Stream SGD

Convergence and Statistical Inference for Simulation-Based Decision Problems

One trajectory: learn the decision and its confidence interval

Xiang Li

University of Pennsylvania

MS57: Advances in Simulation-Based Optimization and Approximation Methods for Data Science (Part III)

SIAM Conference on Mathematics of Data Science (MDS26) · Salt Lake City · November 2026

Joint work with



Xiang Li

Incoming Asst. Prof.,
Statistics, Rutgers



Jiadong Liang

Wharton, UPenn



Xinyun Chen

CUHK-Shenzhen



Zhihua Zhang

Peking University

Main: *Operations Research* 2025, arXiv:2309.09545 · **Exogenous data:** arXiv:2302.07690

Simulation-based decision problems

Choose a design ϑ for a stochastic system that we can only **run** (simulate or operate), not solve.

$$\min_{\vartheta \in \Theta} \; f(\vartheta) = \mathbb{E}_{x \sim \pi_\vartheta}\big[\ell(\vartheta, x)\big], \qquad \pi_\vartheta = \text{steady state of the system run under } \vartheta.$$

EXAMPLES

- Queues: price and capacity
- Inventory: base-stock level
- Reinforcement learning: a policy

Simulation-based decision problems

Choose a design ϑ for a stochastic system that we can only **run** (simulate or operate), not solve.

$$\min_{\vartheta \in \Theta} f(\vartheta) = \mathbb{E}_{x \sim \pi_\vartheta}\big[\ell(\vartheta, x)\big], \qquad \pi_\vartheta = \text{steady state of the system run under } \vartheta.$$

EXAMPLES

- Queues: price and capacity
- Inventory: base-stock level
- Reinforcement learning: a policy

WHAT WE CAN DO

- Run the system under the current ϑ_t: $x_t \sim \mathcal{P}_{\vartheta_t}(x_{t-1}, \cdot)$, one **trajectory**.
- Build a gradient estimate $H(\vartheta_t, x_t)$.

Simulation-based decision problems

Choose a design ϑ for a stochastic system that we can only **run** (simulate or operate), not solve.

$$\min_{\vartheta \in \Theta} f(\vartheta) = \mathbb{E}_{x \sim \pi_\vartheta}\big[\ell(\vartheta, x)\big], \qquad \pi_\vartheta = \text{steady state of the system run under } \vartheta.$$

EXAMPLES

- Queues: price and capacity
- Inventory: base-stock level
- Reinforcement learning: a policy

WHAT WE CAN DO

- Run the system under the current ϑ_t: $x_t \sim \mathcal{P}_{\vartheta_t}(x_{t-1}, \cdot)$, one **trajectory**.
- Build a gradient estimate $H(\vartheta_t, x_t)$.

Why it is hard: we learn while the system runs

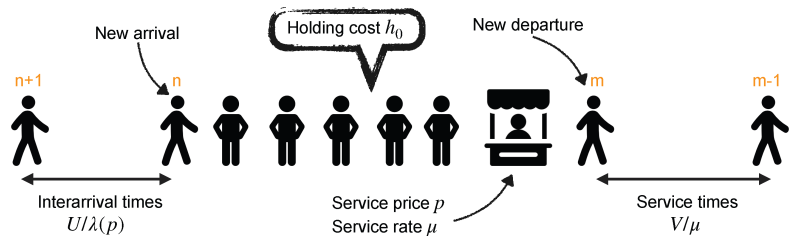
Biased: H has mean ∇f only in steady state.

Correlated: Markov data remember the past.

Moving target: the data change with ϑ_t.

Classical fix: **wait** for steady state (Chen et al., 2023). This talk: **the Poisson equation**.

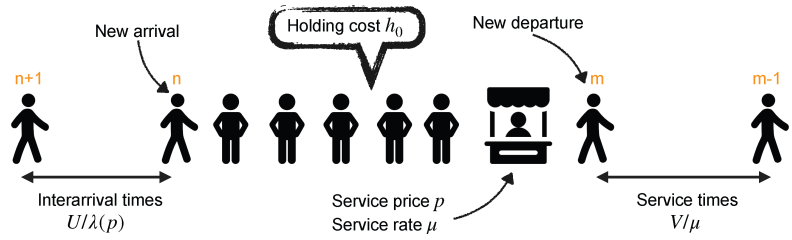
Running example: price and staff a service system



$$\min_{\vartheta=(\mu,p)} \underbrace{f(\vartheta)}_{\text{long-run cost rate}} = \underbrace{h_0\,\mathbb{E}\,Q_\infty(\mu,p)}_{\text{holding cost}} + \underbrace{c(\mu)}_{\text{staffing}} - \underbrace{p\,\lambda(p)}_{\text{revenue}}$$

- Q_∞: number of customers in the system, in steady state. No closed form in general.
- **GI/GI/1**: i.i.d. interarrival and service times, one server. **M/M/1**: both exponential.

Running example: price and staff a service system

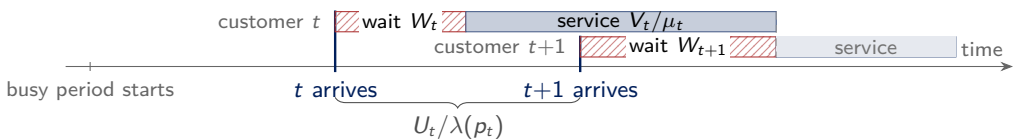


$$\min_{\vartheta=(\mu,p)} \underbrace{f(\vartheta)}_{\text{long-run cost rate}} = \underbrace{h_0\,\mathbb{E}\,Q_\infty(\mu,p)}_{\text{holding cost}} + \underbrace{c(\mu)}_{\text{staffing}} - \underbrace{p\,\lambda(p)}_{\text{revenue}}$$

- Q_∞: number of customers in the system, in steady state. No closed form in general.
- **GI/GI/1**: i.i.d. interarrival and service times, one server. **M/M/1**: both exponential.
- **Learn while operating**: after each customer, observe its wait and update (μ, p).

Learning while the system runs: stream SGD

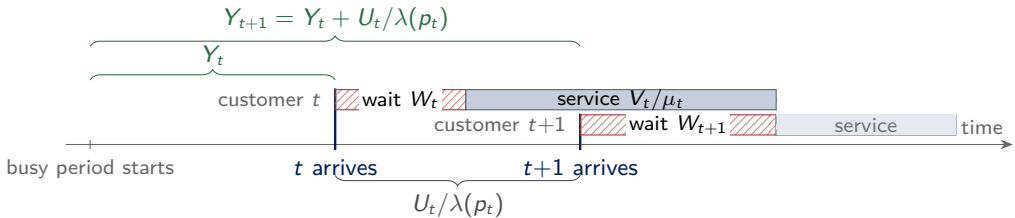
Under $\vartheta_t = (\mu_t, p_t)$ record W_t, customer t's **wait**, and Y_t, the **server**'s busy time so far.



Wait (Lindley): $W_{t+1} = \left(W_t + V_t/\mu_t - U_t/\lambda(p_t)\right)_+$

Learning while the system runs: stream SGD

Under $\vartheta_t = (\mu_t, p_t)$ record W_t, customer t's **wait**, and Y_t, the **server**'s busy time so far.

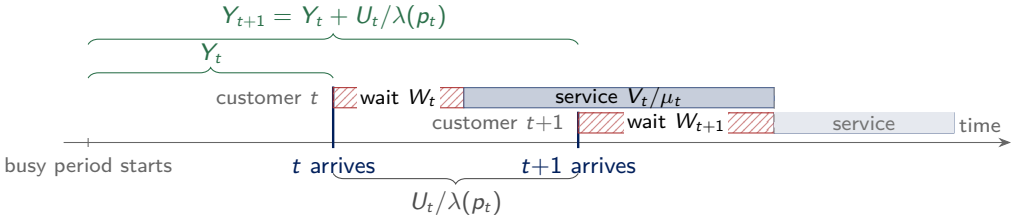


Wait (Lindley): $W_{t+1} = \left(W_t + V_t/\mu_t - U_t/\lambda(p_t)\right)_+$

Busy time: $Y_{t+1} = \left(Y_t + U_t/\lambda(p_t)\right) \mathbf{1}\{W_{t+1} > 0\}$ (reset to 0 if empty)

Learning while the system runs: stream SGD

Under $\vartheta_t = (\mu_t, p_t)$ record W_t, customer t's **wait**, and Y_t, the **server**'s busy time so far.



