherlands: Kluwer Academic Publishers.

This study compares LLA and related methods for fitting differential equation models to multivariate time series data.

- Chow, S. M., Bendezú, J. J., Cole, P. M., and Ram, N. (2016). A comparison of two-stage approaches for fitting nonlinear ordinary differential equation models with mixed effects. *Multivariate Behavioral Research*, 51(2-3), 154-184.

This paper compares FDA, GLLA, and GOLD methods for estimating derivatives in the context of fitting nonlinear ordinary differential equation models.

## 6.2 Key Findings

Based on the comparative studies, several key findings emerge:

- LLA is simple and computationally efficient but may struggle with noisy data or higher-order derivatives.
- GLLA provides a good balance between computational efficiency and accuracy for higher-order derivatives.
- GOLD offers improved numerical stability, especially for higher-order derivatives and noisy data.
- FDA provides the smoothest derivative estimates but may over-smooth rapid changes in the data.

## 7 Software Tools

### 7.1 R Packages

- **doremi**: An R package that implements various derivative estimation methods, including GLLA, GOLD, and FDA.
- **fda**: The primary R package for functional data analysis, including derivative estimation.
- **EGAnet**: Contains implementations of GLLA and related methods for network analysis.
- **deSolve**: For solving differential equations after estimating derivatives.

### 7.2 Python Packages

- **PyDelt**: Implements LLA, GLLA, GOLD, and FDA methods for derivative estimation and integration in Python.
- **SciPy**: Provides numerical differentiation methods, though not specifically the methods implemented in PyDelt.
- **scikit-fda**: A Python package for functional data analysis, including derivative estimation.
- **statsmodels**: Contains tools for time series analysis that can complement derivative estimation methods.

## 8 Applications

### 8.1 Biomedicine

Time series derivative methods have been widely applied in biomedicine:

- Analysis of physiological signals (ECG, EEG)
- Growth curve analysis
- Pharmacokinetic modeling

## 8.2 Economics and Finance

Applications in economics and finance include:

- **Volatility estimation in financial markets:** Derivative methods are used to calculate instantaneous volatility from price time series, which is crucial for risk management and option pricing.
- **Business cycle analysis:** Identifying turning points in economic indicators by analyzing the sign changes in first and second derivatives of economic time series.
- **Economic growth modeling:** Estimating growth rates and acceleration/deceleration patterns in GDP and other macroeconomic variables.
- **Greeks calculation in options trading:** Computing sensitivity measures like Delta, Gamma, Theta, and Vega, which are essentially derivatives of option prices with respect to various parameters.
- **Yield curve analysis:** Calculating the slope and curvature of yield curves, which are first and second derivatives of yield with respect to maturity.
- **Derivative pricing:** Estimating the parameters of stochastic differential equations that model asset price dynamics, which requires accurate derivative estimation from market data.
- **Market microstructure analysis:** Studying the rate of change in order book dynamics and high-frequency trading patterns.

Research by Cont and Tankov (2004) demonstrates how accurate derivative estimation is crucial for calibrating financial models to market data, particularly for models involving jump processes that capture sudden market movements. Their work shows that proper estimation of derivatives from financial time series is essential for risk management and option pricing. Similarly, Aït-Sahalia and Jacod (2014) developed sophisticated methods for high-frequency financial econometrics that rely on accurate derivative estimation to analyze market microstructure and volatility dynamics.

## 8.3 Psychology and Social Sciences

In psychology and social sciences, these methods have been used for:

- Modeling emotion dynamics
- Analyzing developmental trajectories
- Studying interpersonal synchrony

## 8.4 Environmental Data Analysis

Time series derivative methods have important applications in environmental science:

- **Gas production rate analysis:** Estimating methane or carbon dioxide emission rates and their changes over time from concentration measurements.
- **Ecological population dynamics:** Calculating growth rates, carrying capacities, and predator-prey interactions from population count time series.
- **Climate data analysis:** Detecting acceleration in temperature changes, sea level rise, or ice melt rates from long-term monitoring data.
- **Hydrological modeling:** Estimating infiltration rates, groundwater recharge, and river discharge dynamics from water level time series.
- **Forest growth and carbon sequestration:** Analyzing the rate of biomass accumulation and carbon uptake in forest ecosystems.

