Simulation-based Time Series Analysis:

Likelihood-free Estimation, Testing for Structure, and Prediction with Transformers using Synthetic Data

Michael Wieck-Sosa

Department of Statistics & Data Science, Carnegie Mellon University

April 3, 2026

Overview of presentation

- Introduction to time series

Overview of presentation

- Introduction to time series
- Conditional independence testing

Overview of presentation

- Introduction to time series
- Conditional independence testing
- Simulation-based estimation

Overview of presentation

- Introduction to time series
- Conditional independence testing
- Simulation-based estimation
- Prediction with transformers trained on synthetic data

Time series setting

- We observe X_t, $t = 1, \ldots, n$, for some sample size $n \in \mathbb{N}$

Time series setting

- We observe X_t, $t = 1, \ldots, n$, for some sample size $n \in \mathbb{N}$
- Each observation X_t takes values in $\mathbb{R}^d$ for some dimension $d \in \mathbb{N}$

Time series setting

- We observe X_t, $t = 1, \ldots, n$, for some sample size $n \in \mathbb{N}$
- Each observation X_t takes values in $\mathbb{R}^d$ for some dimension $d \in \mathbb{N}$
- **Temporal dependence:** X_t may depend on the past $X_{t-1}, X_{t-2}, \ldots$

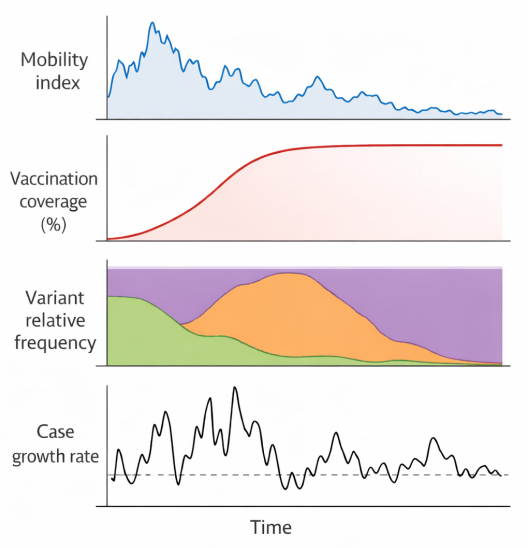
Time series setting

- We observe X_t, $t = 1, \ldots, n$, for some sample size $n \in \mathbb{N}$
- Each observation X_t takes values in $\mathbb{R}^d$ for some dimension $d \in \mathbb{N}$
- **Temporal dependence:** X_t may depend on the past $X_{t-1}, X_{t-2}, \ldots$
- **Nonstationarity:** the joint distribution of $X_{t_1+\tau}, \ldots, X_{t_m+\tau}$ at $m \in \mathbb{N}$ distinct time points $t_1, \ldots, t_m$ may change with the time-offset τ

Time series setting

- We observe X_t, $t = 1, \ldots, n$, for some sample size $n \in \mathbb{N}$
- Each observation X_t takes values in $\mathbb{R}^d$ for some dimension $d \in \mathbb{N}$
- **Temporal dependence:** X_t may depend on the past $X_{t-1}, X_{t-2}, \ldots$
- **Nonstationarity:** the joint distribution of $X_{t_1+\tau}, \ldots, X_{t_m+\tau}$ at $m \in \mathbb{N}$ distinct time points $t_1, \ldots, t_m$ may change with the time-offset τ
- Example: time series in epidemiology

COVID-19 time series May 2020–2022 (n=365x2=730, d=4)



Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$

Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$
- Let these variables incorporate any desired leads and lags (to simplify notation)

Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$
- Let these variables incorporate any desired leads and lags (to simplify notation)
- The conditioning set of variables Z_t must be known at time t

Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$
- Let these variables incorporate any desired leads and lags (to simplify notation)
- The conditioning set of variables Z_t must be known at time t
- We present a test for the null hypothesis that

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

Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$
- Let these variables incorporate any desired leads and lags (to simplify notation)
- The conditioning set of variables Z_t must be known at time t
- We present a test for the null hypothesis that

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

- Motivations: understanding structure, variable selection, and causal discovery

Conditional independence testing

- Let $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$ be nonstationary time series taking values in $\mathbb{R}^{d_X}$, $\mathbb{R}^{d_Y}$, $\mathbb{R}^{d_Z}$
- Let these variables incorporate any desired leads and lags (to simplify notation)
- The conditioning set of variables Z_t must be known at time t
- We present a test for the null hypothesis that

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

- Motivations: understanding structure, variable selection, and causal discovery
- Example: links in the global economic system

Real data application: links in the global economic system

- S&P 500 (US), FTSE 100 (UK), Hang Seng (HK), and Nikkei 225 (JP)

Real data application: links in the global economic system

- S&P 500 (US), FTSE 100 (UK), Hang Seng (HK), and Nikkei 225 (JP)
- Daily log returns based on adjusted closing prices

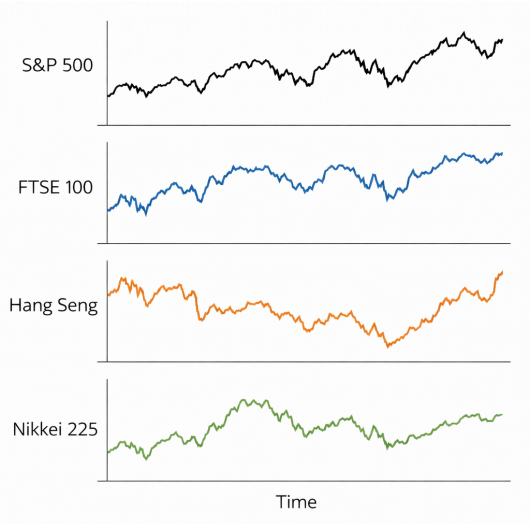
Real data application: links in the global economic system

- S&P 500 (US), FTSE 100 (UK), Hang Seng (HK), and Nikkei 225 (JP)
- Daily log returns based on adjusted closing prices
- Trading days from January 2022 to April 2025 ($n = 860$)

Real data application: links in the global economic system

- S&P 500 (US), FTSE 100 (UK), Hang Seng (HK), and Nikkei 225 (JP)
- Daily log returns based on adjusted closing prices
- Trading days from January 2022 to April 2025 ($n = 860$)
- We show a subset of the results based on BH-adjusted p-values

Closing prices of each stock index over time



Independence testing reveals same day returns are dependent

- For each pair $X, Y \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

$$X_t \perp\!\!\!\perp Y_t \text{ for all times } t \in [n]$$

Independence testing reveals same day returns are dependent

- For each pair $X, Y \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

$$X_t \perp\!\!\!\perp Y_t \text{ for all times } t \in [n]$$

- We reject *every* null hypothesis at the significance level $\alpha = 0.05$

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

- **Retain:** `S&P(t)` is independent of `HangSeng(t)` and `Nikkei(t)` given `FTSE(t)`

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

- **Retain:** `S&P(t)` is independent of `HangSeng(t)` and `Nikkei(t)` given `FTSE(t)`
- **Reject:** `S&P(t)` is independent of `FTSE(t)` given `HangSeng(t)` or `Nikkei(t)`