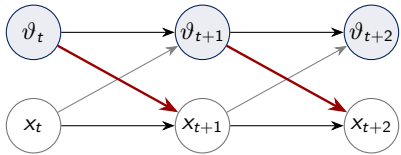
Wait (Lindley): $W_{t+1} = \left(W_t + V_t/\mu_t - U_t/\lambda(p_t)\right)_+$

Busy time: $Y_{t+1} = \left(Y_t + U_t/\lambda(p_t)\right) \mathbf{1}\{W_{t+1} > 0\}$ (reset to 0 if empty)

Update: $\vartheta_{t+1} = \Pi_{\mathcal{B}}\left(\vartheta_t - \eta_t\, H(\vartheta_t, x_t)\right)$, $x_t = (W_t, Y_t)$. Batch size 1 (earlier: $\asymp \log t$).

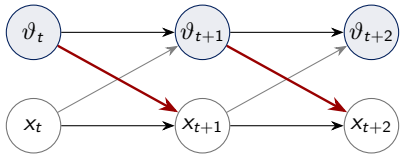
Gradient estimate: $\mathbb{E}_{\pi_\vartheta} H(\vartheta, x) = \nabla f(\vartheta)$: **unbiased only in steady state**.

Our questions: convergence and inference



Red: $x_{t+1} \sim \mathcal{P}_{\vartheta_t}(x_t, \cdot)$
the decision drives the data

Our questions: convergence and inference

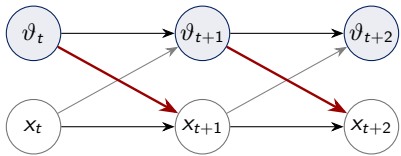


Red: $x_{t+1} \sim \mathcal{P}_{\vartheta_t}(x_t, \cdot)$
the decision drives the data

Q1. Convergence

Biased gradients; **Markov** data moving with ϑ_t.
Does SGD with **batch size one** converge? Rate? Regret?

Our questions: convergence and inference



Red: $x_{t+1} \sim \mathcal{P}_{\vartheta_t}(x_t, \cdot)$
the decision drives the data

Q1. Convergence

Biased gradients; **Markov** data moving with ϑ_t.
Does SGD with **batch size one** converge? Rate? Regret?

Q2. Inference

From **one run**: a confidence interval for $\vartheta^\star$ (e.g. $p^\star$), online, $O(1)$ memory, **no covariance estimate**?

The same framework covers more than queues

DATA DEPEND ON THE DECISION (Li et al., 2025)

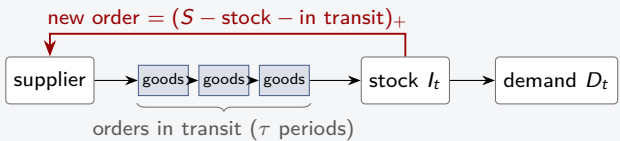
Queue: price and staff

GI/GI/1: set price p and service rate μ from each customer's wait.

Today's main example.

Inventory with lead time τ

base-stock level S, lost sales



EXOGENOUS MARKOV DATA (Li et al., 2023)

Q-learning

Estimate $Q^\star$ from one behavior trajectory (tabular, linear, soft).

SGD on Markov data

Logistic regression with Markov (not i.i.d.) covariates.

LoRA fine-tuning

Train $\mathbf{W} = \mathbf{BA}$ on Markov data; infer the product $\mathbf{W}$.

2

Simulation-Based Decisions

Convergence

Non-stationary data, the Poisson equation, and a functional CLT

Inference from the Path

Applications

The Poisson equation: bias you can sum

For each ϑ, let U_ϑ solve

$$U_\vartheta - \mathcal{P}_\vartheta U_\vartheta = H(\vartheta, \cdot) - \nabla f(\vartheta), \qquad \text{i.e.} \quad U_\vartheta(x) = \sum_{k \geq 0} (\mathcal{P}_\vartheta^k H(\vartheta, \cdot)(x) - \nabla f(\vartheta)).$$

- $U_\vartheta(x)$: total future bias of the gradient estimate if the chain starts at x. Finite if the chain forgets its start.

The Poisson equation: bias you can sum

For each ϑ, let U_ϑ solve

$$U_\vartheta - \mathcal{P}_\vartheta U_\vartheta = H(\vartheta, \cdot) - \nabla f(\vartheta), \qquad \text{i.e.} \quad U_\vartheta(x) = \sum_{k \geq 0} (\mathcal{P}_\vartheta^k H(\vartheta, \cdot)(x) - \nabla f(\vartheta)).$$

- $U_\vartheta(x)$: total future bias of the gradient estimate if the chain starts at x. Finite if the chain forgets its start.

Then the noise $\xi_t = H(\vartheta_t, x_t) - \nabla f(\vartheta_t)$ splits into three pieces:

$$\underbrace{U_{\vartheta_t}(x_t) - \mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_{t-1})}_{\text{martingale difference}} + \underbrace{\mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_{t-1}) - \mathcal{P}_{\vartheta_{t+1}} U_{\vartheta_{t+1}}(x_t)}_{\text{telescopes}} + \underbrace{\mathcal{P}_{\vartheta_{t+1}} U_{\vartheta_{t+1}}(x_t) - \mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_t)}_{\text{moving kernel}}$$

The Poisson equation: bias you can sum

For each ϑ, let U_ϑ solve

$$U_\vartheta - \mathcal{P}_\vartheta U_\vartheta = H(\vartheta, \cdot) - \nabla f(\vartheta), \qquad \text{i.e.} \quad U_\vartheta(x) = \sum_{k \geq 0} (\mathcal{P}_\vartheta^k H(\vartheta, \cdot)(x) - \nabla f(\vartheta)).$$

- $U_\vartheta(x)$: total future bias of the gradient estimate if the chain starts at x. Finite if the chain forgets its start.

Then the noise $\xi_t = H(\vartheta_t, x_t) - \nabla f(\vartheta_t)$ splits into three pieces:

$$\underbrace{U_{\vartheta_t}(x_t) - \mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_{t-1})}_{\text{martingale difference}} + \underbrace{\mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_{t-1}) - \mathcal{P}_{\vartheta_{t+1}} U_{\vartheta_{t+1}}(x_t)}_{\text{telescopes}} + \underbrace{\mathcal{P}_{\vartheta_{t+1}} U_{\vartheta_{t+1}}(x_t) - \mathcal{P}_{\vartheta_t} U_{\vartheta_t}(x_t)}_{\text{moving kernel}}$$

- The last piece exists **only** because the kernel moves with ϑ. It is small if $\vartheta \mapsto \mathcal{P}_\vartheta U_\vartheta$ is Lipschitz.

How each piece is controlled

Error $\Delta_t = \vartheta_t - \vartheta^\star$, gradient noise $\xi_t = H(\vartheta_t, x_t) - \nabla f(\vartheta_t)$, strong convexity K_0:

$$\mathbb{E}\|\Delta_{t+1}\|^2 \le (1 - 2K_0\eta_t)\,\mathbb{E}\|\Delta_t\|^2 \underbrace{- 2\eta_t\,\mathbb{E}\langle \Delta_t, \xi_t\rangle}_{\substack{\text{gradient bias} \\ \neq 0 \text{ for Markov data}}} + \underbrace{C\eta_t^2}_{\text{variance}} .$$

How each piece is controlled

Error $\Delta_t = \vartheta_t - \vartheta^\star$, gradient noise $\xi_t = H(\vartheta_t, x_t) - \nabla f(\vartheta_t)$, strong convexity K_0:

$$\mathbb{E}\|\Delta_{t+1}\|^2 \le (1 - 2K_0\eta_t)\,\mathbb{E}\|\Delta_t\|^2 \underbrace{- 2\eta_t\,\mathbb{E}\langle \Delta_t, \xi_t\rangle}_{\substack{\text{gradient bias} \\ \neq 0 \text{ for Markov data}}} + \underbrace{C\eta_t^2}_{\text{variance}} .$$

Piece of ξ_t	**Why it is small**	**Cost in $\eta_t \mathbb{E}\langle \Delta_t, \xi_t\rangle$**
martingale	mean zero given the past	0
telescoping	sum by parts; Δ_t moves by $O(\eta_t)$	$O(\eta_t^2)$
moving kernel	$\|\vartheta_{t+1} - \vartheta_t\| = O(\eta_t)$	$O(\eta_t^2)$

How each piece is controlled

Error $\Delta_t = \vartheta_t - \vartheta^\star$, gradient noise $\xi_t = H(\vartheta_t, x_t) - \nabla f(\vartheta_t)$, strong convexity K_0:

$$\mathbb{E}\|\Delta_{t+1}\|^2 \le (1 - 2K_0\eta_t)\,\mathbb{E}\|\Delta_t\|^2 \underbrace{- 2\eta_t\,\mathbb{E}\langle \Delta_t, \xi_t\rangle}_{\substack{\text{gradient bias} \\ \neq 0 \text{ for Markov data}}} + \underbrace{C\eta_t^2}_{\text{variance}} .$$

Piece of ξ_t	**Why it is small**	**Cost in $\eta_t\mathbb{E}\langle\Delta_t, \xi_t\rangle$**
martingale	mean zero given the past	0
telescoping	sum by parts; Δ_t moves by $O(\eta_t)$	$O(\eta_t^2)$
moving kernel	$\|\vartheta_{t+1} - \vartheta_t\| = O(\eta_t)$	$O(\eta_t^2)$

- One-sample bias is $O(1)$ but **averages out**: total cost $O(\eta_t^2)$, like the variance.
- Hence $\mathbb{E}\|\Delta_t\|^2 = O(\eta_t)$, with no batching.

