Carnegie Mellon University
Department of Statistics & Data Science

Thesis Proposal

Draft as of February 27, 2026

Advances in Time Series Analysis: Simulation-based Approaches to Estimation, Testing for Structure, and Prediction with Transformers

Michael Wieck-Sosa

Date of Proposal (TBD)

Committee:
Jon Snow, Advisor
Florence Nightingale, King's College, London
Henry Scheffé
John Tukey

Abstract

Technological advances have enabled the collection of large-scale time series data in many fields. Traditional time series methods often struggle to capture the complex dynamics present in such data because of their restrictive assumptions, such as stationarity, linear dynamics, or Gaussian noise. As a result, there is growing demand for more flexible time series methods across economics, finance, epidemiology, and the Earth sciences. We therefore propose to study topics at the intersection of time series analysis and machine learning.

In the first part of this thesis proposal, we introduce a bootstrap-based framework for conditional independence testing with nonstationary time series that utilizes nonlinear regression. In the second part, we develop a simulation-based parameter estimation framework for time series models. In the third part, we explore goodness-of-fit testing and confidence set construction via simulation. In the fourth part, we develop a simulation-and-regression framework for forecasting and smoothing with transformers.

Contents

1 Introduction

1.1 Thesis Agenda

We provide a concise overview of the organization of this thesis proposal, along with some motivations for each topic.

I. Conditional Independence Testing (Section 2). Given nonstationary time series $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, we develop a nonlinear regression-based test of the null hypothesis that X_t is conditionally independent of Y_t given Z_t at all times t, with any desired leads and lags incorporated into these variables. Tests of conditional independence are used in many problems, such as variable selection, causal discovery, the validation of theory-based causal graphs, and the evaluation of predictor fairness; see Hardt et al. (2016).

II. Simulation-based Estimation (Section 3). Next, we introduce a simulation-based framework for estimating the unknown parameter in a parametric statistical model for a time series $X_{1:n}$. Such methods are desirable for models where the likelihood is analytically or computationally intractable, but simulating from the model is easy. For example, nonlinear state-space models, discrete-time dynamical systems and ODEs with observational noise, and discretized Lévy-driven SDEs.

III. Simulation-based Inference (Section 4). We explore fundamental topics in simulation-based inference for nonstationary time series: goodness-of-fit testing and confidence set construction. The aim of goodness-of-fit testing is to test whether the distribution of the observed time series lies in some model class, while the goal of confidence set construction is to quantify uncertainty about the true parameter.

IV. Prediction with Transformers (Section 5). Lastly, we develop a simulation-and-regression framework for training transformers for the tasks of forecasting and smoothing. Forecasting aims to characterize the future values of a stochastic process given the current information. It underpins monetary policy decisions in economics, portfolio optimization in finance, public health policy decisions during epidemics, and early warning systems for extreme weather events. Smoothing aims to characterize the past values of a latent stochastic process given the current information. Applications include estimating sentiment in the social sciences, volatility in finance, notions of infectivity in epidemiology, and data assimilation in the Earth Sciences.

2 Conditional Independence Testing

2.1 Overview

In this part of the thesis, we introduce a framework for conditional independence testing with nonstationary time series. To the best of our knowledge, this is the first conditional independence testing framework for nonstationary time series outside of the linear-Gaussian setting. In particular, we use a theoretical framework for nonstationary time series based on the foundational work of Wu (2005); Zhou and Wu (2009). We propose a bootstrap-based testing procedure, which we justify with a distribution-uniform version of the strong Gaussian approximation from Mies and Steland (2023). We prove that our tests have Type I error control, *uniformly* over a large collection of null distributions. The main ideas are presented here, and we refer readers to the paper (Wieck-Sosa et al., 2025) for more details.

2.2 Setting

Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$, respectively, for some $n, d_X, d_Y, d_Z \in \mathbb{N}$. To simplify the notation, let these variables incorporate any desired leads and lags. The only requirement is that the conditioning set Z_t is known at time t. In this section, we present a test for the null hypothesis that

$$X_t \perp\!\!\!\perp Y_t \mid Z_t \text{ for all } t \in [n]. \tag{1}$$

We allow the time series to have nonlinear temporal dynamics and general forms of nonstationarity. The paper (Wieck-Sosa et al., 2025) includes the details about the asymptotic framework, the assumptions about the nonstationarity and decay of temporal dependence using the physical dependence measure of Wu (2005), and the rates at which the number of dimensions and time-offsets can grow with n.

2.3 Time-varying Regression and Error Processes

We can always decompose

$$X_t = f_t(Z_t) + \varepsilon_t, \quad Y_t = g_t(Z_t) + \xi_t, \tag{2}$$

where $f_t(z) = \mathbb{E}(X_t \mid Z_t = z)$ and $g_t(z) = \mathbb{E}(Y_t \mid Z_t = z)$ are the regression functions, which may vary over time. The error processes $\varepsilon_{1:n}$ and $\xi_{1:n}$ can

also be nonstationary and temporally dependent. Denote the vectorized outer product of the errors at time t by

$$R_t = \text{vec}(\varepsilon_t \xi_t^\top). \tag{3}$$

Next, let $\hat{f}_t$ and $\hat{g}_t$ be estimates of f_t and g_t, respectively, and let

$$\hat{\varepsilon}_t = X_t - \hat{f}_t(Z_t), \quad \hat{\xi}_t = Y_t - \hat{g}_t(Z_t), \tag{4}$$

be the corresponding residuals. Denote the vectorized outer product of these residuals at time t by

$$\hat{R}_t = \text{vec}(\hat{\varepsilon}_t \hat{\xi}_t^\top). \tag{5}$$

Crucially, our testing framework can be used with *any regression estimator*. The required rates of convergence for the regression estimators are discussed in the paper (Wieck-Sosa et al., 2025). In our experiments, we use the sieve estimator from Ding and Zhou (2021), though there are many other regression estimators for nonstationary time series to choose from (Vogt, 2012; Zhang and Wu, 2015; Yousuf and Ng, 2021; Chen et al., 2022; Kurisu et al., 2025). Our test possesses a property known as *rate double robustness*, which means that we only require modest convergence rates for the products of the estimation errors, rather than for each estimation error individually.

2.4 Testing Procedure

Our proposed testing procedure can be viewed as an extension of the *generalized covariance measure* (GCM) test from Shah and Peters (2020) from the iid setting to the nonstationary time series setting, so we refer to it as the *dynamic generalized covariance measure* (dGCM) test. The core idea is that, under the null hypothesis of conditional independence from (1), the covariance of the errors from (2) will be zero at all times. Note that these covariances can be zero at all times, even under alternatives in which the corresponding conditional dependencies always hold. Consequently, we can only hope to have power against alternatives in which these covariances are non-zero for at least *some* times.