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

- **Retain:** `S&P(t)` is independent of `HangSeng(t)` and `Nikkei(t)` given `FTSE(t)`
- **Reject:** `S&P(t)` is independent of `FTSE(t)` given `HangSeng(t)` or `Nikkei(t)`
- UK has info about Asia-Pacific markets on the same day, but not vice versa

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

- **Retain:** `S&P(t)` is independent of `HangSeng(t)` and `Nikkei(t)` given `FTSE(t)`
- **Reject:** `S&P(t)` is independent of `FTSE(t)` given `HangSeng(t)` or `Nikkei(t)`
- UK has info about Asia-Pacific markets on the same day, but not vice versa
- Why? Earliest to latest closing exchanges: Tokyo, Hong Kong, London, New York

Conditional independence testing reveals structure

- For each triplet $X, Y, Z \in \{\texttt{S\&P}, \texttt{FTSE}, \texttt{HangSeng}, \texttt{Nikkei}\}$, test

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

- **Retain:** `S&P(t)` is independent of `HangSeng(t)` and `Nikkei(t)` given `FTSE(t)`
- **Reject:** `S&P(t)` is independent of `FTSE(t)` given `HangSeng(t)` or `Nikkei(t)`
- UK has info about Asia-Pacific markets on the same day, but not vice versa
- Why? Earliest to latest closing exchanges: Tokyo, Hong Kong, London, New York
- Next, we present the testing procedure

Dynamic generalized covariance measure test (1/2)

- **Inputs:** Data $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, significance level α, number of simulations s, $p \in [2, \infty]$ for norm in test statistic, and method for selecting the window size L

Dynamic generalized covariance measure test (1/2)

- **Inputs:** Data $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, 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 $\hat{\varepsilon}_t$, $t \in [n]$

Dynamic generalized covariance measure test (1/2)

- **Inputs:** Data $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, 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 $\hat{\xi}_t$, $t \in [n]$

Dynamic generalized covariance measure test (1/2)

- **Inputs:** Data $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, 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 $\hat{R}_t = \mathrm{vec}(\hat{\varepsilon}_t \hat{\xi}_t^\top)$, $t \in [n]$

Dynamic generalized covariance measure test (1/2)

- **Inputs:** Data $X_{1:n}$, $Y_{1:n}$, $Z_{1:n}$, 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

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

Dynamic generalized covariance measure test (2/2)

- Select the window size L for covariance estimation using $\hat{R}_{1:n}$

Dynamic generalized covariance measure test (2/2)

- 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

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

Dynamic generalized covariance measure test (2/2)

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

Dynamic generalized covariance measure test (2/2)

- 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

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

Dynamic generalized covariance measure test (2/2)

- 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

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

- Calculate the $1 - \alpha$ empirical quantile $\hat{q}_{1-\alpha}^{\text{boot}}$ of $T(\tilde{R}_{1:n}^{(1)}), \ldots, T(\tilde{R}_{1:n}^{(s)})$

Dynamic generalized covariance measure test (2/2)

- 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

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

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

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

- If a CI test has Type-I error control (i.e. size α), then it cannot have power against any alternative (i.e. power against each alternative $Q \in \mathcal{Q}$ is at most α)

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

- If a CI test has Type-I error control (i.e. size α), then it cannot have power against any alternative (i.e. power against each alternative $Q \in \mathcal{Q}$ is at most α)
- If a CI test has power β against an alternative $Q \in \mathcal{Q}$, then there exists a null distribution $P \in \mathcal{P}$ s.t. the test will reject with prob. greater than or equal to β

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

- If a CI test has Type-I error control (i.e. size α), then it cannot have power against any alternative (i.e. power against each alternative $Q \in \mathcal{Q}$ is at most α)
- If a CI test has power β against an alternative $Q \in \mathcal{Q}$, then there exists a null distribution $P \in \mathcal{P}$ s.t. the test will reject with prob. greater than or equal to β

- This result has been extended to the time series setting (Bodik and Pasche 2024)

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

- If a CI test has Type-I error control (i.e. size α), then it cannot have power against any alternative (i.e. power against each alternative $Q \in \mathcal{Q}$ is at most α)
- If a CI test has power β against an alternative $Q \in \mathcal{Q}$, then there exists a null distribution $P \in \mathcal{P}$ s.t. the test will reject with prob. greater than or equal to β

- This result has been extended to the time series setting (Bodik and Pasche 2024)
- **Implication:** We can only hope to achieve Type-I error control on a subset of null distributions $\mathcal{P}_n^{\mathrm{CI}}$, so must restrict the null hypothesis to make CI testing feasible

The hardness of conditional independence (CI) testing

No-free-lunch in CI testing (Shah and Peters 2020)

- If a CI test has Type-I error control (i.e. size α), then it cannot have power against any alternative (i.e. power against each alternative $Q \in \mathcal{Q}$ is at most α)
- If a CI test has power β against an alternative $Q \in \mathcal{Q}$, then there exists a null distribution $P \in \mathcal{P}$ s.t. the test will reject with prob. greater than or equal to β

- This result has been extended to the time series setting (Bodik and Pasche 2024)
- **Implication:** We can only hope to achieve Type-I error control on a subset of null distributions $\mathcal{P}_n^{\mathrm{CI}}$, so must restrict the null hypothesis to make CI testing feasible
- **Key assumption:** uniform decay of physical dependence measure (Wu 2005)

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Let $(\epsilon_i)_{i \in \mathbb{Z}}$ be an iid sequence of $U[0,1]$ random seeds

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Let $(\epsilon_i)_{i \in \mathbb{Z}}$ be an iid sequence of $U[0,1]$ random seeds
- Define the sequence of noise inputs up to time t by $\boldsymbol{\epsilon}_t = (\epsilon_t, \epsilon_{t-1}, \ldots)$

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Let $(\epsilon_i)_{i \in \mathbb{Z}}$ be an iid sequence of $U[0,1]$ random seeds
- Define the sequence of noise inputs up to time t by $\boldsymbol{\epsilon}_t = (\epsilon_t, \epsilon_{t-1}, \ldots)$
- Let Θ be a parameter space, which can be infinite-dimensional

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Let $(\epsilon_i)_{i\in\mathbb{Z}}$ be an iid sequence of $U[0,1]$ random seeds
- Define the sequence of noise inputs up to time t by $\boldsymbol{\epsilon}_t = (\epsilon_t, \epsilon_{t-1}, \ldots)$
- Let Θ be a parameter space, which can be infinite-dimensional
- Let $G_t^{(n)} : \mathbb{R}^\infty \times \Theta \to \mathbb{R}^d$ be some measurable function, $t, n, d \in \mathbb{N}$

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Given a parameter $\theta \in \Theta$, index $n \in \mathbb{N}$ (linked to sample size and approximation), and noise inputs $\boldsymbol{\epsilon}_t$, a time series $X_{1:n}$ is *generated by nature* as

$$X_t = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta\right), \quad t = 1, \ldots, n$$

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Given a parameter $\theta \in \Theta$, index $n \in \mathbb{N}$ (linked to sample size and approximation), and noise inputs $\boldsymbol{\epsilon}_t$, a time series $X_{1:n}$ is *generated by nature* as

$$X_t = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta\right), \quad t = 1, \ldots, n$$