When does the Poisson equation have a good solution?

The proof needs: U_ϑ exists, and U_ϑ, $\mathcal{P}_\vartheta U_\vartheta$ are Lipschitz **in** x **and in** ϑ.

Queue states are unbounded, so total variation is too strong. We measure distance by

$$\mathcal{W}_{\rho,V}(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_\gamma\big[\rho(X,Y)\,\{1+V(X)+V(Y)\}\big].$$

When does the Poisson equation have a good solution?

The proof needs: U_ϑ exists, and U_ϑ, $\mathcal{P}_\vartheta U_\vartheta$ are Lipschitz **in** x **and in** ϑ.

Queue states are unbounded, so total variation is too strong. We measure distance by

$$\mathcal{W}_{\rho,V}(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_\gamma\big[\rho(X,Y)\,\{1+V(X)+V(Y)\}\big].$$

- **Contraction**: $\mathcal{W}_{\rho,V}(\delta_x\mathcal{P}_\vartheta^n, \delta_y\mathcal{P}_\vartheta^n) \le Ce^{-cn}\rho(x,y)\{\ldots\}$, uniformly in ϑ.
  $\Rightarrow$ the series $\sum_k(\mathcal{P}_\vartheta^k H - \nabla f)$ converges: U_ϑ **exists, Lipschitz in** x.

When does the Poisson equation have a good solution?

The proof needs: U_ϑ exists, and U_ϑ, $\mathcal{P}_\vartheta U_\vartheta$ are Lipschitz **in** x **and in** ϑ.

Queue states are unbounded, so total variation is too strong. We measure distance by

$$\mathcal{W}_{\rho,V}(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_\gamma\big[\rho(X,Y)\,\{1+V(X)+V(Y)\}\big].$$

- **Contraction**: $\mathcal{W}_{\rho,V}(\delta_x\mathcal{P}_\vartheta^n, \delta_y\mathcal{P}_\vartheta^n) \le Ce^{-cn}\rho(x,y)\{\ldots\}$, uniformly in ϑ.
  $\Rightarrow$ the series $\sum_k(\mathcal{P}_\vartheta^k H - \nabla f)$ converges: U_ϑ **exists, Lipschitz in** x.
- **Kernel continuity**: $\mathcal{W}(\delta_x\mathcal{P}_\vartheta, \delta_x\mathcal{P}_{\vartheta'}) \le L\|\vartheta-\vartheta'\|\,(1+V(x)^2)$.
  $\Rightarrow$ U_ϑ, $\mathcal{P}_\vartheta U_\vartheta$ **Lipschitz in** ϑ: controls the moving-kernel piece.

When does the Poisson equation have a good solution?

The proof needs: U_ϑ exists, and U_ϑ, $\mathcal{P}_\vartheta U_\vartheta$ are Lipschitz **in** x **and in** ϑ.

Queue states are unbounded, so total variation is too strong. We measure distance by

$$\mathcal{W}_{\rho,V}(\mu,\nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \mathbb{E}_\gamma\big[\rho(X,Y)\,\{1+V(X)+V(Y)\}\big].$$

- **Contraction**: $\mathcal{W}_{\rho,V}(\delta_x\mathcal{P}_\vartheta^n, \delta_y\mathcal{P}_\vartheta^n) \le Ce^{-cn}\rho(x,y)\{\ldots\}$, uniformly in ϑ.
  $\Rightarrow$ the series $\sum_k(\mathcal{P}_\vartheta^k H - \nabla f)$ converges: U_ϑ **exists, Lipschitz in** x.
- **Kernel continuity**: $\mathcal{W}(\delta_x\mathcal{P}_\vartheta, \delta_x\mathcal{P}_{\vartheta'}) \le L\|\vartheta-\vartheta'\|\,(1+V(x)^2)$.
  $\Rightarrow$ U_ϑ**,** $\mathcal{P}_\vartheta U_\vartheta$ **Lipschitz in** ϑ: controls the moving-kernel piece.

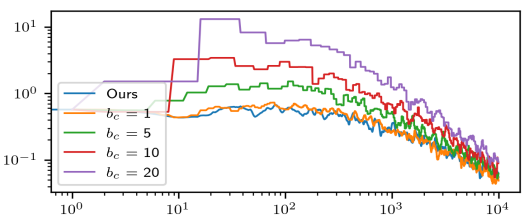
Both checked by coupling: two queues, same customers, merge once both are empty.

One customer per update is enough

Batch methods wait $1 + b_c \log t$ customers per update (Chen et al., 2023); "Ours" uses **one**.

Error

$\mathbb{E}\|\vartheta_t - \vartheta^\star\|^2 = O(1/t)$ (optimal rate)



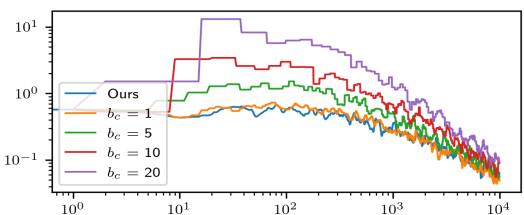
error vs. customers t (log-log); M/M/1, 200 runs

One customer per update is enough

Batch methods wait $1 + b_c \log t$ customers per update (Chen et al., 2023); "Ours" uses **one**.

Error

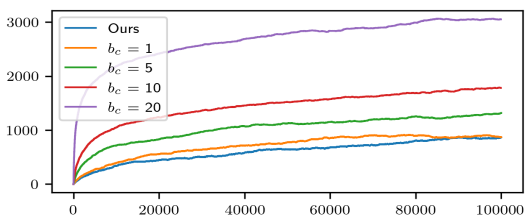
$$\mathbb{E}\|\vartheta_t - \vartheta^\star\|^2 = O(1/t) \quad \text{(optimal rate)}$$



error vs. customers t (log-log); M/M/1, 200 runs

Regret

$$R(T) = \mathbb{E}\sum_{t \le T}\big(\underbrace{\boxed{\text{cost}_t}}_{\text{cost rate under } \vartheta_t} - f(\vartheta^\star)\big)\underbrace{\boxed{l_t}}_{\text{cycle length}} = O(\log T)$$



cumulative regret vs. customers t

From one number to a whole path

CLT for $\bar{\vartheta}_T$: variance $G^{-1}SG^{-\top}$ (Hessian and long-run covariance: hard online).

From one number to a whole path

CLT for $\bar{\vartheta}_T$: variance $G^{-1}SG^{-\top}$ (Hessian and long-run covariance: hard online).

Theorem (FCLT) (Li et al., 2025)

Partial-sum path $\phi_T(r) = \dfrac{1}{\sqrt{T}} \displaystyle\sum_{t \le Tr} (\vartheta_t - \vartheta^\star), \quad r \in [0,1]$. Then

$$\phi_T(\cdot) \xrightarrow{d} G^{-1} S^{1/2} \mathcal{W}(\cdot) \quad \text{in } D[0,1], \qquad \mathcal{W} = \text{standard Brownian motion.}$$

From one number to a whole path

CLT for $\bar{\vartheta}_T$: variance $G^{-1}SG^{-\top}$ (Hessian and long-run covariance: hard online).