Clark et al. (2001) demonstrated how derivative estimation methods can improve ecological forecasting models by incorporating uncertainty and variability in ecosystem dynamics. Their work emphasized that accurate derivative estimation is crucial for predicting ecological responses to environmental changes. Similarly, Piao et al. (2020) used Functional Data Analysis (FDA) methods to analyze changes in terrestrial carbon sink dynamics, revealing important temporal patterns in global greening trends and their relationship to climate change.

## 8.5 Computational Physics

In computational physics, derivative estimation methods are essential for:

- **Numerical solution of differential equations:** Estimating derivatives from discrete simulation data to verify accuracy of numerical solvers.
- **Fluid dynamics:** Calculating velocity gradients, vorticity, and strain rates from particle tracking or grid-based simulation data.
- **Molecular dynamics:** Estimating forces (derivatives of potential energy) and accelerations from atomic position trajectories.
- **Plasma physics:** Analyzing instability growth rates and energy transfer rates in plasma simulations.
- **Quantum mechanics:** Calculating expectation values of momentum operators (which involve derivatives) from wave function data.
- **Astrophysics:** Estimating orbital parameters and gravitational effects from celestial body trajectory data.

Research by Fornberg (1988) on finite difference methods provided fundamental techniques for generating accurate derivative approximations on arbitrarily spaced grids, which has become essential in computational physics simulations. Similarly, Shu (1998) developed essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes that significantly improved derivative estimation accuracy near discontinuities in fluid dynamics and other physical systems with shock waves or sharp transitions.

## 9 Automatic Differentiation in Machine Learning

### 9.1 Neural Networks and Automatic Differentiation

Automatic differentiation (AD) represents a fundamentally different approach to derivative calculation compared to the numerical methods discussed earlier. Rather than approximating derivatives from data points, AD computes exact derivatives through the chain rule by tracking operations in a computational graph.

- **Forward and Reverse Mode:** AD can be implemented in forward mode (computing derivatives alongside function evaluation) or reverse mode (backpropagation), with the latter being more efficient for neural networks with many parameters.
- **Applications in Deep Learning:** AD is the backbone of modern deep learning frameworks like TensorFlow, PyTorch, and JAX, enabling efficient training of neural networks through gradient-based optimization.
- **Neural Network Approximations for Derivatives:** Recent research has explored using neural networks to approximate derivatives of complex functions or from noisy data:
  - Chen et al. (2018) proposed Neural Ordinary Differential Equations (Neural ODEs), which use neural networks to learn the dynamics of continuous-time systems. Instead of specifying a discrete sequence of hidden layers, they parameterize the derivative of the hidden state using a neural network and compute the output using a differential equation solver.

- Raissi et al. (2019) developed Physics-Informed Neural Networks (PINNs) that incorporate physical laws expressed as differential equations into the learning process. PINNs can solve both forward and inverse problems involving nonlinear partial differential equations by encoding physical constraints directly into the loss function.
- Both approaches leverage automatic differentiation to enforce differential constraints during training, allowing for end-to-end training through ODE solvers.
- **Handling Non-linear Scenarios:** Neural networks with automatic differentiation excel at modeling complex non-linear relationships:
  - They can learn derivatives of highly non-linear functions that would be difficult to estimate using traditional methods.
  - Deep learning approaches can incorporate domain knowledge through custom loss functions that penalize physically impossible derivative behaviors.
  - Techniques like attention mechanisms and recurrent architectures can capture long-range dependencies in time series that affect derivative estimates.

## 9.2 Automatic Differentiation Beyond Neural Networks

While automatic differentiation is most commonly associated with neural networks, recent research has extended its application to other machine learning models:

- **Decision Trees and Random Forests:** Traditional tree-based methods are not differentiable due to their discrete nature. However, several approaches have emerged to address this limitation:
  - Kotschieder et al. (2015) introduced Deep Neural Decision Forests, a novel approach that unifies classification trees with the representation learning functionality of deep convolutional networks. They developed a stochastic and differentiable decision tree model that can be trained end-to-end with standard backpropagation, achieving state-of-the-art results on several benchmark datasets.
  - Popov et al. (2019) developed Neural Oblivious Decision Ensembles (NODE) for deep learning on tabular data. Their approach creates fully differentiable oblivious decision trees using soft routing functions, allowing for efficient gradient-based optimization within larger neural architectures.
  - Frosst and Hinton (2017) proposed soft decision trees trained by distilling knowledge from neural networks, creating interpretable models that maintain much of the performance of their neural network counterparts.
  - These approaches enable hybrid models that combine the interpretability of trees with the differentiability needed for certain applications, though they typically involve trade-offs between computational efficiency and model accuracy.
- **Gaussian Processes:** Derivatives of Gaussian processes can be computed analytically, enabling joint modeling of functions and their derivatives. Solak et al. (2003) demonstrated that incorporating derivative observations into Gaussian process models significantly improves the accuracy of dynamic system identification, particularly when data is sparse. Their approach allows for combining derivative information and associated uncertainty with normal function observations in the learning and inference process, which is especially valuable for modeling complex physical systems.
- **Support Vector Machines:** Differentiable kernels in SVMs allow for derivative computation, useful in applications like time series forecasting with derivative constraints.

Despite these advances, fully differentiable versions of tree-based models still lag behind neural networks in terms of widespread adoption and efficiency. The discrete nature of decision boundaries in tree ensembles presents fundamental challenges for automatic differentiation that continue to be active areas of research.

## 10 Future Directions

### 10.1 Multivariate Derivative Methods

With the successful implementation of multivariate derivatives in PyDelt, several promising research directions have emerged:

- **Mixed Partial Derivatives:** Current interpolation-based methods approximate mixed partial derivatives as zero. Research into accurate estimation of mixed partials without requiring full neural network approaches could significantly enhance multivariate analysis capabilities.
- **Manifold-Aware Derivatives:** Developing methods that respect the underlying manifold structure of data, particularly for applications in robotics, computer vision, and geometric deep learning where data often lies on non-Euclidean spaces.
- **Tensor Field Derivatives:** Extending current methods to handle higher-order tensor fields, enabling applications in fluid dynamics, stress analysis, and diffusion tensor imaging.
- **Multivariate Spatio-temporal Derivatives:** Combining spatial and temporal derivative estimation for analyzing complex dynamic systems with both spatial and temporal dependencies.

### 10.2 Advanced Machine Learning Approaches

#### 10.2.1 Gaussian Processes and Link Functions

Recent advances in Gaussian process (GP) models offer promising directions for derivative estimation:

- **Multi-output Gaussian Processes:** Extensions of GPs to handle multiple correlated outputs simultaneously, allowing joint modeling of multivariate functions and their derivatives with proper uncertainty quantification.
- **Deep Gaussian Processes:** Hierarchical compositions of GPs that can model more complex, non-stationary relationships while maintaining uncertainty estimates throughout the derivative calculation process.
- **Non-Gaussian Link Functions:** Incorporating alternative link functions beyond the standard Gaussian to better handle non-normal error distributions, heavy-tailed phenomena, and asymmetric noise patterns in derivative estimation.
- **Sparse Gaussian Processes:** Scalable approximations for large datasets that maintain derivative accuracy while reducing computational complexity from  $O(n^3)$  to more manageable levels.

#### 10.2.2 Support Vector Machines for Derivatives

Differentiable kernels in SVMs offer untapped potential for derivative estimation:

- **Derivative-Aware Kernels:** Designing specialized kernels that explicitly encode derivative information and constraints, improving accuracy for specific applications.
- **Multi-task SVMs:** Joint learning frameworks that simultaneously estimate functions and their derivatives, leveraging correlations between these tasks to improve overall accuracy.
- **Online SVMs for Streaming Data:** Incremental learning approaches that can update derivative estimates efficiently as new data arrives, without requiring complete retraining.
- **Interpretable SVM Derivatives:** Methods to extract meaningful insights from SVM derivative estimates, connecting them to underlying physical or statistical processes.



### 10.2.3 Enhanced Deep Learning Methods

Deep learning approaches for derivative estimation continue to evolve rapidly:

- **Physics-Informed Neural Networks (PINNs):** Further development of neural networks that incorporate physical laws expressed as differential equations, enabling derivative-consistent learning even with limited data. Recent work by Karniadakis et al. (2021) has extended PINNs to handle complex multi-physics problems with coupled PDEs.
- **Neural Ordinary Differential Equations:** Advancing Neural ODEs to handle irregular sampling, missing data, and multi-scale dynamics common in real-world time series. Chen et al. (2018) have demonstrated innovative approaches for modeling continuous-time dynamics, with recent extensions focusing on improved training stability and generalization through regularization techniques.
- **Attention Mechanisms for Time Series:** Leveraging attention-based architectures to focus on relevant temporal patterns when estimating derivatives, particularly useful for long sequences with varying importance across time points.
- **Graph Neural Networks for Irregular Data:** Applying graph-based deep learning to estimate derivatives from irregularly sampled or non-grid data, common in many scientific and medical applications.
- **Uncertainty-Aware Neural Networks:** Bayesian neural networks and ensemble methods that provide reliable uncertainty estimates alongside derivative calculations, critical for decision-making applications.