- We **do not know** the generative model $G_t^{(n)}$, noise inputs $\boldsymbol{\epsilon}_t$, parameter θ

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Given a parameter $\theta \in \Theta$, index $n \in \mathbb{N}$ (linked to sample size and approximation), and noise inputs $\boldsymbol{\epsilon}_t$, a time series $X_{1:n}$ is *generated by nature* as

$$X_t = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta\right), \quad t = 1, \ldots, n$$

- We **do not know** the generative model $G_t^{(n)}$, noise inputs $\boldsymbol{\epsilon}_t$, parameter θ
- Let $P_{\theta,n}$ be the (finite-dimensional) distribution of $(G_t^{(n)}(\boldsymbol{\epsilon}_t, \theta))_{t\in[n]}$

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Given a parameter $\theta \in \Theta$, index $n \in \mathbb{N}$ (linked to sample size and approximation), and noise inputs $\boldsymbol{\epsilon}_t$, a time series $X_{1:n}$ is *generated by nature* as

$$X_t = G_t^{(n)}(\boldsymbol{\epsilon}_t, \theta), \quad t = 1, \ldots, n$$

- We **do not know** the generative model $G_t^{(n)}$, noise inputs $\boldsymbol{\epsilon}_t$, parameter θ
- Let $P_{\theta,n}$ be the (finite-dimensional) distribution of $(G_t^{(n)}(\boldsymbol{\epsilon}_t, \theta))_{t \in [n]}$
- Let P_θ be the law of $(G_t(\boldsymbol{\epsilon}_t, \theta))_{t \in \mathbb{N}}$ for some limiting mapping G_t

Key assumption: Representation in terms of noise inputs (Wu 2005)

- Given a parameter $\theta \in \Theta$, index $n \in \mathbb{N}$ (linked to sample size and approximation), and noise inputs $\boldsymbol{\epsilon}_t$, a time series $X_{1:n}$ is *generated by nature* as

$$X_t = G_t^{(n)}(\boldsymbol{\epsilon}_t, \theta), \quad t = 1, \ldots, n$$

- We **do not know** the generative model $G_t^{(n)}$, noise inputs $\boldsymbol{\epsilon}_t$, parameter θ
- Let $P_{\theta,n}$ be the (finite-dimensional) distribution of $(G_t^{(n)}(\boldsymbol{\epsilon}_t, \theta))_{t \in [n]}$
- Let P_θ be the law of $(G_t(\boldsymbol{\epsilon}_t, \theta))_{t \in \mathbb{N}}$ for some limiting mapping G_t
- Denote the collections by $\mathcal{P}_n = \{P_{\theta,n} : \theta \in \Theta\}$ and $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$

Key assumption: Decaying influence of noise inputs (Wu 2005)

- Let $(\epsilon_i^*)_{i\in\mathbb{Z}}$ be an iid copy of $(\epsilon_i)_{i\in\mathbb{Z}}$. For $\ell \in \mathbb{Z}$, $j \in \mathbb{N}_0$, denote

$$\tilde{\boldsymbol{\epsilon}}_{\ell,j} = \left(\epsilon_\ell, \ldots, \epsilon_{\ell-j+1}, \epsilon^*_{\ell-j}, \epsilon_{\ell-j-1}, \ldots\right)$$

Key assumption: Decaying influence of noise inputs (Wu 2005)

- Let $(\epsilon_i^*)_{i\in\mathbb{Z}}$ be an iid copy of $(\epsilon_i)_{i\in\mathbb{Z}}$. For $\ell \in \mathbb{Z}$, $j \in \mathbb{N}_0$, denote

$$\tilde{\boldsymbol{\epsilon}}_{\ell,j} = \left(\epsilon_\ell, \ldots, \epsilon_{\ell-j+1}, \epsilon^*_{\ell-j}, \epsilon_{\ell-j-1}, \ldots\right)$$

- There exist $\Psi, \rho > 0$ such that, for some $q > 2$ and all $j \in \mathbb{N}$, we have

$$\sup_{\theta\in\Theta}\sup_{\ell\in\mathbb{Z}}\sup_{i\in\mathbb{N}}\sup_{t\in\mathbb{N}} \left\| G_t^{(i)}(\boldsymbol{\epsilon}_\ell, \theta) - G_t^{(i)}(\tilde{\boldsymbol{\epsilon}}_{\ell,j}, \theta)\right\|_{L^q(\theta)} \leq \Psi(j \vee 1)^{-\rho}$$

$$\sup_{\theta\in\Theta}\sup_{\ell\in\mathbb{Z}}\sup_{i\in\mathbb{N}}\sup_{t\in\mathbb{N}} \left\| G_t^{(i)}(\boldsymbol{\epsilon}_\ell, \theta)\right\|_{L^q(\theta)} \leq \Psi$$

where $\|\cdot\|_{L^q(\theta)} = \mathbb{E}_\theta\left(\|\cdot\|_2^q\right)^{1/q}$, $\mathbb{E}_\theta(\cdot)$ is expectation w.r.t. θ-dependent distribution

Key assumption: Decaying influence of noise inputs (Wu 2005)

- Let $(\epsilon_i^*)_{i\in\mathbb{Z}}$ be an iid copy of $(\epsilon_i)_{i\in\mathbb{Z}}$. For $\ell \in \mathbb{Z}$, $j \in \mathbb{N}_0$, denote

$$\tilde{\boldsymbol{\epsilon}}_{\ell,j} = \left(\epsilon_\ell, \ldots, \epsilon_{\ell-j+1}, \epsilon^*_{\ell-j}, \epsilon_{\ell-j-1}, \ldots\right)$$

- There exist $\Psi, \rho > 0$ such that, for some $q > 2$ and all $j \in \mathbb{N}$, we have

$$\sup_{\theta\in\Theta} \sup_{\ell\in\mathbb{Z}} \sup_{i\in\mathbb{N}} \sup_{t\in\mathbb{N}} \left\| G_t^{(i)}(\boldsymbol{\epsilon}_\ell, \theta) - G_t^{(i)}(\tilde{\boldsymbol{\epsilon}}_{\ell,j}, \theta) \right\|_{L^q(\theta)} \leq \Psi(j \vee 1)^{-\rho}$$

$$\sup_{\theta\in\Theta} \sup_{\ell\in\mathbb{Z}} \sup_{i\in\mathbb{N}} \sup_{t\in\mathbb{N}} \left\| G_t^{(i)}(\boldsymbol{\epsilon}_\ell, \theta) \right\|_{L^q(\theta)} \leq \Psi$$

  where $\|\cdot\|_{L^q(\theta)} = \mathbb{E}_\theta \left(\|\cdot\|_2^q\right)^{1/q}$, $\mathbb{E}_\theta(\cdot)$ is expectation w.r.t. θ-dependent distribution

- We make this assumption for $X_{1:n}$, $Y_{1:n}$, and $Z_{1:n}$

We prove our test has uniformly asymptotic Type I error control

- For each $n \in \mathbb{N}$, let $\mathcal{P}_n^* \subset \mathcal{P}_n^{\mathrm{CI}}$ be a collection of joint distributions for $X_{1:n}$, $Y_{1:n}$, and $Z_{1:n}$ for which the null hypothesis holds *and all assumptions are satisfied*