Theorem (FCLT) (Li et al., 2025)

Partial-sum path $\phi_T(r) = \dfrac{1}{\sqrt{T}} \sum_{t \le Tr} (\vartheta_t - \vartheta^\star)$, $r \in [0,1]$. Then

$$\phi_T(\cdot) \xrightarrow{d} G^{-1} S^{1/2} \mathcal{W}(\cdot) \quad \text{in } D[0,1], \qquad \mathcal{W} = \text{standard Brownian motion.}$$

- The running sum behaves like a Brownian motion with **one unknown scale**.
- The same holds for exogenous Markov data (Li et al., 2023).

Why the FCLT holds: the idea

Near $\vartheta^\star$, $\nabla f(\vartheta) \approx G\Delta$. Rearranging the update,

$$G\,\Delta_t = \frac{\Delta_t - \Delta_{t+1}}{\eta_t} - \underbrace{\varepsilon_t}_{\text{martingale}} - \underbrace{r_t}_{\text{remainders}}.$$

Why the FCLT holds: the idea

Near $\vartheta^\star$, $\nabla f(\vartheta) \approx G\Delta$. Rearranging the update,

$$G\,\Delta_t = \frac{\Delta_t - \Delta_{t+1}}{\eta_t} - \underbrace{\varepsilon_t}_{\text{martingale}} - \underbrace{r_t}_{\text{remainders}} .$$

Sum over $t \le Tr$ and divide by $\sqrt{T}$:

$$G\,\phi_T(r) = \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} \frac{\Delta_t - \Delta_{t+1}}{\eta_t}}_{\text{sum by parts} \to 0} - \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} \varepsilon_t}_{\xrightarrow{d}\ S^{1/2}\mathcal{W}(r)} - \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} r_t}_{\to 0} .$$

Why the FCLT holds: the idea

Near $\vartheta^\star$, $\nabla f(\vartheta) \approx G\Delta$. Rearranging the update,

$$G\,\Delta_t = \frac{\Delta_t - \Delta_{t+1}}{\eta_t} - \underbrace{\varepsilon_t}_{\text{martingale}} - \underbrace{r_t}_{\text{remainders}} .$$

Sum over $t \le Tr$ and divide by $\sqrt{T}$:

$$G\,\phi_T(r) = \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} \frac{\Delta_t - \Delta_{t+1}}{\eta_t}}_{\text{sum by parts} \to 0} - \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} \varepsilon_t}_{\xrightarrow{d}\ S^{1/2}\mathcal{W}(r)} - \underbrace{\frac{1}{\sqrt{T}} \sum_{t \le Tr} r_t}_{\to 0} .$$

- Middle term: martingale FCLT. Its covariance S comes from the Poisson solution at $\vartheta^\star$.
- Outer terms vanish **uniformly in** r: needs $\mathbb{E}\|\Delta_t\|^2 = O(\eta_t)$ and $\sum_t \eta_t^{1.5} < \infty$.

3

Simulation-Based Decisions

Convergence

Inference from the Path

Cancel the unknown scale, and pay as little as possible for it

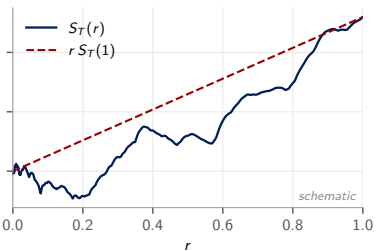
Applications

What we can compute from the path

Direction v (e.g. the price), $Y_t = v^\top \vartheta_t$. Computable partial sums:

$$S_T(r) = \frac{1}{\sqrt{T}} \sum_{t \le Tr} Y_t = \underbrace{\sqrt{T}\, r\, v^\top \vartheta^\star}_{\text{line, unknown slope}} + \underbrace{v^\top \phi_T(r)}_{\approx \sigma_v \mathcal{W}(r)}.$$

Partial-sum path



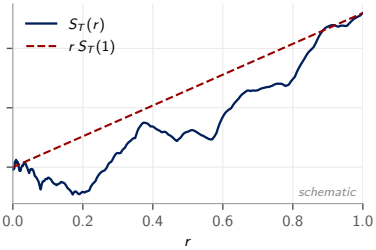
A line with unknown slope, plus fluctuation.

What we can compute from the path

Direction v (e.g. the price), $Y_t = v^\top \vartheta_t$. Computable partial sums:

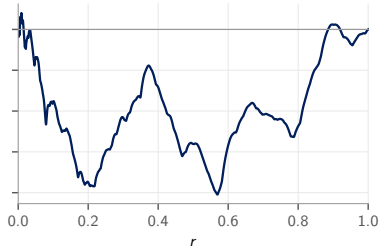
$$S_T(r) = \frac{1}{\sqrt{T}} \sum_{t \le Tr} Y_t = \underbrace{\sqrt{T}\, r\, v^\top \vartheta^\star}_{\text{line, unknown slope}} + \underbrace{v^\top \phi_T(r)}_{\approx \sigma_v \mathcal{W}(r)}.$$

Partial-sum path



A line with unknown slope, plus fluctuation.

Bridge: subtract the chord



$B_T(r) = S_T(r) - r\,S_T(1) \xrightarrow{d} \sigma_v \mathcal{B}(r)$: no $\vartheta^\star$.

Cancel the scale: a self-normalized pivot

Recall: bridge $B_T(r) = S_T(r) - r\,S_T(1) \xrightarrow{d} \sigma_v \mathcal{B}(r)$, computable, no $\vartheta^\star$.

Normalizer $\mathcal{D}$: continuous, $\mathcal{D}(cg) = |c|\,\mathcal{D}(g)$, $\mathcal{D}(\mathcal{B}) > 0$.

Cancel the scale: a self-normalized pivot

Recall: bridge $B_T(r) = S_T(r) - r\,S_T(1) \xrightarrow{d} \sigma_v \mathcal{B}(r)$, computable, no $\vartheta^\star$.

Normalizer $\mathcal{D}$: continuous, $\mathcal{D}(cg) = |c|\,\mathcal{D}(g)$, $\mathcal{D}(\mathcal{B}) > 0$.

$$\frac{\sqrt{T}\,(\bar{Y}_T - v^\top \vartheta^\star)}{\mathcal{D}(B_T)} \xrightarrow{d} \frac{\sigma_v\,\mathcal{W}(1)}{\sigma_v\,\mathcal{D}(\mathcal{B})} = \frac{\mathcal{W}(1)}{\mathcal{D}(\mathcal{B})} \qquad \text{free of } G,\ S$$

- The unknown σ_v appears on top and bottom, so it cancels.
- The bridge $\mathcal{B}$ is independent of the endpoint $\mathcal{W}(1)$.

Cancel the scale: a self-normalized pivot

Recall: bridge $B_T(r) = S_T(r) - r\,S_T(1) \xrightarrow{d} \sigma_v \mathcal{B}(r)$, computable, no $\vartheta^\star$.

Normalizer $\mathcal{D}$: continuous, $\mathcal{D}(cg) = |c|\,\mathcal{D}(g)$, $\mathcal{D}(\mathcal{B}) > 0$.

$$\frac{\sqrt{T}\,(\bar{Y}_T - v^\top \vartheta^\star)}{\mathcal{D}(B_T)} \xrightarrow{d} \frac{\sigma_v\,\mathcal{W}(1)}{\sigma_v\,\mathcal{D}(\mathcal{B})} = \frac{\mathcal{W}(1)}{\mathcal{D}(\mathcal{B})} \qquad \text{free of } G,\ S$$

- The unknown σ_v appears on top and bottom, so it cancels.
- The bridge $\mathcal{B}$ is independent of the endpoint $\mathcal{W}(1)$.

Random scaling (Lee et al., 2022; Li et al., 2025)

$\mathcal{D}_{L_2}(B) = \left(\int_0^1 B^2\right)^{1/2}$: $\quad \bar{Y}_T \pm 6.747\,\mathcal{D}_{L_2}(B_T)/\sqrt{T}$ (95%, 4 running sums).