Roughly speaking, our test statistic uses the empirical covariances among the residuals from (4) to try to detect non-zero covariances among the errors from (2), for at least *some* times. To estimate the desired quantile of the test statistic, we use a bootstrap procedure justified by a *distribution-uniform* version of the strong Gaussian approximation from Mies and Steland (2023) applied to the process of error products R_t from (3). This is possible because, under the null hypothesis from (1), the error products R_t from (3) will

have mean zero at all times t. The approximating Gaussian process has a time-varying covariance structure, which we estimate using a rolling-window approach; see the paper (Wieck-Sosa et al., 2025) for how to select the window size. We prove that the following procedure has Type I error control, uniformly over a large collection of null distributions.

Dynamic Generalized Covariance Measure (dGCM) Test.

- **Inputs:** Time series $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, significance level α, number of simulations s, $p \in [2, \infty]$ for norm in test statistic, and method for selecting the window size L.
- Regress X_t on Z_t, $t \in [n]$, and obtain residuals $\hat{\varepsilon}_t$, $t \in [n]$.
- Regress Y_t on Z_t, $t \in [n]$, and obtain residuals $\hat{\xi}_t$, $t \in [n]$.
- Obtain residual products $\hat{R}_t = \mathrm{vec}(\hat{\varepsilon}_t \hat{\xi}_t^\top)$, $t \in [n]$.
- Calculate test statistic on time series of residual products

$$T(\hat{R}_{1:n}) = \max_{j \in [n]} \left\| \frac{1}{\sqrt{n}} \sum_{t \leq j} \hat{R}_t \right\|_p .$$

- Select the window size L for covariance estimation using $\hat{R}_{1:n}$.
- For $t \in [n]$, obtain estimates of the time-varying covariances as

$$\hat{\Sigma}_t = \frac{1}{L} \left(\sum_{j=t-L+1}^{t} \hat{R}_j \right) \left(\sum_{j=t-L+1}^{t} \hat{R}_j \right)^\top .$$

- For each $r \in [s]$ and $t \in [n]$, simulate *independent* Gaussians

$$\tilde{R}_t^{(r)} \sim N(0, \hat{\Sigma}_t).$$

- For each $r \in [s]$, calculate test statistic on simulated Gaussian process

$$T(\tilde{R}_{1:n}^{(r)}) = \max_{j \in [n]} \left\| \frac{1}{\sqrt{n}} \sum_{t \leq j} \tilde{R}_t^{(r)} \right\|_p .$$

- Calculate the $1 - \alpha$ empirical quantile $\hat{q}_{1-\alpha}^{\mathrm{boot}}$ of the simulated test statistics $T(\tilde{R}_{1:n}^{(1)}), \ldots, T(\tilde{R}_{1:n}^{(s)})$.
- Reject null hypothesis from (1) at level α if $T(\hat{R}_{1:n}) > \hat{q}_{1-\alpha}^{\mathrm{boot}}$, else retain.

3 Simulation-based Estimation

3.1 Overview

In this part of the thesis, we develop a simulation-based estimation framework for parametric models of stochastic processes. Given an $\mathbb{R}^d$-valued time series $X_{1:n}$ generated by such a model, we aim to estimate the unknown parameter $\theta_0 \in \mathbb{R}^p$ for the model. Unfortunately, the likelihood function is intractable, but simulating from the model with different parameter values is straightforward.

Our core idea is to estimate θ_0 by tuning the parameter values so that *random features* of the simulated data closely match those of the observed data. We use the embedding results of Sauer et al. (1991) from the "geometry from a time series" literature to show that models with a p-dimensional parameter space can typically be identified using only $2p+1$ random features, regardless of the observation dimension. Our approach avoids both user-selected summary statistics and costly neural network-based procedures for learning them.

We introduce two estimators. First, a time-average estimator for processes that are stationary, or at least asymptotically mean stationary. Second, a rolling-window estimator for processes with general forms of nonstationarity. We prove that both estimators are consistent and asymptotically normal under nonrestrictive conditions. Specifically, we use a theoretical framework for nonstationary time series based on Wu (2005); Zhou and Wu (2009); Mies and Steland (2023). Our experiments suggest that simulation-based estimation can be made more robust and automatic by matching random features, with little loss in statistical efficiency.

3.2 Notation and Setting

For real numbers $x, y \in \mathbb{R}$, denote $x \wedge y = \min(x, y)$ and $x \vee y = \max(x, y)$. For a vector $x \in \mathbb{R}^d$, denote the ℓ^p norm by $\|x\|_p$ and the Euclidean norm by $\|x\| = \|x\|_2$. For a random vector X, $\|X\|_{\mathcal{L}^q} = (\mathbb{E}\|X\|^q)^{1/q}$ denotes the $\mathcal{L}^q$ norm of the Euclidean norm. For any natural number $j \in \mathbb{N}$, denote $[j] = \{1, 2, \ldots, j\}$. For a sequence of random vectors X_t, $t = 1, \ldots, n$, each taking values in $\mathbb{R}^d$, we denote a subsequence as $X_{t_1:t_2} = (X_{t_1}, \ldots, X_{t_2})^\top$, which takes values in $\mathbb{R}^{(t_2 - t_1 + 1) \times d}$ for some $t_1, t_2 \in [n]$.

We observe a time series X_t^{obs}, $t = 1, \ldots, n$, taking values in $\mathbb{R}^d$ for some fixed dimension $d \in \mathbb{N}$. The observed time series is assumed to arise from a known generative model with an unknown p-dimensional parameter

$\theta_0 \in \Theta \subset \mathbb{R}^p$, $p \in \mathbb{N}$. Each value of $\theta \in \Theta$ determines a law of a stochastic process. For any value of θ, we can use the generative model to simulate $s \in \mathbb{N}$ realizations of a length n time series $(X^{(r)}_{1:n}(\theta))_{r\in[s]}$. When we do not need to refer to a particular realization, we drop the superscript.

3.3 Random Fourier features

The central idea is that we should estimate the p-dimensional parameter θ_0 by matching the values of a small number of random features of the simulated and observed data. Consider $k = 2p+1$ randomly drawn functions $\varphi_1, \ldots, \varphi_k$ from a distribution D over a suitably nice function class $\mathcal{F}$. The k functions should be sampled independently of one another and of the data. Each function $\varphi_i : \mathbb{R}^{(m+1)\times d} \to \mathbb{R}$ takes as input a length $m+1$ subsequence of the d-dimensional time series, where m is chosen based on the dynamics of the generative model. For instance, if the model is an order m^* Markov process, then set $m = m^*$.