We prove our test has uniformly asymptotic Type I error control

- For each $n \in \mathbb{N}$, let $\mathcal{P}_n^* \subset \mathcal{P}_n^{\mathrm{CI}}$ be a collection of joint distributions for $X_{1:n}$, $Y_{1:n}$, and $Z_{1:n}$ for which the null hypothesis holds *and all assumptions are satisfied*
- Need sufficiently fast convergence rates for the regression estimators

We prove our test has uniformly asymptotic Type I error control

- For each $n \in \mathbb{N}$, let $\mathcal{P}_n^* \subset \mathcal{P}_n^{\mathrm{CI}}$ be a collection of joint distributions for $X_{1:n}$, $Y_{1:n}$, and $Z_{1:n}$ for which the null hypothesis holds *and all assumptions are satisfied*
- Need sufficiently fast convergence rates for the regression estimators
- For all distributions $P \in \mathcal{P}_n^*$, the **product** of $L^2(P)$ norms of the estimation errors must be $o\left(\frac{1}{\sqrt{n}\ \mathrm{polylog}(n)}\right)$, but each $L^2(P)$ norm only needs to be $o\left(\frac{1}{\mathrm{polylog}(n)}\right)$

We prove our test has uniformly asymptotic Type I error control

- For each $n \in \mathbb{N}$, let $\mathcal{P}_n^* \subset \mathcal{P}_n^{\mathrm{CI}}$ be a collection of joint distributions for $X_{1:n}$, $Y_{1:n}$, and $Z_{1:n}$ for which the null hypothesis holds *and all assumptions are satisfied*
- Need sufficiently fast convergence rates for the regression estimators
- For all distributions $P \in \mathcal{P}_n^*$, the **product** of $L^2(P)$ norms of the estimation errors must be $o\left(\frac{1}{\sqrt{n}\ \mathrm{polylog}(n)}\right)$, but each $L^2(P)$ norm only needs to be $o\left(\frac{1}{\mathrm{polylog}(n)}\right)$

Theorem (Informal)

Recall the test statistic $T(\hat{R}_{1:n})$ and estimated quantile $\hat{q}_{1-\alpha}^{\mathrm{boot}}$. We have

$$\limsup_{n \to \infty} \sup_{P \in \mathcal{P}_n^*} \mathbb{P}_P \left(T(\hat{R}_{1:n}) > \hat{q}_{1-\alpha}^{\mathrm{boot}} \right) \leq \alpha.$$

Simulation-based estimation and inference

- Suppose we observe a time series $X_{1:n}^{\text{obs}}$ taking values in $\mathbb{R}^d$

Simulation-based estimation and inference

- Suppose we observe a time series $X_{1:n}^{\text{obs}}$ taking values in $\mathbb{R}^d$
- We assume the observed time series has been *generated by nature* as

$$X_t^{\text{obs}} = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta_0\right), \quad t = 1, \ldots, n$$

at some unknown parameter $\theta_0 \in \Theta \subset \mathbb{R}^p$ with some unknown noise inputs $\boldsymbol{\epsilon}_t$

Simulation-based estimation and inference

- Suppose we observe a time series $X_{1:n}^{\text{obs}}$ taking values in $\mathbb{R}^d$
- We assume the observed time series has been *generated by nature* as

$$X_t^{\text{obs}} = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta_0\right), \quad t = 1, \ldots, n$$

  at some unknown parameter $\theta_0 \in \Theta \subset \mathbb{R}^p$ with some unknown noise inputs $\boldsymbol{\epsilon}_t$
- We want to estimate θ_0, get a confidence set, or test goodness-of-fit

Simulation-based estimation and inference

- Suppose we observe a time series $X_{1:n}^{\text{obs}}$ taking values in $\mathbb{R}^d$
- We assume the observed time series has been *generated by nature* as

$$X_t^{\text{obs}} = G_t^{(n)}\left(\boldsymbol{\epsilon}_t, \theta_0\right), \quad t = 1, \ldots, n$$

  at some unknown parameter $\theta_0 \in \Theta \subset \mathbb{R}^p$ with some unknown noise inputs $\boldsymbol{\epsilon}_t$
- We want to estimate θ_0, get a confidence set, or test goodness-of-fit
- The likelihood is intractable, so we use a simulation-based approach

Simulation-based estimation and inference

- Suppose we observe a time series $X_{1:n}^{\text{obs}}$ taking values in $\mathbb{R}^d$
- We assume the observed time series has been *generated by nature* as

$$X_t^{\text{obs}} = G_t^{(n)}(\epsilon_t, \theta_0), \quad t = 1, \ldots, n$$

  at some unknown parameter $\theta_0 \in \Theta \subset \mathbb{R}^p$ with some unknown noise inputs ϵ_t
- We want to estimate θ_0, get a confidence set, or test goodness-of-fit
- The likelihood is intractable, so we use a simulation-based approach
- **Key assumption:** We know $G_t^{(n)}$ and can simulate from it at any value of $\theta \in \Theta$

How simulation-based estimation is usually done

- **Rough idea:** Tune the parameters θ until some features F "look right"

How simulation-based estimation is usually done

- **Rough idea:** Tune the parameters θ until some features F "look right"
- For (asymptotically) stationary time series, often use minimum distance estimator

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\| F^{\text{obs}} - \frac{1}{s} \sum_{r=1}^{s} F^{(r)}(\theta) \right\|^2$$

How simulation-based estimation is usually done

- **Rough idea:** Tune the parameters θ until some features F "look right"
- For (asymptotically) stationary time series, often use minimum distance estimator

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \left\| F^{\text{obs}} - \frac{1}{s} \sum_{r=1}^{s} F^{(r)}(\theta) \right\|^2$$

- For consistency $\hat{\theta} \xrightarrow{p} \theta_0$, need $F(\theta) \xrightarrow{p} \Phi(\theta)$ for all θ

How simulation-based estimation is usually done

- **Rough idea:** Tune the parameters θ until some features F "look right"
- For (asymptotically) stationary time series, often use minimum distance estimator

$$\hat{\theta} = \operatorname*{argmin}_{\theta \in \Theta} \left\| F^{\text{obs}} - \frac{1}{s} \sum_{r=1}^{s} F^{(r)}(\theta) \right\|^2$$

- For consistency $\hat{\theta} \xrightarrow{p} \theta_0$, need $F(\theta) \xrightarrow{p} \Phi(\theta)$ for all θ
- Hope Φ^{-1} exists and is nice, so we can translate from $\Phi(\theta)$ back to θ

How simulation-based estimation is usually done

- **Rough idea:** Tune the parameters θ until some features F "look right"
- For (asymptotically) stationary time series, often use minimum distance estimator

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\| F^{\text{obs}} - \frac{1}{s} \sum_{r=1}^{s} F^{(r)}(\theta) \right\|^2$$

- For consistency $\hat{\theta} \xrightarrow{p} \theta_0$, need $F(\theta) \xrightarrow{p} \Phi(\theta)$ for all θ
- Hope Φ^{-1} exists and is nice, so we can translate from $\Phi(\theta)$ back to θ
- For nonstationary time series, need rolling-window estimator (not discussed here)

Which features should we use?