Another normalizer: project the bridge on polynomials

Denominator = a projection of the bridge. Project B_T on $1, r, \ldots, r^{K-1}$ and whiten:

$$a_j = \int_0^1 B_T(r)\, r^{j-1} dr \xrightarrow{\text{whiten}} \boldsymbol{z} = \Sigma_K^{-1/2} \boldsymbol{a} \approx \sigma_v\,(Z_1, \ldots, Z_K), \quad Z_j \overset{\text{iid}}{\sim} N(0,1),$$

$$\mathcal{D}^2_{P_K} = \tfrac{1}{K}\|\boldsymbol{z}\|^2 = \tfrac{1}{K}\,\boldsymbol{a}^\top \Sigma_K^{-1} \boldsymbol{a}, \qquad (\Sigma_K)_{jk} = \mathrm{Cov}\Big(\textstyle\int \mathcal{B}\, r^{j-1}, \int \mathcal{B}\, r^{k-1}\Big) = \frac{1}{(j+1)(k+1)(j+k+1)}.$$

Another normalizer: project the bridge on polynomials

Denominator = a projection of the bridge. Project B_T on $1, r, \ldots, r^{K-1}$ and whiten:

$$a_j = \int_0^1 B_T(r)\, r^{j-1} dr \xrightarrow{\text{whiten}} \boldsymbol{z} = \Sigma_K^{-1/2}\boldsymbol{a} \approx \sigma_v\,(Z_1, \ldots, Z_K), \quad Z_j \overset{\text{iid}}{\sim} N(0,1),$$

$$\mathcal{D}^2_{P_K} = \tfrac{1}{K}\|\boldsymbol{z}\|^2 = \tfrac{1}{K}\,\boldsymbol{a}^\top \Sigma_K^{-1}\boldsymbol{a}, \qquad (\Sigma_K)_{jk} = \mathrm{Cov}\Big(\int \mathcal{B}\, r^{j-1}, \int \mathcal{B}\, r^{k-1}\Big) = \frac{1}{(j+1)(k+1)(j+k+1)}.$$

Proposition (Li et al., 2023)

$\mathcal{D}^2_{P_K}/\sigma_v^2 \xrightarrow{d} \chi^2_K/K$, independent of $\mathcal{W}(1) \Rightarrow$ pivot t_K: $\quad \bar{Y}_T \pm t_{K,0.975}\,\dfrac{\mathcal{D}_{P_K}(B_T)}{\sqrt{T}}$.

Another normalizer: project the bridge on polynomials

Denominator = a projection of the bridge. Project B_T on $1, r, \ldots, r^{K-1}$ and whiten:

$$a_j = \int_0^1 B_T(r)\, r^{j-1} dr \xrightarrow{\text{whiten}} \boldsymbol{z} = \Sigma_K^{-1/2}\boldsymbol{a} \approx \sigma_v\,(Z_1, \ldots, Z_K), \quad Z_j \overset{\text{iid}}{\sim} N(0,1),$$

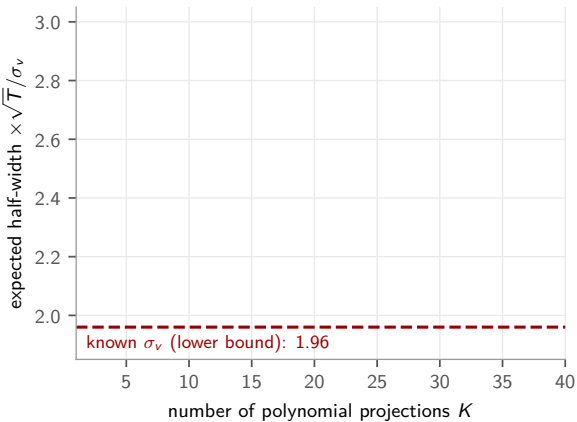
$$\mathcal{D}^2_{P_K} = \tfrac{1}{K}\|\boldsymbol{z}\|^2 = \tfrac{1}{K}\,\boldsymbol{a}^\top \Sigma_K^{-1}\boldsymbol{a}, \qquad (\Sigma_K)_{jk} = \mathrm{Cov}\Big(\int \mathcal{B}\, r^{j-1}, \int \mathcal{B}\, r^{k-1}\Big) = \tfrac{1}{(j+1)(k+1)(j+k+1)}.$$

Proposition (Li et al., 2023)

$\mathcal{D}^2_{P_K}/\sigma_v^2 \xrightarrow{d} \chi^2_K/K$, independent of $\mathcal{W}(1) \Rightarrow$ pivot t_K: $\quad \bar{Y}_T \pm t_{K,0.975}\,\dfrac{\mathcal{D}_{P_K}(B_T)}{\sqrt{T}}$.

- Like a t-test: a sample s.d. from K independent copies of σ_v.
- Online with $2K+1$ running sums.

Not knowing the scale costs length

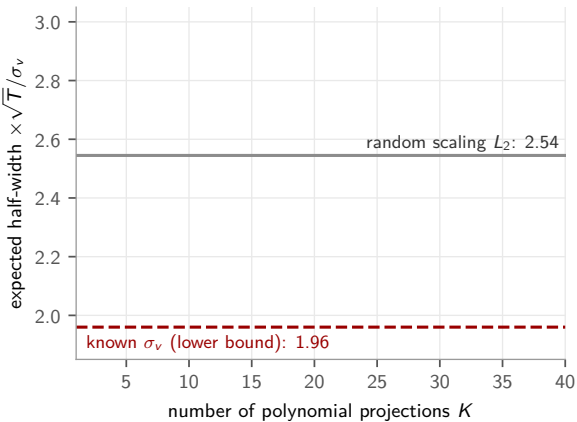


EXPECTED 95% HALF-WIDTH

$c_{\mathcal{D}} \cdot \mathbb{E}\,\mathcal{D}(\mathcal{B})$, in units of $\sigma_v/\sqrt{T}$

- **Known** σ_v: 1.96 (lower bound)

Not knowing the scale costs length

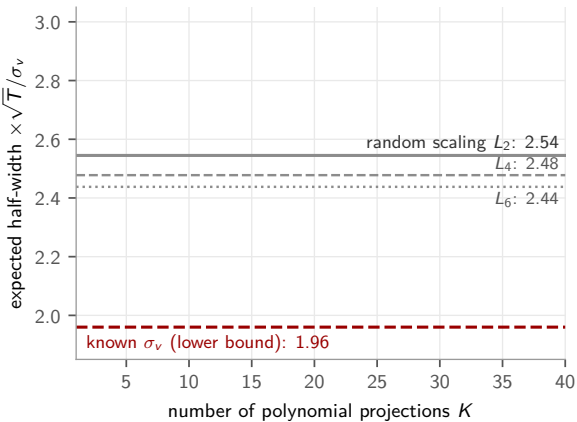


EXPECTED 95% HALF-WIDTH

$c_{\mathcal{D}} \cdot \mathbb{E}\,\mathcal{D}(\mathcal{B})$, in units of $\sigma_v/\sqrt{T}$

- **Known** σ_v: 1.96 (lower bound)
- **Random scaling** L_2: 2.54 (+30%)

Not knowing the scale costs length

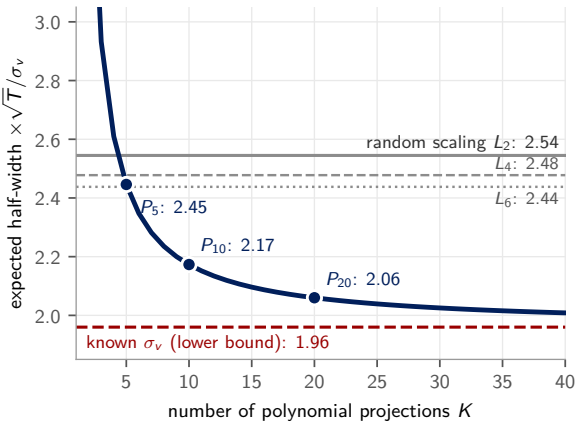


EXPECTED 95% HALF-WIDTH

$c_{\mathcal{D}} \cdot \mathbb{E}\,\mathcal{D}(\mathcal{B})$, in units of $\sigma_v/\sqrt{T}$

- **Known** σ_v: 1.96 (lower bound)
- **Random scaling** L_2: 2.54 (+30%)
- Bridge norms L_4, L_6: 2.48, 2.44

Not knowing the scale costs length



EXPECTED 95% HALF-WIDTH

$c_{\mathcal{D}} \cdot \mathbb{E}\,\mathcal{D}(\mathcal{B})$, in units of $\sigma_v/\sqrt{T}$

- **Known** σ_v: 1.96 (lower bound)
- **Random scaling** L_2: 2.54 (+30%)
- Bridge norms L_4, L_6: 2.48, 2.44
- **Polynomial** P_K: 2.45 (K=5), 2.17 (K=10) $\to$ 1.96