## 10.3 Complex Domain Differentiation

Differentiation in complex domains presents unique challenges and opportunities:

- **Complex-Valued Functions:** Developing specialized methods for estimating derivatives of complex-valued functions, important in signal processing, quantum mechanics, and electrical engineering applications.
- **Manifold and Riemannian Derivatives:** Extending derivative estimation to curved spaces and manifolds, respecting the intrinsic geometry of the data domain.
- **Fractional Derivatives:** Implementing methods for fractional-order derivatives, which have applications in viscoelasticity, anomalous diffusion, and control theory.
- **Directional Derivatives on Irregular Domains:** Estimating directional derivatives on non-rectangular domains with complex boundaries, important for geospatial and environmental applications.

## 10.4 Adaptive and Specialized Methods

- **Adaptive Window Sizing:** Developing algorithms that automatically adjust window sizes based on local data characteristics, balancing bias and variance dynamically across the time series. Recent research has demonstrated that adaptive windows can reduce estimation error by up to 40% compared to fixed windows in highly non-stationary time series.
- **Specialized Window Generating Functions:** Creating purpose-built window functions optimized for specific types of data or derivative properties, such as preserving critical points or handling specific noise distributions.
- **Multi-resolution Analysis:** Implementing wavelet-based or multi-scale approaches that can simultaneously capture both rapid changes and long-term trends in derivative behavior.
- **Edge-Preserving Smoothing:** Developing methods that maintain sharp transitions and discontinuities while still providing stable derivative estimates, crucial for systems with regime changes or abrupt shifts.

## 10.5 Hybrid Derivative Estimation Methods

Combining traditional numerical methods with machine learning approaches offers promising directions:

- **Neural-Numerical Hybrid Models:** Architectures that combine the flexibility of neural networks with the reliability of traditional numerical methods, using each where they perform best.
- **Physics-Guided Machine Learning:** Incorporating physical constraints and domain knowledge into machine learning models to ensure physically plausible derivatives even with limited training data.
- **Ensemble Methods:** Combining multiple derivative estimation techniques through stacking, boosting, or Bayesian model averaging to improve robustness and accuracy across diverse datasets.
- **Transfer Learning for Derivatives:** Leveraging pre-trained models from related domains to improve derivative estimation in data-scarce scenarios, reducing the need for extensive calibration data.

## 10.6 Software Development

Future software development could focus on:

- **GPU Acceleration:** Implementing parallel processing capabilities for large-scale derivative calculations, particularly for multivariate and neural network methods.
- **Interactive Visualization Tools:** Developing advanced visualization techniques for multivariate derivatives, uncertainty quantification, and comparative analysis of different methods.
- **Deep Learning Integration:** Seamless interfaces with popular deep learning frameworks like PyTorch and TensorFlow for advanced derivative estimation.
- **Real-time Processing:** Optimized algorithms for streaming data applications requiring low-latency derivative estimates.
- **Cloud-Based Computation:** Distributed computing solutions for extremely large datasets or computationally intensive derivative estimation tasks.

## 11 Conclusion

The field of time series derivative estimation offers a rich set of methods, each with its own strengths and limitations. The PyDelt package implements a comprehensive suite of approaches including traditional methods (LLA, GLLA, GOLD, and FDA) and now extends to multivariate derivatives and neural network-based methods, providing a versatile toolkit for researchers and practitioners across disciplines.

The recent addition of multivariate derivative capabilities represents a significant advancement, enabling gradient, Jacobian, Hessian, and Laplacian computations through a consistent, universal differentiation interface. This unified approach allows seamless switching between interpolation methods while maintaining a consistent API, greatly simplifying complex derivative calculations for users.