- **Method of simulated moments:** F based on functions of moments

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$
  - Use $F = \hat{\beta}$ from AR(1): $X_t = \beta X_{t-1} + \xi_t$

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$
  - Use $F = \hat{\beta}$ from AR(1): $X_t = \beta X_{t-1} + \xi_t$
- **Synthetic likelihood:** F from user-chosen summary statistics

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$
  - Use $F = \hat{\beta}$ from AR(1): $X_t = \beta X_{t-1} + \xi_t$
- **Synthetic likelihood:** F from user-chosen summary statistics
  - Coefficients from polynomial regression of X_t on $X_{t-1}, \ldots, X_{t-\ell}$

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$
  - Use $F = \hat{\beta}$ from AR(1): $X_t = \beta X_{t-1} + \xi_t$
- **Synthetic likelihood:** F from user-chosen summary statistics
  - Coefficients from polynomial regression of X_t on $X_{t-1}, \ldots, X_{t-\ell}$
- **Neural summary statistics:** F from neural network

Which features should we use?

- **Method of simulated moments:** F based on functions of moments
  - Sample mean, variance, skewness, autocorrelation, and so on
- **Indirect inference:** F from parameter estimates in auxiliary model
  - To get $\hat{\theta}$ for MA(1): $X_t = \epsilon_t + \theta_0 \epsilon_{t-1}$
  - Use $F = \hat{\beta}$ from AR(1): $X_t = \beta X_{t-1} + \xi_t$
- **Synthetic likelihood:** F from user-chosen summary statistics
  - Coefficients from polynomial regression of X_t on $X_{t-1}, \ldots, X_{t-\ell}$
- **Neural summary statistics:** F from neural network
  - Extract early layer of neural network

Which features should we use?

- Good features are sensitive to small parameter changes

Which features should we use?

- Good features are sensitive to small parameter changes
- Takes time to find good features

Which features should we use?

- Good features are sensitive to small parameter changes
- Takes time to find good features
- Can we just use *random features* of the data instead?

Which features should we use?

- Good features are sensitive to small parameter changes
- Takes time to find good features
- Can we just use *random features* of the data instead?
- Automatic feature selection, computationally efficient, little knowledge needed

How many random features?

- **Answer:** Need $2p + 1$ random features for a p-dimensional parameter space Θ

How many random features?

- **Answer:** Need $2p + 1$ random features for a p-dimensional parameter space Θ
- Probabilistic Whitney (1936) embedding theorem from Sauer, Yorke, and Casdagli (1991): If Θ is compact, "almost every" C^1 function from Θ to $\mathbb{R}^{2p+1}$ is injective

How many random features?

- **Answer:** Need $2p + 1$ random features for a p-dimensional parameter space Θ
- Probabilistic Whitney (1936) embedding theorem from Sauer, Yorke, and Casdagli (1991): If Θ is compact, "almost every" C^1 function from Θ to $\mathbb{R}^{2p+1}$ is injective
- Let P_θ be the joint distribution of $[X_t(\theta), \ldots, X_{t-m}(\theta)]^\top$ for some $m \in \mathbb{N}$

How many random features?

- **Answer:** Need $2p + 1$ random features for a p-dimensional parameter space Θ
- Probabilistic Whitney (1936) embedding theorem from Sauer, Yorke, and Casdagli (1991): If Θ is compact, "almost every" C^1 function from Θ to $\mathbb{R}^{2p+1}$ is injective
- Let P_θ be the joint distribution of $[X_t(\theta), \ldots, X_{t-m}(\theta)]^\top$ for some $m \in \mathbb{N}$
- Under certain smoothness conditions on $\theta \mapsto P_\theta$, expectations of $2p + 1$ "generic" continuous, bounded functions $\varphi : \mathbb{R}^{(m+1)\times d} \to \mathbb{R}^{2p+1}$ will be C^1 smooth bijections with C^1 smooth inverse. These expectations $\theta \mapsto \Phi(\theta)$ are given by

$$\Phi(\theta) = \int_{\mathbb{R}^{(m+1)\times d}} \varphi(x) dP_\theta(x)$$

How many random features?

- **Answer:** Need $2p+1$ random features for a p-dimensional parameter space Θ
- Probabilistic Whitney (1936) embedding theorem from Sauer, Yorke, and Casdagli (1991): If Θ is compact, "almost every" C^1 function from Θ to $\mathbb{R}^{2p+1}$ is injective
- Let P_θ be the joint distribution of $[X_t(\theta), \ldots, X_{t-m}(\theta)]^\top$ for some $m \in \mathbb{N}$
- Under certain smoothness conditions on $\theta \mapsto P_\theta$, expectations of $2p+1$ "generic" continuous, bounded functions $\varphi : \mathbb{R}^{(m+1)\times d} \to \mathbb{R}^{2p+1}$ will be C^1 smooth bijections with C^1 smooth inverse. These expectations $\theta \mapsto \Phi(\theta)$ are given by

$$\Phi(\theta) = \int_{\mathbb{R}^{(m+1)\times d}} \varphi(x) dP_\theta(x)$$

- If physical dependence measure (Wu 2005) decays uniformly over Θ, we can estimate $\Phi(\theta)$, $\theta \in \Theta$, via time-averages of $2p+1$ random features

$$\frac{1}{n-m} \sum_{t=m+1}^{n} \varphi\left([X_t(\theta), \ldots, X_{t-m}(\theta)]^\top\right)$$

Example: Mean of Gaussian

- **Goal:** Estimate $\theta_0 = 0$ given $X_1^{\text{obs}}, \ldots, X_n^{\text{obs}} \overset{\text{iid}}{\sim} \mathcal{N}(\theta_0, 1)$ with $\Theta = [-3, 3]$

Example: Mean of Gaussian

- **Goal:** Estimate $\theta_0 = 0$ given $X_1^{\text{obs}}, \ldots, X_n^{\text{obs}} \overset{\text{iid}}{\sim} \mathcal{N}(\theta_0, 1)$ with $\Theta = [-3, 3]$
- Use time-averages of $2p + 1 = 3$ random Fourier features with $m = 0$ lags

Example: Mean of Gaussian

- **Goal:** Estimate $\theta_0 = 0$ given $X_1^{\text{obs}}, \ldots, X_n^{\text{obs}} \overset{\text{iid}}{\sim} \mathcal{N}(\theta_0, 1)$ with $\Theta = [-3, 3]$
- Use time-averages of $2p + 1 = 3$ random Fourier features with $m = 0$ lags
- For $i = 1, \ldots, 2p + 1$, randomly draw frequencies $w_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and phases $b_i \overset{\text{iid}}{\sim} \text{Unif}(-\pi, \pi)$, and define the i-th feature as $\varphi_i(x) = \cos(w_i x + b_i)$