4

Simulation-Based Decisions

Convergence

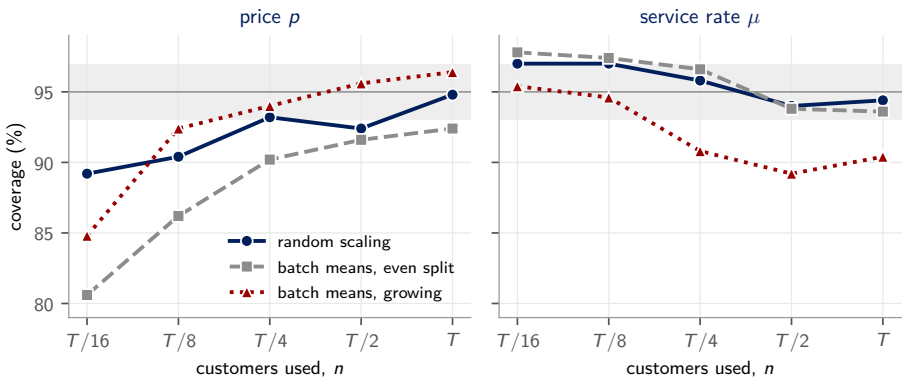
Inference from the Path

Applications

Queues, inventory, and LoRA

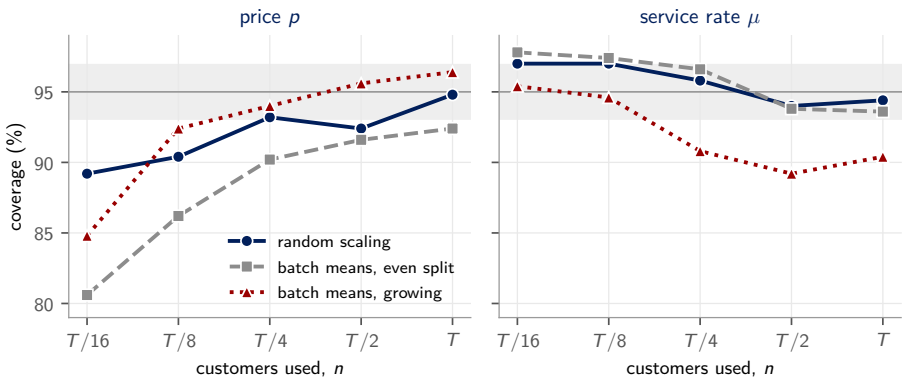
Random scaling in the queue (Li et al., 2025)

M/M/1, $T = 5 \times 10^4$ customers, 500 runs; band: 95%± Monte Carlo error.



Random scaling in the queue (Li et al., 2025)

M/M/1, $T = 5 \times 10^4$ customers, 500 runs; band: 95%± Monte Carlo error.



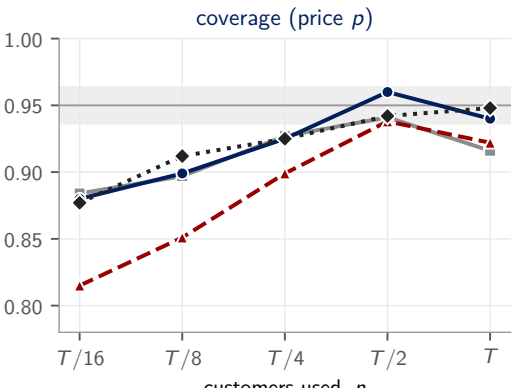
No batch size to tune; close to 95% for both coordinates.

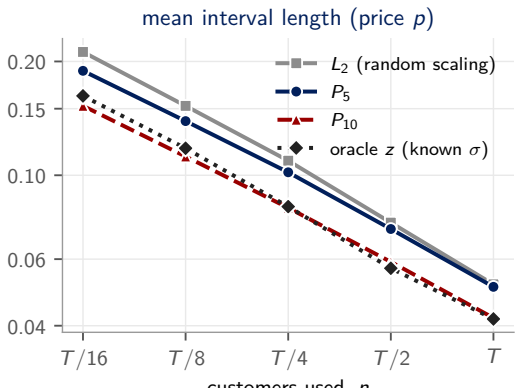
New: the series normalizer in the queue

Corollary. Queue FCLT + $\mathcal{D}_{P_K}$ continuous $\Rightarrow \sqrt{T}\, v^\top(\bar{\vartheta}_T - \vartheta^\star)/\mathcal{D}_{P_K}(B_T) \xrightarrow{d} t_K$.

New: the series normalizer in the queue

Corollary. Queue FCLT + $\mathcal{D}_{P_K}$ continuous $\Rightarrow \sqrt{T}\, v^\top(\bar{\vartheta}_T - \vartheta^\star)/\mathcal{D}_{P_K}(B_T) \xrightarrow{d} t_K$.

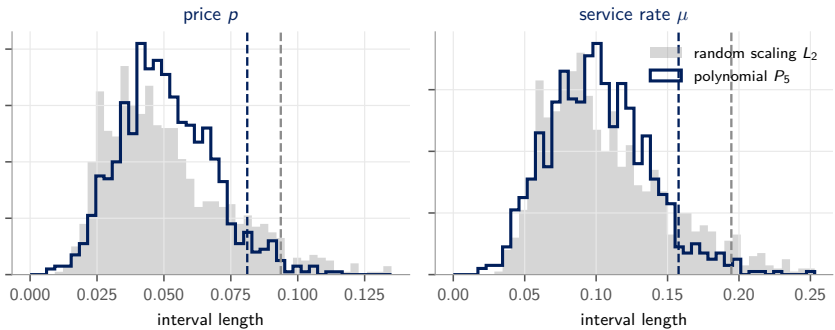




New runs: same M/M/1 model, 1,000 runs; band: 95%± Monte Carlo error. Service rate μ: similar.

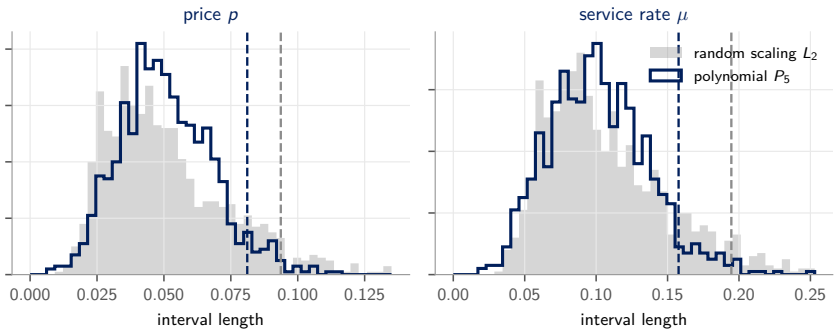
Same coverage, steadier intervals

Both cover about 95%. Final interval length, 1,000 runs (dashed: 95th percentile).



Same coverage, steadier intervals

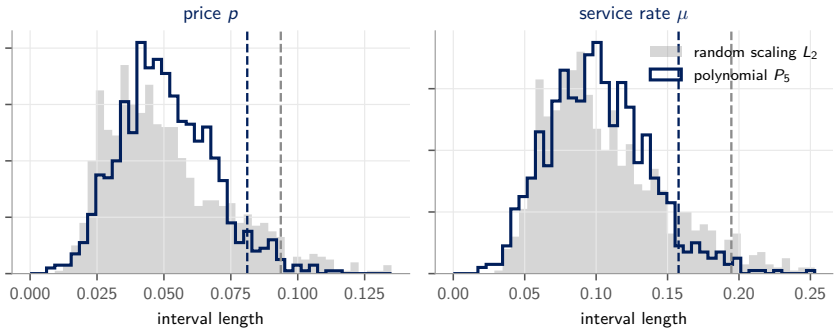
Both cover about 95%. Final interval length, 1,000 runs (dashed: 95th percentile).



Mean length $\approx$ same; **widest intervals** 13–19% **shorter**.

Same coverage, steadier intervals

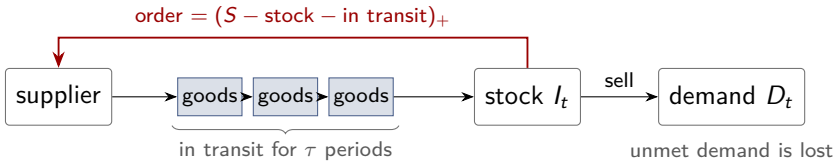
Both cover about 95%. Final interval length, 1,000 runs (dashed: 95th percentile).



Mean length $\approx$ same; **widest intervals** 13–19% **shorter**.