Let $x_1, \ldots, x_{m+1}$ be vectors in $\mathbb{R}^d$, and define $x = (x_1, \ldots, x_{m+1})^\top \in \mathbb{R}^{(m+1)\times d}$. We consider random Fourier features of the form

$$\varphi_i(x) = \cos\left(\sum_{j=1}^{m+1} \Omega_{i,j} \cdot x_j + \alpha_i\right), \tag{6}$$

where $\Omega_{i,j} \overset{\text{iid}}{\sim} N(0, I_d)$ and $\alpha_i \overset{\text{iid}}{\sim} U(-\pi, \pi)$ for $i = 1, \ldots, k$ and $j = 1, \ldots, m+1$. Denote all k functions by $\varphi = (\varphi_1, \ldots, \varphi_k)$, so that $\varphi : \mathbb{R}^{(m+1)\times d} \to \mathbb{R}^k$. Define the i-th random feature of the observed and r-th simulated time series at time $t = m+1, \ldots, n$ as

$$f^{\text{obs}}_{t,i} = \varphi_i(X^{\text{obs}}_{t-m:t}), \quad f^{(r)}_{t,i}(\theta) = \varphi_i\left(X^{(r)}_{t-m:t}(\theta)\right).$$

Write all k random features at time t as

$$f^{\text{obs}}_t = \left(f^{\text{obs}}_{t,1}, \ldots, f^{\text{obs}}_{t,k}\right), \quad f^{(r)}_t(\theta) = \left(f^{(r)}_{t,1}(\theta), \ldots, f^{(r)}_{t,k}(\theta)\right). \tag{7}$$

3.4 Estimators

Time-average estimator. For processes that are asymptotically mean stationary, it suffices to consider the time-average of $k = 2p+1$ random features. Define the observed and simulated time-averages of the random features by

$$F^{\text{obs}} = \frac{1}{n-m}\sum_{t=m+1}^{n} f^{\text{obs}}_t, \quad \bar{F}^{\text{sim}}(\theta) = \frac{1}{n-m}\sum_{t=m+1}^{n} \bar{f}^{\text{sim}}_t(\theta), \tag{8}$$

where $\bar{f}_t^{\text{sim}}(\theta) = \frac{1}{s}\sum_{r=1}^{s} f_t^{(r)}(\theta)$. The time-average estimator is given by

$$\hat{\theta}^{\text{TA}} = \underset{\theta\in\Theta}{\text{argmin}}\ \hat{Q}_n^{\text{TA}}(\theta), \tag{9}$$

where the objective function is defined as

$$\hat{Q}_n^{\text{TA}}(\theta) = \left\| F^{\text{obs}} - \bar{F}^{\text{sim}}(\theta) \right\|^2. \tag{10}$$

Rolling-window estimator. For many nonstationary processes, a different approach is required because the average of the time-varying means of the random features over time may not converge to a limiting mean. Even when it does converge, the limiting mean may not uniquely identify each θ. Thus, we minimize the time-average of the squared distances between rolling-window averages of $k = 2p + 1$ random features.

Define the observed and simulated rolling-window random features at time t by

$$F_t^{\text{obs}} = \frac{1}{w \wedge t}\sum_{j=(t-w)\vee 1}^{t} f_j^{\text{obs}}, \quad \bar{F}_t^{\text{sim}}(\theta) = \frac{1}{w \wedge t}\sum_{j=(t-w)\vee 1}^{t} \bar{f}_j^{\text{sim}}(\theta), \tag{11}$$

where $\bar{f}_t^{\text{sim}}(\theta) = \frac{1}{s}\sum_{r=1}^{s} f_t^{(r)}(\theta)$ and $w \in \mathbb{N}$ is the window size. The rolling-window estimator is given by

$$\hat{\theta}^{\text{RW}} = \underset{\theta\in\Theta}{\text{argmin}}\ \hat{Q}_n^{\text{RW}}(\theta), \tag{12}$$

where the objective function is defined as

$$\hat{Q}_n^{\text{RW}}(\theta) = \frac{1}{n-m}\sum_{t=m+\tau+L}^{n}\left[\left\| F_t^{\text{obs}} - \bar{F}_t^{\text{sim}}(\theta)\right\|^2 + K_t(\theta)\right],$$

$$K_t(\theta) = 2\left(F_{t-L}^{\text{obs}} - \bar{F}_{t-L}^{\text{sim}}(\theta)\right)^{\top}\left(\left[f_t^{\text{obs}} - \bar{f}_t^{\text{sim}}(\theta)\right] - \left[F_{t-L}^{\text{obs}} - \bar{F}_{t-L}^{\text{sim}}(\theta)\right]\right),$$

where $\tau \in \mathbb{N}$ is an initial time-offset and $L \in \mathbb{N}$ is a lag.

In practice, we select the window size, offset, and lag in a similar manner as Mies (2023). The window size w_n is selected as the integer within $[n^{\frac{1}{2}}, n^{\frac{3}{4}}]$ which minimizes the sum of squared distances $\sum_{t=m+1+L}^{n}\left\| F_{t-L}^{\text{obs}} - f_t^{\text{obs}}\right\|^2$, and we set the offset as $\tau_n = w_n$. We let the lag L_n grow slowly with n at the polylogarithmic rate $\lceil \frac{1}{10}\log(n)^2 \rceil$.

Extensions. Several extensions are possible. First, both estimators can be generalized by replacing the Euclidean norm with a weighted norm, using a weight matrix given by the inverse of a suitable estimator of the long-run covariance matrix; see Gourieroux and Monfort (1996). Second, the sample averages used in both estimators can be replaced with different mean estimators. Third, one may extend both estimators to allow for early stopping, so that only a fraction of the time series is used.

3.5 Examples

To illustrate the ideas, let us consider a few examples. The paper contains several more examples. We optimize the objective functions using the differential evolution solver `differential_evolution` from the `scipy.optimize` module in the SciPy Python library, although our experiments indicate that other optimizers perform comparably. For the SIR example, we used the Runge–Kutta 5(4) solver `RK45` from the `scipy.integrate` module in SciPy. Each density plot is constructed from $1,000$ independent estimates of the p-dimensional parameter using $2p + 1$ random Fourier features and 10 simulations per parameter value.

Example 3.1 (Gaussian)**.** For $t = 1, \ldots, n$, we observe

$$X_t = \mu + \epsilon_t,$$

where $\epsilon_t \overset{\text{iid}}{\sim} N(0, 1)$. The unknown parameter is μ, so we use $2p + 1 = 3$ random features.