Example: Mean of Gaussian

- **Goal:** Estimate $\theta_0 = 0$ given $X_1^{\text{obs}}, \ldots, X_n^{\text{obs}} \overset{\text{iid}}{\sim} \mathcal{N}(\theta_0, 1)$ with $\Theta = [-3, 3]$
- Use time-averages of $2p + 1 = 3$ random Fourier features with $m = 0$ lags
- For $i = 1, \ldots, 2p + 1$, randomly draw frequencies $w_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and phases $b_i \overset{\text{iid}}{\sim} \text{Unif}(-\pi, \pi)$, and define the i-th feature as $\varphi_i(x) = \cos(w_i x + b_i)$
- Denoting $\varphi = (\varphi_1, \ldots, \varphi_{2p+1})$, our estimator is given by

$$\hat{\theta} = \underset{\theta \in \Theta}{\text{argmin}} \left\| \frac{1}{n} \sum_{t=1}^{n} \varphi(X_t^{\text{obs}}) - \frac{1}{s} \sum_{r=1}^{s} \left(\frac{1}{n} \sum_{t=1}^{n} \varphi\left(X_t^{(r)}(\theta) \right) \right) \right\|_2$$

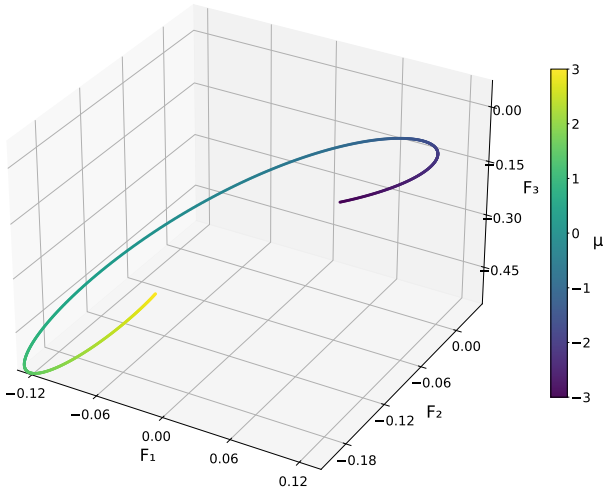
Example: Mean of Gaussian

- **Goal:** Estimate $\theta_0 = 0$ given $X_1^{\text{obs}}, \ldots, X_n^{\text{obs}} \overset{\text{iid}}{\sim} \mathcal{N}(\theta_0, 1)$ with $\Theta = [-3, 3]$
- Use time-averages of $2p + 1 = 3$ random Fourier features with $m = 0$ lags
- For $i = 1, \ldots, 2p + 1$, randomly draw frequencies $w_i \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ and phases $b_i \overset{\text{iid}}{\sim} \text{Unif}(-\pi, \pi)$, and define the i-th feature as $\varphi_i(x) = \cos(w_i x + b_i)$
- Denoting $\varphi = (\varphi_1, \ldots, \varphi_{2p+1})$, our estimator is given by

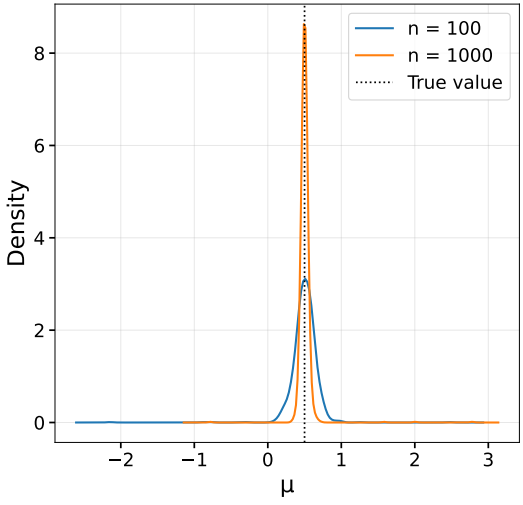
$$\hat{\theta} = \underset{\theta \in \Theta}{\text{argmin}} \left\| \frac{1}{n} \sum_{t=1}^{n} \varphi(X_t^{\text{obs}}) - \frac{1}{s} \sum_{r=1}^{s} \left(\frac{1}{n} \sum_{t=1}^{n} \varphi\left(X_t^{(r)}(\theta) \right) \right) \right\|_2$$

- We will compare our random feature estimator $\hat{\theta}$ with $\hat{\theta}^{\text{MLE}} = \frac{1}{n} \sum_{t=1}^{n} X_t$

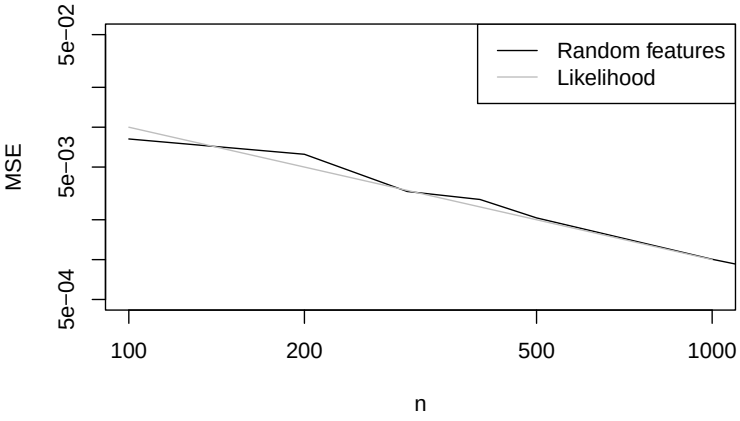
Time-average of 3 features, colored by parameter value ($n = 1000$, $s = 10$)



Density plot based on 1,000 independent estimates



Theoretical MSE of MLE vs MSE of our estimator (100 replicates per n)



Theoretical guarantees for random feature estimators

- We prove consistency $\hat{\theta} \xrightarrow{p} \theta_0$ and asymptotic normality

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \Rightarrow N\left(0, (\Gamma^\top \Gamma)^{-1} \Gamma^\top V \Gamma (\Gamma^\top \Gamma)^{-1}\right)$$

where Γ is the $(2p+1) \times p$ Jacobian matrix of Φ evaluated at θ_0, and V is the $(2p+1) \times (2p+1)$ asymptotic covariance matrix of $\frac{1}{n}\sum_{t=1}^{n} \varphi(X_t^{\text{obs}})$

Theoretical guarantees for random feature estimators

- We prove consistency $\hat{\theta} \xrightarrow{p} \theta_0$ and asymptotic normality

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \Rightarrow N\left(0, (\Gamma^\top \Gamma)^{-1} \Gamma^\top V \Gamma (\Gamma^\top \Gamma)^{-1}\right)$$

  where Γ is the $(2p+1) \times p$ Jacobian matrix of Φ evaluated at θ_0, and V is the $(2p+1) \times (2p+1)$ asymptotic covariance matrix of $\frac{1}{n}\sum_{t=1}^{n} \varphi(X_t^{\text{obs}})$
- Rolling-window estimator also introduced, and we prove analogous guarantees using the framework of locally stationary time series

Theoretical guarantees for random feature estimators

- We prove consistency $\hat{\theta} \xrightarrow{p} \theta_0$ and asymptotic normality