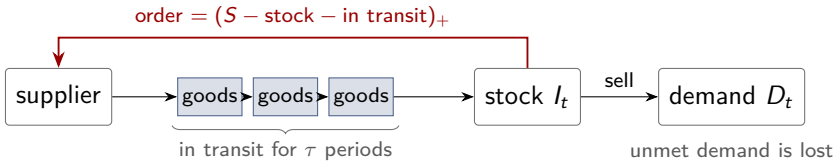
In practice we get **one** run, so fewer very wide intervals matter.

Inventory control with lead times (Li et al., 2025)



- Decision: the **base-stock level** S; cost $h(I - D)_+ + b(D - I)_+$.
- Today's S changes future stock: data depend on the decision.
- Coupling gives the contraction (with $V \equiv 1$).

Inventory control with lead times (Li et al., 2025)



- Decision: the **base-stock level** S; cost $h(I - D)_+ + b(D - I)_+$.
- Today's S changes future stock: data depend on the decision.
- Coupling gives the contraction (with $V \equiv 1$).

Coverage at T (%)	$\tau = 1$	$\tau = 2$
random scaling	**93.6**	**92.6**
batch means	88.0	88.0

$h = 1$, $b = 10$, $D \sim \mathrm{Exp}(1)$; 500 runs.

Exogenous data: LoRA fine-tuning (Li et al., 2023)

Rank-one adapter $\mathbf{W} = \mathbf{BA}$, $\mathbf{B} \in \mathbb{R}^{3\times 1}$, $\mathbf{A} \in \mathbb{R}^{1\times 4}$; covariates $\mathbf{z}_t$ from a sticky Markov chain:

$$\mathbf{y}_t = \mathbf{W}^\star \mathbf{z}_t + \varepsilon_t.$$

- SGD on $(\mathbf{B}, \mathbf{A})$, then rebalance $\|\mathbf{B}\| = \|\mathbf{A}\|$.
- The product $\mathbf{W}_t$ follows a **nonlinear** SA.
- Target: the $(1,1)$ entry $\mathbf{W}^\star_{11}$ of the product, not a factor.

Exogenous data: LoRA fine-tuning (Li et al., 2023)

Rank-one adapter $\mathbf{W} = \mathbf{BA}$, $\mathbf{B} \in \mathbb{R}^{3\times 1}$, $\mathbf{A} \in \mathbb{R}^{1\times 4}$; covariates $\mathbf{z}_t$ from a sticky Markov chain:

$$\mathbf{y}_t = \mathbf{W}^\star \mathbf{z}_t + \varepsilon_t.$$

- SGD on $(\mathbf{B}, \mathbf{A})$, then rebalance $\|\mathbf{B}\| = \|\mathbf{A}\|$.
- The product $\mathbf{W}_t$ follows a **nonlinear** SA.
- Target: the $(1,1)$ entry $\mathbf{W}^\star_{11}$ of the product, not a factor.

Coverage at 10^5 (%)	
P_5	**96.8**
random scaling L_2	96.0
Markov batch means	91.2
online bootstrap	91.2

250 runs. P_5 is 3.6% shorter than L_2; the bootstrap is 2.4× slower.

Summary

OPTIMIZATION

Learn from every customer

- Stream SGD: one sample per update
- $O(1/T)$ error, $O(\log T)$ regret
- Poisson equation; Wasserstein-type divergence checked by coupling

Summary

OPTIMIZATION

Learn from every customer

- Stream SGD: one sample per update
- $O(1/T)$ error, $O(\log T)$ regret
- Poisson equation; Wasserstein-type divergence checked by coupling

INFERENCE

Read the whole path

- FCLT: Brownian, one unknown scale
- Any scale-equivariant $\mathcal{D}$: online CI
- Queues, inventory, LoRA

Summary

OPTIMIZATION

Learn from every customer

- Stream SGD: one sample per update
- $O(1/T)$ error, $O(\log T)$ regret
- Poisson equation; Wasserstein-type divergence checked by coupling

INFERENCE

Read the whole path

- FCLT: Brownian, one unknown scale
- Any scale-equivariant $\mathcal{D}$: online CI
- Queues, inventory, LoRA

Prove the FCLT once; then the inference method is a design choice.

Thank you!

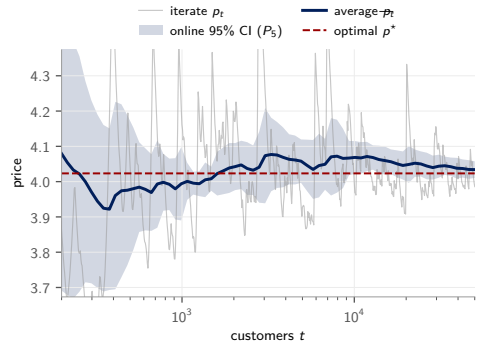
Questions are welcome.

Open: choice of K; finite-T coverage; multi-server queues.

arXiv:2309.09545 · arXiv:2302.07690

github.com/lx10077/StreamSGD

github.com/lx10077/nonlinear-sa-inference

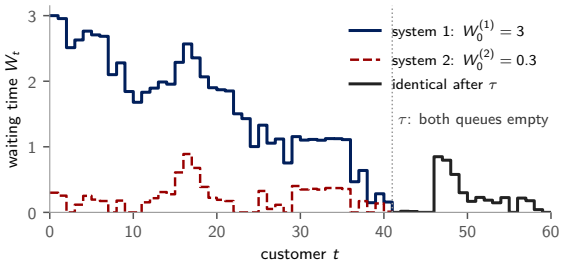


One run of stream SGD: the decision converges, and the same path gives the confidence interval.

Backup Slides

How we use synchronous coupling

Checking contraction in the GI/GI/1 queue.



Two copies of $W_{t+1} = (W_t + V_t/\mu - U_t/\lambda)_+$, same customers (U_t, V_t), different starts.

Gap $\Delta_t = |W_t^{(1)} - W_t^{(2)}|$:

- $\Delta_{t+1} \leq \Delta_t$ ($(\cdot)_+$ is 1-Lipschitz).
- Both empty at $\tau \Rightarrow \Delta_t = 0$, $t \geq \tau$.
- $\mathbb{P}(\tau > t)$ decays exponentially if $\lambda < \mu$.

$\Rightarrow$ contraction, uniformly in ϑ.

The gradient estimate in the queue

With $x = (w, y)$ and service-time correction $w + y + 1/\mu$,

$$H(\vartheta, x) = \begin{pmatrix} c'(\mu) - h_0 \dfrac{\lambda(p)}{\mu}\Big(w + y + \dfrac{1}{\mu}\Big) \\ -\lambda(p) - p\lambda'(p) + h_0\,\lambda'(p)\Big(w + y + \dfrac{1}{\mu}\Big) \end{pmatrix}, \qquad \mathbb{E}_{\pi_\vartheta} H(\vartheta, x) = \nabla f(\vartheta).$$

- Representation of $\nabla \mathbb{E} Q_\infty$ through waiting and busy times (Chen et al., 2023).
- Stream SGD evaluates H at the *current* state, not at steady state; the Poisson equation controls the resulting bias.

Assumptions for decision-dependent data (Li et al., 2025)

1. **Convexity and smoothness:** bounded Θ; $\langle \vartheta - \vartheta^\star, \nabla f(\vartheta) \rangle \geq K_0 \|\vartheta - \vartheta^\star\|^2$; $\|\nabla f(\vartheta)\| \leq K_1 \|\vartheta - \vartheta^\star\|$.
2. **Steps:** decreasing, bounded, $|\eta_t^{-1} - \eta_{t-1}^{-1}| \leq \varkappa$.
3. **Joint continuity:** $\|\mathcal{P}_\vartheta H(\vartheta, x) - \mathcal{P}_{\vartheta'} H(\vartheta', x')\| \leq L(1 + V(x))(\|x - x'\| + \|\vartheta - \vartheta'\|)$.
4. **Kernel continuity:** $\mathcal{W}_{\|\cdot\|, V^2}(\delta_x \mathcal{P}_\vartheta, \delta_x \mathcal{P}_{\vartheta'}) \leq L\|\vartheta - \vartheta'\|(1 + V(x)^2)$.
5. **Contraction:** $\mathcal{W}_{\rho, \tilde{V}}(\delta_x \mathcal{P}_\vartheta^n, \delta_y \mathcal{P}_\vartheta^n) \leq C\rho(x, y) e^{-cn}(2 + \tilde{V}(x)^2 + \tilde{V}(y)^2)$.
6. **For the FCLT:** $\|\nabla f(\vartheta) - G(\vartheta - \vartheta^\star)\| \leq L_G \|\vartheta - \vartheta^\star\|^2$ near $\vartheta^\star$; $\sum \eta_t = \infty$, $\sum \eta_t^{1.5} < \infty$.