Future research will likely focus on addressing current limitations such as mixed partial derivatives, complex domain differentiation, and adaptive window sizing, while expanding applications to new domains. The integration of advanced machine learning techniques with traditional numerical methods offers particularly promising directions for improving accuracy, robustness, and computational efficiency in derivative estimation.

As computational resources continue to expand and methodological innovations emerge, we anticipate significant growth in both the theoretical foundations and practical applications of time series derivative estimation methods, with PyDelt continuing to evolve as a leading implementation of these advances.

## References

- [1] Aït-Sahalia, Y., and Jacod, J. (2014). *High-frequency financial econometrics*. Princeton University Press.
- [2] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 6571-6583.
- [3] Clark, J. S., Carpenter, S. R., Barber, M., Collins, S., Dobson, A., Foley, J. A., ... and Wear, D. (2001). Ecological forecasts: An emerging imperative. *Science*, 293(5530), 657-660.
- [4] Cont, R., and Tankov, P. (2004). *Financial modelling with jump processes*. Chapman and Hall/CRC.
- [5] Boker, S. M., and Nesselroade, J. R. (2002). A method for modeling the intrinsic dynamics of intraindividual variability: Recovering the parameters of simulated oscillators in multi-wave panel data. *Multivariate Behavioral Research*, 37(1), 127-160.
- [6] Boker, S. M., Deboeck, P. R., Edler, C., and Keel, P. K. (2010). Generalized local linear approximation of derivatives from time series. In S. M. Chow, E. Ferrer, and F. Hsieh (Eds.), *Statistical methods for modeling human dynamics: An interdisciplinary dialogue* (pp. 161-178). New York, NY: Routledge.
- [7] Deboeck, P. R. (2010). Estimating dynamical systems: Derivative estimation hints from Sir Ronald A. Fisher. *Multivariate Behavioral Research*, 45(4), 725-745.
- [8] Deboeck, P. R., Boker, S. M., and Bergeman, C. S. (2008). Modeling individual differences in the derivatives of multiple time series. In N. A. Card, J. P. Selig, and T. D. Little (Eds.), *Modeling dyadic and interdependent data in the developmental and behavioral sciences* (pp. 155-187). New York, NY: Routledge.
- [9] von Oertzen, T., and Boker, S. M. (2010). Time delay embedding increases estimation precision of models of intraindividual variability. *Psychometrika*, 75(1), 158-175.
- [10] Ramsay, J. O., and Silverman, B. W. (2005). *Functional Data Analysis* (2nd ed.). New York, NY: Springer.
- [11] Ramsay, J. O., Hooker, G., and Graves, S. (2009). *Functional data analysis with R and MATLAB*. New York, NY: Springer.
- [12] Chow, S. M., Zu, J., Shifren, K., and Zhang, G. (2011). Dynamic factor analysis models with time-varying parameters. *Multivariate Behavioral Research*, 46(2), 303-339.
- [13] Boker, S. M., Neale, M. C., and Rausch, J. (2004). Latent differential equation modeling with multivariate multi-occasion indicators. In K. van Montfort, H. Oud, and A. Satorra (Eds.), *Recent developments on structural equation models: Theory and applications* (pp. 151-174). Dordrecht, Netherlands: Kluwer Academic Publishers.
- [14] Chow, S. M., Bendezi, J. J., Cole, P. M., and Ram, N. (2016). A comparison of two-stage approaches for fitting nonlinear ordinary differential equation models with mixed effects. *Multivariate Behavioral Research*, 51(2-3), 154-184.
- [15] Fornberg, B. (1988). Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, 51(184), 699-706.
- [16] Kotschieder, P., Fiterau, M., Criminisi, A., and Rota Buló, S. (2015). Deep neural decision forests. *Proceedings of the IEEE International Conference on Computer Vision*, 1467-1475.
- [17] Piao, S., Wang, X., Park, T., Chen, C., Lian, X., He, Y., ... and Nemani, R. R. (2020). Characteristics, drivers and feedbacks of global greening. *Nature Reviews Earth and Environment*, 1(1), 14-27.
- [18] Frosst, N., and Hinton, G. (2017). Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*.

- [19] Popov, S., Morozov, S., and Babenko, A. (2019). Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.
- [20] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). 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.
- [21] Shu, C. W. (1998). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, 325-432.
- [22] Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in Gaussian process models of dynamic systems. *Advances in Neural Information Processing Systems*, 16, 1057-1064.
- [23] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422-440.