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \Rightarrow N\left(0, (\Gamma^\top \Gamma)^{-1} \Gamma^\top V \Gamma (\Gamma^\top \Gamma)^{-1}\right)$$

  where Γ is the $(2p+1) \times p$ Jacobian matrix of Φ evaluated at θ_0, and V is the $(2p+1) \times (2p+1)$ asymptotic covariance matrix of $\frac{1}{n}\sum_{t=1}^{n} \varphi(X_t^{\text{obs}})$
- Rolling-window estimator also introduced, and we prove analogous guarantees using the framework of locally stationary time series
- Examples: ODEs observed through dependent noise, nonstationary state-space models, and structural time series models

Paradigm shift in prediction: Time series foundation models (TSFMs)

- Inspired by success of transformer-based foundation models in language and vision

Paradigm shift in prediction: Time series foundation models (TSFMs)

- Inspired by success of transformer-based foundation models in language and vision
- Many TSFMs are *pre-trained* on synthetic time series generated by *parametric* time series models or *convex combinations* of such models

Paradigm shift in prediction: Time series foundation models (TSFMs)

- Inspired by success of transformer-based foundation models in language and vision
- Many TSFMs are *pre-trained* on synthetic time series generated by *parametric* time series models or *convex combinations* of such models
- Statistical foundations are unclear, especially for nonstationary time series

Paradigm shift in prediction: Time series foundation models (TSFMs)

- Inspired by success of transformer-based foundation models in language and vision
- Many TSFMs are *pre-trained* on synthetic time series generated by *parametric* time series models or *convex combinations* of such models
- Statistical foundations are unclear, especially for nonstationary time series
- We develop theory, and propose methods for smoothing and forecasting

Simulate to get pre-training data for forecasting with TSFMs

- For forecasting with TSFMs, Dooley et al. (2023) suggest training on simulations from multiplicative seasonality-trend-noise decomposition models (many $\theta \in \Theta$)

Simulate to get pre-training data for forecasting with TSFMs

- For forecasting with TSFMs, Dooley et al. (2023) suggest training on simulations from multiplicative seasonality-trend-noise decomposition models (many $\theta \in \Theta$)
- Consider a version with multiplicative autoregressive noise

$$Y_t = \mathrm{Se}\left(t, \theta^{Se}\right) \mathrm{Tr}\left(t, \theta^{Tr}\right) \exp\left(\kappa X_t - \frac{\kappa^2 \sigma_X^2}{2(1-\phi^2)}\right) \exp\left(\varepsilon_t^Y - \frac{\sigma_Y^2}{2}\right)$$
$$X_t = \phi X_{t-1} + \varepsilon_t^X$$

for some scaling factor $\kappa > 0$ and autoregressive coefficient $\phi \in (-1, 1)$, with $X_0 \sim N(0, \frac{\sigma_X^2}{1-\phi^2})$ and $\varepsilon_t^X \overset{\text{iid}}{\sim} N(0, \sigma_X^2)$, $\varepsilon_t^Y \overset{\text{iid}}{\sim} N(0, \sigma_Y^2)$ for some $\sigma_X, \sigma_Y > 0$

Simulate to get pre-training data for forecasting with TSFMs

- For forecasting with TSFMs, Dooley et al. (2023) suggest training on simulations from multiplicative seasonality-trend-noise decomposition models (many $\theta \in \Theta$)
- Consider a version with multiplicative autoregressive noise

$$Y_t = \mathrm{Se}\left(t, \theta^{Se}\right) \mathrm{Tr}\left(t, \theta^{Tr}\right) \exp\left(\kappa X_t - \frac{\kappa^2 \sigma_X^2}{2(1-\phi^2)}\right) \exp\left(\varepsilon_t^Y - \frac{\sigma_Y^2}{2}\right)$$
$$X_t = \phi X_{t-1} + \varepsilon_t^X$$

for some scaling factor $\kappa > 0$ and autoregressive coefficient $\phi \in (-1, 1)$, with $X_0 \sim N(0, \frac{\sigma_X^2}{1-\phi^2})$ and $\varepsilon_t^X \overset{\text{iid}}{\sim} N(0, \sigma_X^2)$, $\varepsilon_t^Y \overset{\text{iid}}{\sim} N(0, \sigma_Y^2)$ for some $\sigma_X, \sigma_Y > 0$

- Seasonality: linear combination of sines, cosines (weekly, monthly, yearly periods)

Simulate to get pre-training data for forecasting with TSFMs

- For forecasting with TSFMs, Dooley et al. (2023) suggest training on simulations from multiplicative seasonality-trend-noise decomposition models (many $\theta \in \Theta$)
- Consider a version with multiplicative autoregressive noise

$$Y_t = \mathrm{Se}\left(t, \theta^{Se}\right) \mathrm{Tr}\left(t, \theta^{Tr}\right) \exp\left(\kappa X_t - \frac{\kappa^2 \sigma_X^2}{2(1-\phi^2)}\right) \exp\left(\varepsilon_t^Y - \frac{\sigma_Y^2}{2}\right)$$
$$X_t = \phi X_{t-1} + \varepsilon_t^X$$

for some scaling factor $\kappa > 0$ and autoregressive coefficient $\phi \in (-1, 1)$, with $X_0 \sim N(0, \frac{\sigma_X^2}{1-\phi^2})$ and $\varepsilon_t^X \overset{\text{iid}}{\sim} N(0, \sigma_X^2)$, $\varepsilon_t^Y \overset{\text{iid}}{\sim} N(0, \sigma_Y^2)$ for some $\sigma_X, \sigma_Y > 0$

- Seasonality: linear combination of sines, cosines (weekly, monthly, yearly periods)
- Trend: product of linear trend and exponential trend

Simulate to get pre-training data for forecasting with TSFMs

- For forecasting with TSFMs, Dooley et al. (2023) suggest training on simulations from multiplicative seasonality-trend-noise decomposition models (many $\theta \in \Theta$)
- Consider a version with multiplicative autoregressive noise

$$Y_t = \mathrm{Se}\left(t, \theta^{Se}\right) \mathrm{Tr}\left(t, \theta^{Tr}\right) \exp\left(\kappa X_t - \frac{\kappa^2 \sigma_X^2}{2(1-\phi^2)}\right) \exp\left(\varepsilon_t^Y - \frac{\sigma_Y^2}{2}\right)$$
$$X_t = \phi X_{t-1} + \varepsilon_t^X$$

for some scaling factor $\kappa > 0$ and autoregressive coefficient $\phi \in (-1, 1)$, with $X_0 \sim N(0, \frac{\sigma_X^2}{1-\phi^2})$ and $\varepsilon_t^X \overset{\text{iid}}{\sim} N(0, \sigma_X^2)$, $\varepsilon_t^Y \overset{\text{iid}}{\sim} N(0, \sigma_Y^2)$ for some $\sigma_X, \sigma_Y > 0$

- Seasonality: linear combination of sines, cosines (weekly, monthly, yearly periods)
- Trend: product of linear trend and exponential trend
- Our rolling-window estimator can be used (consistent and asymptotically normal)