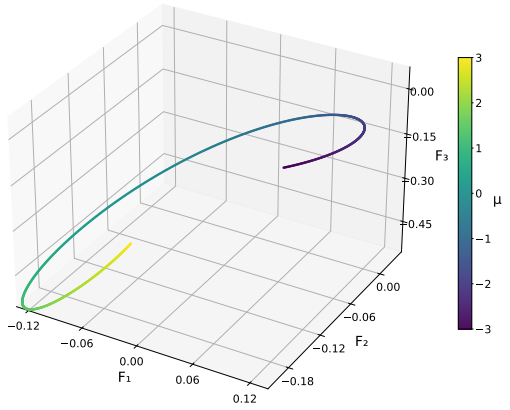
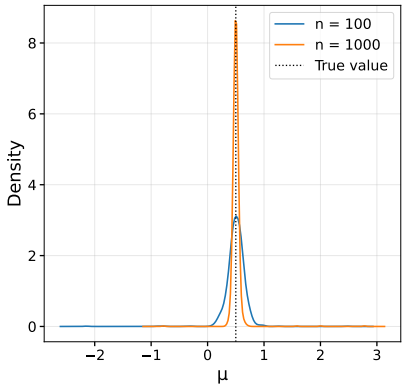


Figure 1: Three random features averaged over $n = 1000$ times and $s = 10$ simulations, for each parameter value on a grid in $[-3, 3]$. Color denotes μ.

We aim to estimate

$$\mu_0 = 0.5,$$

using the time-average estimator.



Example 3.2 (Logistic Map)**.** For $t = 1, \ldots, n$, define the state by

$$Z_t = \rho Z_{t-1}(1 - Z_{t-1}).$$

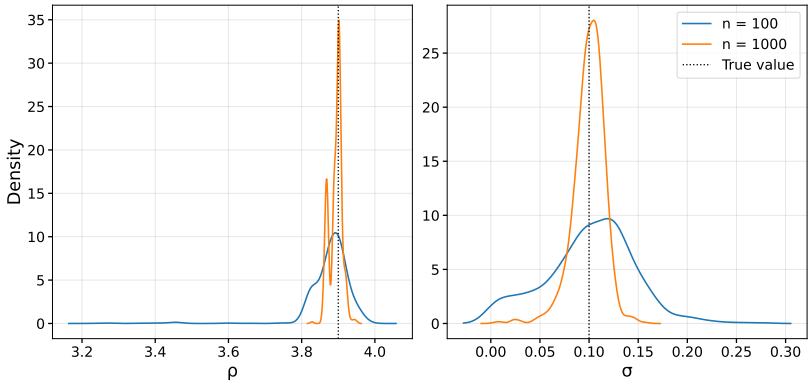
We observe

$$X_t = Z_t + \sigma\epsilon_t,$$

where $\epsilon_t \overset{\text{iid}}{\sim} N(0,1)$ and the unknown initial value Z_0 is sampled from a $U[0,1]$ distribution, so that it is fixed before running the procedure. Our goal is to estimate

$$\rho_0 = 3.9, \quad \sigma_0 = 0.1,$$

using the time-average estimator. We emphasize that the logistic map is in a chaotic regime when $\rho = 3.9$; see Devaney (2018) for more information.



Example 3.3 (SIR model)**.** For time horizon $T = 50$, let $(S_v, I_v, R_v)_{v\in[0,T]}$, be defined by the susceptible-infected-recovered (SIR) system of ODEs

$$\begin{aligned}\frac{\mathrm{d}S_v}{\mathrm{d}v} &= -\frac{\beta}{N} S_v\, I_v,\\ \frac{\mathrm{d}I_v}{\mathrm{d}v} &= \frac{\beta}{N} S_v\, I_v - \gamma I_v,\\ \frac{\mathrm{d}R_v}{\mathrm{d}v} &= \gamma I_v,\end{aligned}$$

where S_v/N, I_v/N, and R_v/N denote the fractions of the population that are susceptible, infected, and recovered (or removed), respectively, $\beta > 0$ is the transmission rate, and $\gamma > 0$ is the recovery rate. Let the known initial values be $S_0 = 980,000$, $I_0 = 20,000$, $R_0 = 0$, where the number of individuals is $N = S_0 + I_0 + R_0 = 1,000,000$.

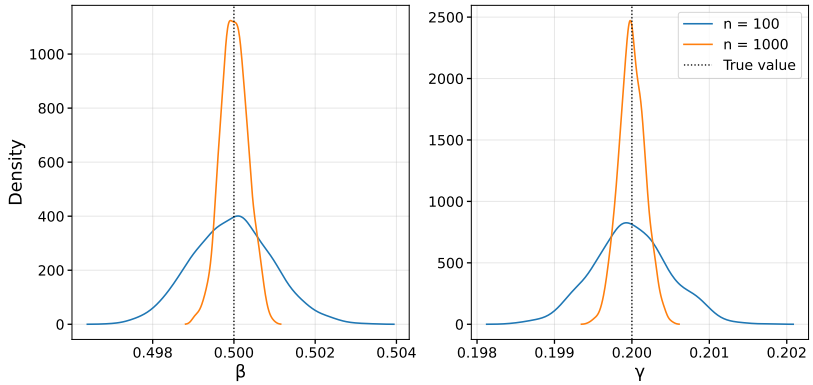
For each $t = 1, \ldots, n$, we test $\mathrm{TE}_t = \lfloor N/100 \rfloor$ people with equal probability using a test with sensitivity $\mathrm{SE}_t = 0.95$ and specificity $\mathrm{SP}_t = 0.99$. We observe the proportion of positive tests

$$X_t = Z_t/\mathrm{TE}_t, \quad Z_t \sim \mathrm{Binomial}\left(\mathrm{TE}_t, \pi_t\right),$$

where $\pi_t = \mathrm{SE}_t I_{Tt/n}/N + (1 - \mathrm{SP}_t)\left(1 - I_{Tt/n}/N\right)$. This model allows the number of tests TE_t, sensitivity SE_t, and specificity SP_t to change with t, although we keep them constant for simplicity. We aim to estimate

$$\beta_0 = 0.5, \quad \gamma_0 = 0.2,$$

using the rolling-window estimator. Note that the bounds for the optimizer were set to $[0.01, 1]$ for β and $[0.01, 0.5]$ for γ.



4 Simulation-based Inference

4.1 Overview

In this part of the thesis, we propose a framework for goodness-of-fit testing and confidence set construction. *Simulation-based inference* (SBI) is a form of statistical inference that uses simulated data generated by a model, for example, to construct confidence sets for the true parameter. SBI is particularly useful in settings where the likelihood is intractable. To give a sense of how SBI works in the context of frequentist inference, we explain how to construct confidence sets using nonparametric regression and quantile regression, as in Dalmasso et al. (2024). We will refer to these procedures when discussing our proposed work in Section 4.2.