Here $\mathcal{W}_{\rho, V}(\mu, \nu) = \inf_\gamma \mathbb{E}_\gamma[\rho(X, Y)\{1 + V(X) + V(Y)\}]$. GI/GI/1: $V(w, y) = y + e^{\kappa w}$. Inventory: $V \equiv 1$.

A menu of normalizers

Normalizer	$\mathcal{D}(B)$	**Limit pivot**	**Online state**
Random scaling, L_2	$\left(\int_0^1 B^2\right)^{1/2}$	tabulated; $c_{.95} = 6.747$	4 scalars
Bridge norm, L_m	$\left(\int_0^1 \lvert B\rvert^m\right)^{1/m}$	simulated once	$m + 2$ scalars
Area method, P_1	$\sqrt{12}\left\lvert\int_0^1 B\right\rvert$	t_1 (Cauchy)	3 scalars
Polynomial series, P_K	$\left(K^{-1}\boldsymbol{a}^\top \Sigma_K^{-1}\boldsymbol{a}\right)^{1/2}$	t_K	$2K + 1$ scalars

Abadir and Paruolo (1997); Lee et al. (2022); Glynn and Iglehart (1990); Sun (2013); Li et al. (2023).

Why not take K large?

- The limit says: larger K is always shorter.
- In finite samples, higher-order projections read **finer-scale** features of the path, where the Brownian approximation is still poor (transient, Markov memory).

Ablation over five exogenous-data applications (Li et al., 2023)

	P_5	P_{10}	P_{20}
Average coverage (nominal 95%)	94.4%	94.2%	92.5%
Mean length relative to L_2	94.2%	82.0%	73.5%

In the queue, P_{10} covers only 81.5% for the price at $n = T/16$.

Exogenous Markov data: five applications (Li et al., 2023)

250 replications, 10^5 observations, nominal 95%.

	Coverage				Length
	P_5	L_2	Markov OBM	Bootstrap	P_5/L_2
RiverSwim, tabular Q	91.2	94.4	84.4	94.4	0.90
Projected linear Q	95.2	94.8	83.2	92.4	0.96
Soft Q, CliffWalking	94.4	92.4	87.2	91.2	0.94
Markov logistic SGD	94.4	93.2	84.8	88.8	0.96
Balanced LoRA	96.8	96.0	91.2	91.2	0.96

Bootstrap ($B = 10$): 2.4–2.8$\times$ longer intervals in Q-learning, 2.4$\times$ runtime, 560 vs. 36 floats.

Balanced LoRA in detail (Li et al., 2023)

Rank-one $\mathbf{W} = \mathbf{BA}$, $\mathbf{B} \in \mathbb{R}^{3\times 1}$, $\mathbf{A} \in \mathbb{R}^{1\times 4}$. Covariate $\mathbf{z}_t$ persists with probability 0.85, else redrawn from $\{\pm 2\mathbf{e}_j\}$; $\mathbf{y}_t = \mathbf{W}^\star \mathbf{z}_t + \varepsilon_t$.

$$\Gamma_t = (\mathbf{W}_t \mathbf{z}_t - \mathbf{y}_t)\mathbf{z}_t^\top, \qquad \tilde{\mathbf{B}} = \mathbf{B}_t - \eta_t \Gamma_t \mathbf{A}_t^\top, \quad \tilde{\mathbf{A}} = \mathbf{A}_t - \eta_t \mathbf{B}_t^\top \Gamma_t,$$

then rebalance so that $\|\mathbf{B}_{t+1}\| = \|\mathbf{A}_{t+1}\|$ (the product is unchanged). The product follows

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \{\Gamma_t (\mathbf{W}_t^\top \mathbf{W}_t)^{1/2} + (\mathbf{W}_t \mathbf{W}_t^\top)^{1/2} \Gamma_t\} + \eta_t^2 \Gamma_t \mathbf{W}_t^\top \Gamma_t,$$

a nonlinear SA through the matrix square roots. Steps $\eta_t = \min\{0.4,\, 0.75(t+1)^{-0.55}\}$.

Online recursions

RANDOM SCALING (Li et al., 2025)

With $O_{t,1} = \sum_{i \le t} i^2 (v^\top \bar{\vartheta}_i)^2$, $O_{t,2} = \sum_{i \le t} i^2 v^\top \bar{\vartheta}_i$, $O_{t,3} = \sum_{i \le t} i^2$:

$$\bar{\sigma}_t^2 = \frac{1}{t^2} \Big\{ O_{t,1} - 2 O_{t,2}\, v^\top \bar{\vartheta}_t + O_{t,3} (v^\top \bar{\vartheta}_t)^2 \Big\}, \qquad v^\top \bar{\vartheta}_t \pm 6.747\, \frac{\bar{\sigma}_t}{\sqrt{t}}.$$

POLYNOMIAL SERIES (Li et al., 2023)

With $m_{j,T} = T^{-(j+1)} \sum_{t \le T} t^j Y_t$ and $p_{j,T} = T^{-(j+1)} \sum_{t \le T} t^j$,

$$m_{j,T} = \Big(\tfrac{T-1}{T} \Big)^{j+1} m_{j,T-1} + \frac{Y_T}{T}, \qquad \tilde{a}_{j,T} = -\frac{\sqrt{T}}{j} \big(m_{j,T} - \bar{Y}_T\, p_{j,T} \big),$$

and $p_{j,T}$ likewise. In total $2K + 1$ scalars. The code uses a shifted-Legendre basis; the interval is unchanged.

Inventory with lead times (Li et al., 2025)

State $X_t = (Q_{t-1}, \ldots, Q_{t-\tau+1}, I_t)$; $I_{t+1} = (I_t - D_t)_+ + Q_{t-\tau+1}$; order $Q_t = (S - X_t\mathbf{1})_+$. Cost $h(I-D)_+ + b(D-I)_+$ with $h = 1$, $b = 10$, $D \sim \mathrm{Exp}(1)$; sample-path derivative of I_t in S gives H; $\eta_t = 2t^{-0.67}$.

Coverage (%)		$T/16$	$T/8$	$T/4$	$T/2$	T
$\tau = 1$	random scaling	84.4	89.2	86.6	89.2	93.6
	batch means, even split	74.4	81.6	83.2	83.8	88.0
	batch means, growing	96.0	92.6	86.4	90.0	91.6
$\tau = 2$	random scaling	87.2	90.2	91.8	91.8	92.6
	batch means, even split	78.2	83.6	88.0	88.8	88.0
	batch means, growing	93.0	89.0	86.2	81.8	84.4

500 repetitions. $S^\star$ is itself a simulation estimate, so we did not rerun P_K on this model.

References I

K. M. Abadir and P. Paruolo. Two mixed normal densities from cointegration analysis. *Econometrica*, 65(3): 671–680, 1997.

C. Andrieu, É. Moulines, and P. Priouret. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on Control and Optimization*, 44(1):283–312, 2005.

E. Che, J. Dong, and X. T. Tong. Stochastic gradient descent with adaptive data. *Operations Research*, 2024. doi:10.1287/opre.2024.1014.

X. Chen, Y. Liu, and G. Hong. An online learning approach to dynamic pricing and capacity sizing in service systems. *Operations Research*, 2023.

P. W. Glynn and D. L. Iglehart. Simulation output analysis using standardized time series. *Mathematics of Operations Research*, 15(1):1–16, 1990.

W. T. Huh, G. Janakiraman, J. A. Muckstadt, and P. Rusmevichientong. An adaptive algorithm for finding the optimal base-stock policy in lost sales inventory systems with censored demand. *Mathematics of Operations Research*, 34(2):397–416, 2009.

S. Lee, Y. Liao, M. H. Seo, and Y. Shin. Fast and robust online inference with stochastic gradient descent via random scaling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

X. Li, J. Liang, and Z. Zhang. Online statistical inference for nonlinear stochastic approximation with Markovian data. *arXiv preprint arXiv:2302.07690*, 2023.

References II

X. Li, J. Liang, X. Chen, and Z. Zhang. Convergence and inference of stream stochastic gradient descent, with applications to queueing systems and inventory control. *Operations Research*, 2025. doi:10.1287/opre.2025.1662; arXiv:2309.09545.

Y. Sun. A heteroskedasticity and autocorrelation robust F test using an orthonormal series variance estimator. *The Econometrics Journal*, 16(1):1–26, 2013.