Unifying framework: Simulation-and-regression with parameter uncertainty

- We propose minimizing the Monte Carlo approximation to

$$R_{P_{\widehat{\text{mix}}}}(f) = \mathbb{E}_{P_{\widehat{\text{mix}}}}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right] = \mathbb{E}_{\theta\sim\widehat{\text{CD}}_n}\left[\mathbb{E}_{P_\theta}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right]\right]$$

for some loss L, where the expectation is w.r.t. the mixture distribution $P_{\widehat{\text{mix}}} = \int_\Theta P_\theta d\widehat{\text{CD}}_n(\theta)$ of sequences X, Y corresponding to some *estimated confidence distribution* $\widehat{\text{CD}}_n$ over $\Theta \subset \mathbb{R}^p$ for θ_0 (D. Liu, R. Y. Liu, and Xie 2022)

Unifying framework: Simulation-and-regression with parameter uncertainty

- We propose minimizing the Monte Carlo approximation to

$$R_{P_{\widehat{\text{mix}}}}(f) = \mathbb{E}_{P_{\widehat{\text{mix}}}}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right] = \mathbb{E}_{\theta\sim\widehat{\text{CD}}_n}\left[\mathbb{E}_{P_\theta}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right]\right]$$

  for some loss L, where the expectation is w.r.t. the mixture distribution $P_{\widehat{\text{mix}}} = \int_\Theta P_\theta d\widehat{\text{CD}}_n(\theta)$ of sequences X, Y corresponding to some *estimated confidence distribution* $\widehat{\text{CD}}_n$ over $\Theta \subset \mathbb{R}^p$ for θ_0 (D. Liu, R. Y. Liu, and Xie 2022)
- When $\widehat{\text{CD}}_n$ is uniform over Θ, this is like TSFM pre-training (no information)

Unifying framework: Simulation-and-regression with parameter uncertainty

- We propose minimizing the Monte Carlo approximation to

$$R_{P_{\widehat{\text{mix}}}}(f) = \mathbb{E}_{P_{\widehat{\text{mix}}}}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right] = \mathbb{E}_{\theta\sim\widehat{\text{CD}}_n}\left[\mathbb{E}_{P_\theta}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f(Y)]_t\right)\right]\right]$$

  for some loss L, where the expectation is w.r.t. the mixture distribution $P_{\widehat{\text{mix}}} = \int_\Theta P_\theta d\widehat{\text{CD}}_n(\theta)$ of sequences X, Y corresponding to some *estimated confidence distribution* $\widehat{\text{CD}}_n$ over $\Theta \subset \mathbb{R}^p$ for θ_0 (D. Liu, R. Y. Liu, and Xie 2022)
- When $\widehat{\text{CD}}_n$ is uniform over Θ, this is like TSFM pre-training (no information)
- When $\widehat{\text{CD}}_n$ is Dirac delta at $\hat{\theta}$, this is like classical simulation-and-regression, which aims to minimize the Monte Carlo approximation to

$$R_{P_{\hat{\theta}}}(f) = \mathbb{E}_{P_{\hat{\theta}}}\left[\sum_{t\in\mathcal{T}} L(X_t - [f(Y)]_t)\right]$$

  where expectation is w.r.t. distribution $P_{\hat{\theta}}$ of sequences X, Y corresponding to $\hat{\theta}$

Questions we aim to answer

- **Calibration:** Does accounting for parameter uncertainty improve the calibration of pre-trained/untrained transformers that output multiple conditional quantiles?

Questions we aim to answer

- **Calibration:** Does accounting for parameter uncertainty improve the calibration of pre-trained/untrained transformers that output multiple conditional quantiles?
- **Pre-training:** When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?

Questions we aim to answer

- **Calibration:** Does accounting for parameter uncertainty improve the calibration of pre-trained/untrained transformers that output multiple conditional quantiles?
- **Pre-training:** When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?
- **Fine-tuning and simulation-and-regression:** When does accounting for parameter uncertainty improve on simulation-and-regression methods and TSFM fine-tuning methods that are based on parameter point estimates?

Thank you!

Questions or comments?

You can reach me at:
mwiecksosa@cmu.edu

Does accounting for parameter uncertainty improve the calibration of pre-trained/untrained transformers that output conditional quantiles?

- For untrained transformers, see answer to question: "In the context of simulation-and-regression, does accounting for parameter uncertainty achieve better performance than using parameter estimate?"

Does accounting for parameter uncertainty improve the calibration of pre-trained/untrained transformers that output conditional quantiles?

- For untrained transformers, see answer to question: "In the context of simulation-and-regression, does accounting for parameter uncertainty achieve better performance than using parameter estimate?"
- For pre-trained transformers, see answer to question: "When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?"

In the context of simulation-and-regression, does accounting for parameter uncertainty achieve better performance than using parameter estimate?

- When does this inequality hold:

$$\mathbb{E}_{P_{\theta_0}}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f^*_{P_{\widehat{\text{mix}}}}(Y)]_t\right)\right] < \mathbb{E}_{P_{\theta_0}}\left[\sum_{t\in\mathcal{T}} L\left(X_t - [f^*_{P_{\hat{\theta}}}(Y)]_t\right)\right]$$

where both expectations are with respect to the distribution P_{θ_0} of the sequences corresponding to the unknown true parameter θ_0, $f^*_{P_{\widehat{\text{mix}}}}$ is the minimizer in the function class of $R_{P_{\widehat{\text{mix}}}}(f)$, and $f^*_{P_{\hat{\theta}}}$ is the minimizer in the function class of $R_{P_{\hat{\theta}}}(f)$

When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?

- Intuitively, this occurs when the following conditions hold

When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?

- Intuitively, this occurs when the following conditions hold
- Large parameter uncertainty, e.g., when the sample size of the observed time series is small relative to the noise

When does a pre-trained transformer fine-tuned with our method outperform a transformer trained from scratch with our method?

- Intuitively, this occurs when the following conditions hold
- Large parameter uncertainty, e.g., when the sample size of the observed time series is small relative to the noise
- The distributions used in the pre-training phase are close to the true distribution, e.g., when we have domain knowledge or we have previously observed time series with similar distributions

References I

- Bodik, Juraj and Olivier C. Pasche (2024). "Granger causality in extremes". arXiv preprint arXiv: 2407.09632.
- Dooley, Samuel et al. (2023). "Forecastpfn: Synthetically-trained zero-shot forecasting". In: *Advances in Neural Information Processing Systems* 36, pp. 2403–2426.
- Liu, Dungang, Regina Y. Liu, and Min-ge Xie (2022). "Nonparametric fusion learning for multiparameters: Synthesize inferences from diverse sources using data depth and confidence distribution". In: *Journal of the American Statistical Association* 117.540, pp. 2086–2104.
- Sauer, Tim, James A. Yorke, and Martin Casdagli (1991). "Embedology". In: *Journal of Statistical Physics* 65.3, pp. 579–616.

References II

- Shah, Rajen D. and Jonas Peters (2020). "The hardness of conditional independence testing and the generalised covariance measure". In: *Annals of Statistics* 48.3, pp. 1514–1538.
- Whitney, Hassler (1936). "Differentiable manifolds". In: *Annals of Mathematics* 37.3, pp. 645–680.
- Wu, Wei Biao (2005). "Nonlinear system theory: another look at dependence". In: *Proceedings of the National Academy of Sciences* 102.40, pp. 14150–14154.