Suppose the observed time series $X_{1:n}^{\text{obs}}$ was generated according to a parametric model with true parameter $\theta_0 \in \Theta \subset \mathbb{R}^p$, for some $n, p \in \mathbb{N}$. Denote the indicator function by $\mathbb{I}$. For some test statistic T, let

$$B(\theta, X_{1:n}(\theta), X_{1:n}^{\text{obs}}) = \mathbb{I}\{T(\theta, X_{1:n}(\theta)) \geq T(\theta, X_{1:n}^{\text{obs}})\},$$

so that the p-value for testing the null hypothesis that the true parameter $\theta_0 = \theta$ for some $\theta \in \Theta$ is given by

$$\text{pv}(\theta, X_{1:n}^{\text{obs}}) = \mathbb{E}_\theta[B(\theta, X_{1:n}(\theta), X_{1:n}^{\text{obs}})],$$

where θ and $X_{1:n}^{\text{obs}}$ are treated as fixed, and the expectation is with respect to the distribution $P_{n,\theta}$ of $X_{1:n}(\theta)$. Therefore, an exact $1-\alpha$ confidence set for the true parameter θ_0 is given by

$$C_{1-\alpha}(X_{1:n}^{\text{obs}}) = \{\theta : \text{pv}(\theta, X_{1:n}^{\text{obs}}) \geq \alpha\},$$

so that, for the true parameter $\theta_0 \in \Theta$, we have

$$P_{n,\theta_0}\left(\theta_0 \in C_{1-\alpha}(X_{1:n}^{\text{obs}})\right) = 1 - \alpha.$$

In practice, we estimate the p-value function pv to get an approximate $1-\alpha$ confidence set using the following procedure. Let π be a reference distribution that has full support on the parameter space Θ.

Frequentist Confidence Sets using Nonparametric Regression.

- **Inputs:** Observations $X_{1:n}^{\text{obs}}$, model, reference distribution π, test statistic T, number of simulations s, and nonparametric regression method.

- For each $r \in [s]$, draw $\theta^{(r)} \sim \pi$ and simulate from the model,

$$X_{1:n}^{(r)} \sim P_{n,\theta^{(r)}},$$

  and obtain $B^{(r)} = \mathbb{I}\{T(\theta^{(r)}, X_{1:n}^{(r)}) \geq T(\theta^{(r)}, X_{1:n}^{\text{obs}})\}$.

- Regress $B^{(r)}$ on $\theta^{(r)}$, $r \in [s]$, to get an estimate $\hat{\text{pv}}(\theta, X_{1:n}^{\text{obs}})$ of the p-value function $\text{pv}(\theta, X_{1:n}^{\text{obs}})$, yielding

$$\hat{C}_{1-\alpha}(X_{1:n}^{\text{obs}}) = \{\theta : \hat{\text{pv}}(\theta, X_{1:n}^{\text{obs}}) \geq \alpha\}.$$

Similarly, one may obtain frequentist confidence sets via quantile regression as in Dalmasso et al. (2024). Let $q_{1-\alpha}(\theta)$ be the $1-\alpha$ quantile of the test statistic, so that

$$P_{n,\theta}\left(T(\theta, X_{1:n}(\theta)) \leq q_{1-\alpha}(\theta)\right) = 1 - \alpha.$$

An exact $1-\alpha$ confidence set for the true parameter θ_0 is given by

$$C_{1-\alpha}(X_{1:n}^{\text{obs}}) = \{\theta : T(\theta, X_{1:n}^{\text{obs}}) \leq q_{1-\alpha}(\theta)\},$$

so that, for the true parameter $\theta_0 \in \Theta$, we have

$$P_{n,\theta_0}\left(\theta_0 \in C_{1-\alpha}(X_{1:n}^{\text{obs}})\right) = 1 - \alpha.$$

We use the following procedure to estimate the $1-\alpha$ quantile $q_{1-\alpha}(\theta)$ and obtain an approximate $1-\alpha$ confidence set.

Frequentist Confidence Sets using Quantile Regression.

- **Inputs:** Observations $X_{1:n}^{\text{obs}}$, model, reference distribution π, test statistic T, number of simulations s, and quantile regression method.

- For each $r \in [s]$, draw $\theta^{(r)} \sim \pi$ and simulate from the model,

$$X_{1:n}^{(r)} \sim P_{n,\theta^{(r)}},$$

  and obtain $T^{(r)} = T(\theta^{(r)}, X_{1:n}^{(r)})$.

- Perform quantile regression of $T^{(r)}$ on $\theta^{(r)}$, $r \in [s]$, to get an estimate $\hat{q}(\theta)$ of the $1-\alpha$ quantile function $q(\theta)$, yielding

$$\hat{C}_{1-\alpha}(X_{1:n}^{\text{obs}}) = \{\theta : T(\theta, X_{1:n}^{\text{obs}}) \leq \hat{q}_{1-\alpha}(\theta)\}.$$

4.2 Proposed Work I: Goodness-of-fit and Confidence Sets

Building on Section 3, we propose developing a simulation-based inference framework using random features. First, we aim to develop a goodness-of-fit test for the model $\mathcal{P}_n = \{P_{n,\theta} : \theta \in \Theta\}$. Second, we seek to develop a framework for constructing confidence sets.

We begin with the goodness-of-fit test using random features. Specifically, we suggest a test for the null hypothesis that the true distribution P_n is contained in the model class $\mathcal{P}_n$. Our proposed test is based on the goodness-of-fit test from Tomaselli et al. (2025), which is shown to have asymptotic Type I error control. We will study the power and computational cost of our random features-based test as we increase the number of random features used.

For the following procedure, it will be useful to introduce notation for random feature discrepancies, which are based on the objectives from Section 3.4. First, define the time-average random feature discrepancy as

$$d^{\mathrm{TA}}(P_n^X, P_{n,\theta}^Y) = \left\| F^X - \bar{F}^Y(\theta) \right\|^2, \tag{13}$$

where P_n^X is the distribution of $X_{1:n}$, $P_{n,\theta}^Y \in \mathcal{P}_n$ is the distribution of $Y_{1:n}(\theta)$,

$$F^X = \frac{1}{n-m} \sum_{t=m+1}^{n} f_t^X,$$

is the time-average of random features of the time series $X_{1:n}$ as in (8),

$$\bar{F}^Y(\theta) = \frac{1}{n-m} \sum_{t=m+1}^{n} \bar{f}_t^Y(\theta),$$

where $\bar{f}_t^Y(\theta) = \frac{1}{s} \sum_{r=1}^{s} f_t^{Y(r)}(\theta)$, is the time-average of the simulation-average of the random features of the time series $Y_{1:n}(\theta)$ as in (8). Second, define the rolling-window random feature discrepancy as

$$d^{\mathrm{RW}}(P_n^X, P_{n,\theta}^Y) = \frac{1}{n-m} \sum_{t=m+\tau+L}^{n} \left[\left\| F_t^X - \bar{F}_t^Y(\theta) \right\|^2 + K_t(\theta) \right], \tag{14}$$

where

$$K_t(\theta) = 2 \left(F_{t-L}^X - \bar{F}_{t-L}^Y(\theta) \right)^\top \left(\left[f_t^X - \bar{f}_t^Y(\theta) \right] - \left[F_{t-L}^X - \bar{F}_{t-L}^Y(\theta) \right] \right).$$

Here, P_n^X is the distribution of $X_{1:n}$, $P_{n,\theta}^Y \in \mathcal{P}_n$ is the distribution of $Y_{1:n}(\theta)$,

$$F_t^X = \frac{1}{w \wedge t} \sum_{j=(t-w)\vee 1}^{t} f_j^X,$$

is the rolling-window average of random features of the time series $X_{1:n}$ as in (11),

$$\bar{F}_t^Y(\theta) = \frac{1}{w \wedge t} \sum_{j=(t-w)\vee 1}^{t} \bar{f}_j^Y(\theta),$$

where $\bar{f}_t^Y(\theta) = \frac{1}{s}\sum_{r=1}^{s} f_t^{Y(r)}(\theta)$, is the rolling-window time-average of the simulation-average of the random features of the time series $Y_{1:n}(\theta)$ as in (11). Note that P_n^X is not necessarily in the model class $\mathcal{P}_n$, while $P_{n,\theta}^Y$ is.

Goodness-of-fit Test.

- **Inputs:** Observations $X_{1:n}^{\text{obs}}$, model, reference distribution π, random features $\varphi_1, \ldots, \varphi_k$, random feature discrepancy $d(\cdot,\cdot)$ as in (13) or (14), number of simulations s, kernel K, and method for selecting the bandwidth h.

- For each $r \in [s]$, draw $\theta^{(r)} \sim \pi$ and simulate two independent realizations of the process,

$$X_{1:n}^{(r)} \sim P_{n,\theta^{(r)}}, \quad \tilde{X}_{1:n}^{(r)} \sim \tilde{P}_{n,\theta^{(r)}},$$

  where $\tilde{P}_{n,\theta^{(r)}} = P_{n,\theta^{(r)}}$.

- Evaluate

$$\hat{T}_n = \min_{j\in[s]} d(P_n, \tilde{P}_{n,\theta^{(j)}}),$$

  where $\hat{T}_n$ uses $X_{1:n}^{\text{obs}}$ and $(\tilde{X}_{1:n}^{(j)})_{j\in[s]}$.

- For each $r \in [s]$, evaluate

$$\hat{T}_n(\theta^{(r)}) = \min_{j\in[s]} d(P_{n,\theta^{(r)}}, \tilde{P}_{n,\theta^{(j)}}),$$

  where $\hat{T}_n(\theta^{(r)})$ uses $X_{1:n}^{(r)}$ and $(\tilde{X}_{1:n}^{(j)})_{j\in[s]}$.

- For each $j \in [s]$, calculate

$$\hat{\text{pv}}(\theta^{(j)}) = \frac{\sum_{r \in [s]} K_h(\theta^{(r)} - \theta^{(j)}) \mathbb{I}(\hat{T}_n(\theta^{(r)}) \geq \hat{T}_n)}{\sum_{r \in [s]} K_h(\theta^{(r)} - \theta^{(j)})},$$

  where K is a kernel and h is the bandwidth.

- Obtain the estimated p-value

$$\hat{\text{pv}} = \max_{j \in [s]} \hat{\text{pv}}(\theta^{(j)}).$$

Next, we discuss confidence set construction in the setting where the model is correctly specified. This should be a straightforward application of the procedures from Dalmasso et al. (2024) for constructing confidence sets via nonparametric regression or quantile regression (see Section 4.1). In particular, we will use test statistics based on the random feature discrepancies that were introduced above. We will focus on studying the tradeoffs between the number of random features used, the computational cost, and the size of the resulting confidence sets.

If time allows, we will also consider confidence set construction in the misspecified setting. Here, the goal is to construct confidence sets for the projection parameter θ_* that minimizes $d(P_{n,\theta}, P_n)$ over $\theta \in \Theta$ for some discrepancy $d(\cdot, \cdot)$. We will use a discrepancy based on the random feature discrepancies introduced above. We suspect that it is possible to extend ideas from Tomaselli et al. (2025) from the iid setting to the nonstationary time series setting. However, this may be difficult, because Algorithm 1 in Section 5 of Tomaselli et al. (2025) relies on sample splitting. If we are unable to extend these ideas in a satisfactory way, we will construct confidence sets using M-estimator asymptotic methods enabled by making stronger assumptions.

5 Prediction with Transformers

5.1 Overview

In this part of the thesis, we propose a transformer-based framework for forecasting and smoothing for state-space models. *State-space models* decompose observed time series $Y_{1:t}$ into latent states $X_{1:t}$, control inputs $Z_{1:t}$, and noise. Consider the general state-space model for an $\mathbb{R}^{d_Y}$-valued observed process

$Y_{1:n}$ and an $\mathbb{R}^{d_X}$-valued latent state process $X_{1:n}$. For times $t = 1, \ldots, n$, let

$$\begin{aligned} Y_t &= M_t(X_t, Z_t, \varepsilon_t^Y, \theta), & (\varepsilon_t^X, \varepsilon_t^Y) &\sim P_\theta^{\varepsilon_t^X, \varepsilon_t^Y}, \\ X_t &= S_t(X_{t-1}, Z_t, \varepsilon_t^X, \theta), & X_0 &\sim P_\theta^{X_0}, \end{aligned} \tag{15}$$

where M_t is the measurement function, S_t is the state transition function, and θ is a p-dimensional parameter. To give a sense of the models we are interested in, we give some examples. First, consider an autoregressive process (with no controls) observed through noise

$$\begin{aligned} Y_t &= X_t + \varepsilon_t^Y, & (\varepsilon_t^X, \varepsilon_t^Y) &\sim N(\mathbf{0}, \mathbf{\Sigma}), \\ X_t &= \phi X_{t-1} + \varepsilon_t^X, & X_0 &\sim N(0, \sigma_1^2), \end{aligned} \tag{16}$$

where

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_X^2 & \rho\, \sigma_X \sigma_Y \\ \rho\, \sigma_X \sigma_Y & \sigma_Y^2 \end{pmatrix},$$

for some $\phi \in (-1, 1)$, $\rho \in [0, 1]$, and $\sigma_1, \sigma_X, \sigma_Y > 0$. Second, consider a discrete-valued, non-Gaussian process with $\mathbb{R}^{d_Z}$-valued controls $Z_{1:n}$

$$\begin{aligned} Y_t &\sim \text{Poisson}(X_t), \\ \log(X_t) &= \beta \alpha_t + \beta_Z^\top Z_t + \mu, \\ \alpha_t &= \phi \alpha_{t-1} + \epsilon_t, \end{aligned} \tag{17}$$

where $\epsilon_t \sim t_5$, $\alpha_0 \sim t_5$, $\phi \in (-1, 1)$, $\beta \in \mathbb{R}$, $\beta_Z \in \mathbb{R}^{d_Z}$, and $\mu \in \mathbb{R}$.

In the context of state-space modeling, *smoothing* is the task of estimating the past latent states $X_{1:t}$ using the information up to time t. This can be done via simulation-based estimates of the regression functions $\mathbb{E}(X_j \mid Y_{1:t} = y_{1:t}, Z_{1:t} = z_{1:t})$ for $j = 1, \ldots, t$. *Forecasting* is the task of predicting the future observations or states multiple time-steps ahead. In the state-space setting, a future path of the control inputs must be specified, either deterministically or stochastically, or the model must be specified so that no controls are used. Given the observations $Y_{1:t}$ and (optionally) control inputs $Z_{1:t+h}$, we aim to make point forecasts via simulation-based estimates of $\mathbb{E}(X_{t+j} \mid Y_{1:t} = y_{1:t}, Z_{1:t+j} = z_{1:t+j})$ or $\mathbb{E}(Y_{t+j} \mid Y_{1:t} = y_{1:t}, Z_{1:t+j} = z_{1:t+j})$ for $j = 1, \ldots, h$. Similar ideas apply for estimating the conditional quantiles.

5.2 Proposed Work II: Extremum Monte Carlo Methods

Moussa et al. (2026) introduce a flexible simulation-and-regression method called *extremum Monte Carlo* (XMC). The XMC method uses nonparametric

regression, such as random forests and XGBoost, to carry out forecasting and smoothing for state-space models when the parameter is assumed to be known. We propose two procedures that extend the original XMC method.

First, we propose the *robust extremum Monte Carlo* (rXMC) method. We develop a new estimation procedure that incorporates uncertainty about the true parameter, rather than assuming it is known. We also modify the original XMC framework to allow the use of sequence-to-sequence models, such as transformers.

Second, we propose an *accelerated robust extremum Monte Carlo* (arXMC) method. We explore whether it is possible to improve the computational efficiency of the rXMC method by using a certain Gaussian approximation in the algorithm. This approximation is enabled when the estimator $\hat{\theta}$ is asymptotically normal.

We aim to show that the estimators obtained by the rXMC and arXMC procedures are consistent as the sample size n and number of simulations s tend to infinity, with convergence rates determined by the parameter estimator and regression estimator. We will study the finite-sample performance of the rXMC and arXMC procedures, with comparisons to baselines including particle smoothing, the Kalman smoother, and the original XMC procedure when using an estimator $\hat{\theta}$ of the true parameter θ_0. We will also consider forecasting and conditional quantile estimation.

The following algorithm describes the proposed rXMC method in the context of smoothing. Specifically, estimating the conditional expectations of the past states given the current information. Similar ideas apply for forecasting and estimating conditional quantiles.

Let π be a reference distribution with full support on the parameter space Θ. The algorithm utilizes an estimator $\hat{\theta}$ of the true parameter θ_0 of the model from (15), which is assumed to be consistent. The control inputs are treated as fixed, and are absorbed into the definition of the model (15).

Robust Extremum Monte Carlo (rXMC).

- **Inputs:** Observations $Y_{1:n}^{\text{obs}}$, state-space model (15), estimator $\hat{\theta}$, K for cross-validation, reference distribution π, and number of simulations s.
- For each $r \in [s]$, draw $\theta^{(r)} \sim \pi$ and simulate from the model (15),

$$X_{1:n}^{(r)}, Y_{1:n}^{(r)} \sim P_{n,\theta^{(r)}},$$

  and obtain estimates $\hat{\theta}^{(r)}$ of $\theta^{(r)}$.

- Partition $[s]$ into K folds $F_1, \ldots, F_K$ of roughly equal size. For each $k \in [K]$ and each value of the tuning parameter $\lambda \in \Lambda$, perform sequence-to-sequence regression to obtain the minimizer $\hat{f}_\lambda^{-k}$ of

$$\sum_{j \in [K] \setminus \{k\}} \sum_{r \in F_j} \sum_{t \in [n]} (X_t^{(r)} - f[Y_{1:n}^{(r)}, \hat{\theta}^{(r)}]_t)^2,$$

  over the function space of transformers, record the corresponding error on the k-th validation set

$$e_k(\lambda) = \sum_{r \in F_k} \sum_{t \in [n]} (X_t^{(r)} - \hat{f}_\lambda^{-k}[Y_{1:n}^{(r)}, \hat{\theta}^{(r)}]_t)^2,$$

  then compute the average error over the K folds

$$\mathrm{CV}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} e_k(\lambda).$$

- Select the value of the tuning parameter λ that minimizes $\mathrm{CV}(\lambda)$, to obtain the sequence-to-sequence function estimate $\hat{f}$.
- Use $Y_{1:n}^{\mathrm{obs}}$ to obtain estimate $\hat{\theta}$ of the true parameter θ_0, then apply the estimated sequence-to-sequence function to predict the latent states

$$\hat{X}_{1:n} = \hat{f}(Y_{1:n}^{\mathrm{obs}}, \hat{\theta}).$$

Next, we present the arXMC procedure, which only requires $s_1 \ll s$ parameter estimates by using a Gaussian approximation justified by the asymptotic normality of the estimator.

Accelerated Robust Extremum Monte Carlo (arXMC).

- **Inputs:** Observations $Y_{1:n}^{\mathrm{obs}}$, state-space model (15), estimator $\hat{\theta}$, K for cross-validation, reference distribution π, number of simulations s, and splitting proportion $c \in (0, 1)$.
- Let $s_1 = \lfloor cs \rfloor$ and $s_2 = s - s_1$.
- For each $r \in [s_1]$, draw $\theta^{(r)} \sim \pi$ and simulate from the model (15),

$$X_{1:n}^{(r)}, Y_{1:n}^{(r)} \sim P_{n,\theta^{(r)}},$$

  and obtain estimates $\hat{\theta}^{(r)}$ of $\theta^{(r)}$.

- Regress $\hat{\theta}^{(r)}$ on $\theta^{(r)}$, $r \in [s_1]$, to get an estimate $\hat{m}(\theta)$ of

$$m(\theta) = \mathbb{E}_\theta[\hat{\theta}].$$

- Regress $\hat{\theta}^{(r)}\hat{\theta}^{(r)\top}$ on $\theta^{(r)}$, $r \in [s_1]$, to get an estimate $\hat{M}(\theta)$ of

$$M(\theta) = \mathbb{E}_\theta[\hat{\theta}\hat{\theta}^\top].$$

- Calculate $\hat{\Sigma}(\theta) = \hat{M}(\theta) - \hat{m}(\theta)\hat{m}(\theta)^\top$ to get an estimate of

$$\Sigma(\theta) = M(\theta) - m(\theta)m(\theta)^\top.$$

- For each $r \in [s_2]$, draw $\theta^{(r)} \sim \pi$ and simulate from the model (15),

$$X_{1:n}^{(r)}, Y_{1:n}^{(r)} \sim P_{n,\theta^{(r)}},$$

  and set $\tilde{\theta}^{(r)} = \theta^{(r)} + \varepsilon^{(r)}$, where $\varepsilon^{(r)} \sim N(\hat{m}(\theta^{(r)}), \hat{\Sigma}(\theta^{(r)}))$.

- Partition $[s_2]$ into K folds $F_1, \ldots, F_K$ of roughly equal size. For each $k \in [K]$ and each value of the tuning parameter $\lambda \in \Lambda$, perform sequence-to-sequence regression to obtain the minimizer $\hat{f}_\lambda^{-k}$ of

$$\sum_{j \in [K]\setminus\{k\}} \sum_{r \in F_j} \sum_{t \in [n]} (X_t^{(r)} - f[Y_{1:n}^{(r)}, \tilde{\theta}^{(r)}]_t)^2,$$

  over the function space of transformers, record the corresponding error on the k-th validation set

$$e_k(\lambda) = \sum_{r \in F_k} \sum_{t \in [n]} (X_t^{(r)} - \hat{f}_\lambda^{-k}[Y_{1:n}^{(r)}, \tilde{\theta}^{(r)}]_t)^2,$$

  then compute the average error over the K folds

$$\mathrm{CV}(\lambda) = \frac{1}{K} \sum_{k=1}^{K} e_k(\lambda).$$

- Select the value of the tuning parameter λ that minimizes $\mathrm{CV}(\lambda)$, to obtain the sequence-to-sequence function estimate $\hat{f}$.

- Use $Y_{1:n}^{\mathrm{obs}}$ to obtain estimate $\hat{\theta}$ of the true parameter θ_0, then apply the estimated sequence-to-sequence function to predict the latent states

$$\hat{X}_{1:n} = \hat{f}(Y_{1:n}^{\mathrm{obs}}, \hat{\theta}).$$

6 Timeline

Step 1a: Proposal. I plan to propose my thesis in late March, 2026.

Step 1b: Revise and Submit. Concurrently, I plan to revise the conditional independence testing paper corresponding to Section 2, which has received a revise decision from the journal, and to submit the simulation-based parameter estimation paper corresponding to Section 3.

Step 2: Simulation-based Inference (Section 4). I plan to work on this topic from March 2026 to August 2026.

Step 3: Prediction with Transformers (Section 5). I plan to work on this topic from March 2026 to March 2027.

Step 4: Defense. I plan to defend my thesis before the spring defense deadline set by the Department, April 15, 2027, and submit the paperwork by May 1, 2027.

References

Chen, L., Smetanina, E., and Wu, W. B. (2022). Estimation of nonstationary nonparametric regression model with multiplicative structure. *The Econometrics Journal*, 25(1):176–214.

Dalmasso, N., Masserano, L., Zhao, D., Izbicki, R., and Lee, A. B. (2024). Likelihood-free frequentist inference: Bridging classical statistics and machine learning for reliable simulator-based inference. *Electronic Journal of Statistics*, 18(2):5045–5090.

Devaney, R. L. (2018). *A first course in chaotic dynamical systems: theory and experiment*. CRC Press.

Ding, X. and Zhou, Z. (2021). Simultaneous sieve inference for time-inhomogeneous nonlinear time series regression. arXiv preprint arXiv:2112.08545.

Gourieroux, C. and Monfort, A. (1996). *Simulation-based econometric methods*. Oxford University Press, Oxford, England.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 29:3323–3331.

Kurisu, D., Fukami, R., and Koike, Y. (2025). Adaptive deep learning for nonlinear time series models. *Bernoulli*, 31(1):240–270.

Mies, F. (2023). Functional estimation and change detection for nonstationary time series. *Journal of the American Statistical Association*, 118(542):1011–1022.

Mies, F. and Steland, A. (2023). Sequential gaussian approximation for nonstationary time series in high dimensions. *Bernoulli*, 29(4):3114–3140.

Moussa, K., Blasques, F., and Koopman, S. J. (2026). Extremum monte carlo filters: signal extraction via simulation and regression. *Journal of Business and Economic Statistics*, pages 1–13.

Sauer, T., Yorke, J. A., and Casdagli, M. (1991). Embedology. *Journal of Statistical Physics*, 65(3):579–616.

Shah, R. D. and Peters, J. (2020). The hardness of conditional independence testing and the generalised covariance measure. *Annals of Statistics*, 48(3):1514–1538.

Tomaselli, L., Ventura, V., and Wasserman, L. (2025). Robust simulation based inference. arXiv preprint arXiv:2508.02404.

Vogt, M. (2012). Nonparametric regression for locally stationary time series. *Annals of Statistics*, 50(5):2601–2633.

Wieck-Sosa, M., Haddad, M. F. C., and Ramdas, A. (2025). Conditional independence testing with a single realization of a multivariate nonstationary nonlinear time series. arXiv preprint arXiv:2504.21647.

Wu, W. B. (2005). Nonlinear system theory: another look at dependence. *Proceedings of the National Academy of Sciences*, 102(40):14150–14154.

Yousuf, K. and Ng, S. (2021). Boosting high dimensional predictive regressions with time varying parameters. *Journal of Econometrics*, 224(1):60–87.

Zhang, T. and Wu, W. B. (2015). Time-varying nonlinear regression models: nonparametric estimation and model selection. *Annals of Statistics*, 43(2):741–768.

Zhou, Z. and Wu, W. B. (2009). Local linear quantile estimation for nonstationary time series. *The Annals of Statistics*, 37(5B):2696–2729.