Probabilistic Causal Off-Policy Evaluation for LQG Control

Kasra Fallah, PhD Student, Columbia University

Kasra.fallah@columbia.edu

Probabilistic Models and Machine Learning Course Project

COLUMBIA UNIVERSITY

Motivation

Off-policy evaluation (OPE) asks how a new *evaluation* policy π_e would perform using data collected under a different *behavior* policy π_b. In feedback-controlled systems, directly deploying π_e without guarantees can be dangerous, so OPE provides a **causal** counterfactual estimate of long-run performance. This is critical for **safety-critical** domains (robotics, medical devices) where testing a new controller online risks instability or failure. The partially observed linear–Gaussian setting (LQG) introduces additional challenges: the system state x_t is latent and data are from a closed-loop probabilistic model. Thus, OPE in this context becomes a problem of **probabilistic modeling and inference** (to reconstruct latent trajectories) combined with counterfactual weighting.

Model

We consider a discrete-time **linear–Gaussian state-space model** representing the dynamical system:

$$x_{t+1} = A\,x_t + B\,u_t + w_t, \qquad w_t \sim \mathcal{N}(0,\, W) \tag{1}$$

$$y_t = C\,x_t + v_t, \qquad v_t \sim \mathcal{N}(0,\, V)\;. \tag{2}$$

Here $x_t \in \mathbb{R}^n$ is the latent state and $y_t \in \mathbb{R}^p$ is the observed output at time t. Both the behavior and evaluation policies are **output-feedback controllers** with Gaussian exploration:

$$u_t \mid y_t \sim \mathcal{N}(K\,y_t,\, \Sigma_u), \qquad \pi_b : K = K_b,\;\; \pi_e : K = K_e\;. \tag{3}$$

The system is operated under π_b to collect a dataset $D = \{(y_t,\, u_t,\, \ell_t)\}_{t=0}^{H-1}$ of length H. We define the long-run average cost under π_e as

$$J(\pi_e) \;=\; \mathbb{E}_{\pi_e}\!\left[\frac{1}{H}\sum_{t=0}^{H-1} \ell(x_t,\, u_t)\right], \tag{4}$$

where a standard quadratic loss $\ell(x,u) = x^\top Q\,x + u^\top R\,u$ is used. Our goal is to estimate $J(\pi_e)$ off-policy using the logged data from π_b.

Inference

Since x_t is latent, we perform **probabilistic system identification** via EM: the E-step runs a Kalman smoother to infer posterior trajectories, and the M-step updates (A, B, C, W, V) via expected complete-data likelihood. This yields a fitted model $\hat{M} = (\hat{A}, \hat{B}, \hat{C}, \hat{W}, \hat{V})$ used for downstream simulation.

Estimators

We compare three OPE estimators for $J(\pi_e)$:

- **Model-Based (MB):** Use the learned model $\hat{M}$ as a simulator. We generate trajectories under π_e in $\hat{M}$ and estimate $J(\pi_e)$ by the sample average of costs: $\hat{J}_{\mathrm{MB}}(\pi_e) = \frac{1}{H}\sum_{t=0}^{H-1} \hat{\ell}_t^{\pi_e}$. MB OPE has low variance but can be **biased** if $\hat{M}$ is misspecified.
- **Self-Normalized PDIS (SN-PDIS):** A purely data-driven approach. We reweight logged costs by the importance ratio $\rho_t = \dfrac{\pi_e(u_t \mid y_t)}{\pi_b(u_t \mid y_t)}$ and $w_t = \prod_{s=0}^{t} \rho_s$:

$$\hat{J}_{\mathrm{SN}}(\pi_e) \;=\; \frac{\sum_{t=0}^{H-1} w_t\,\ell_t}{\sum_{t=0}^{H-1} w_t}\,. \tag{7}$$

  SN-PDIS is unbiased in the limit but variance can be enormous when w_t are variable.
- **Doubly Robust (DR):** A hybrid estimator combining model prediction with importance weighting for residuals (Dudík et al., 2011; Jiang & Li, 2016).

$$\hat{J}_{\mathrm{DR}}(\pi_e) \;=\; \hat{J}_{\mathrm{MB}}(\pi_e) \;+\; \frac{1}{H}\sum_{t=0}^{H-1} \tilde{w}_t \left(\ell_t \;-\; \hat{\ell}_t^{\pi_e}\right). \tag{8}$$

  DR uses the model as a baseline and relies on weights only for the *difference*. It is **doubly robust**: unbiased if either the model or the importance weights are correct.

Theory

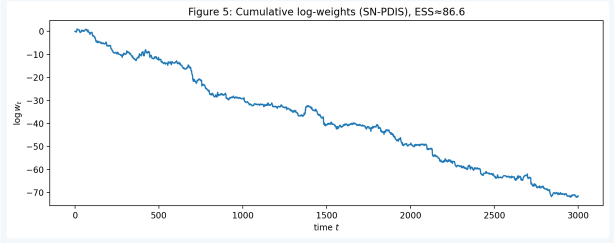
In OPE for partially observed systems, importance weights can become unstable. Let the per-step likelihood ratio be $\rho_t = \frac{\pi_e(u_t|y_t)}{\pi_b(u_t|y_t)}$, $\quad \omega_t = \prod_{s=0}^{t} \rho_s$. As the controller shift $\delta = \|K_e - K_b\|$ grows, $\log \rho_t$ accumulates with non-zero drift, and ω_t exhibits exponential growth or decay. This leads to **weight degeneracy**, where only a few trajectories dominate. The **effective sample size** (ESS) drops sharply:

$$\mathrm{ESS} = \frac{\left(\sum_t \omega_t\right)^2}{\sum_t \omega_t^2}\,. \tag{6}$$

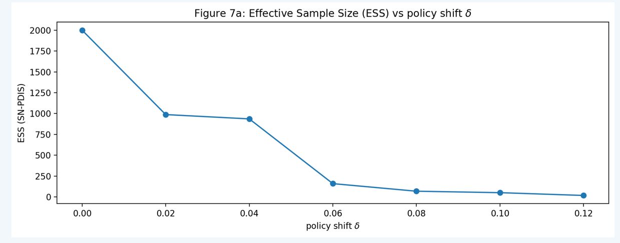
In the low-ESS regime, SN-PDIS exhibits high variance and bias. The **DR estimator** combines model predictions with weighted residuals, achieving low error if either model or weights are accurate.

Results

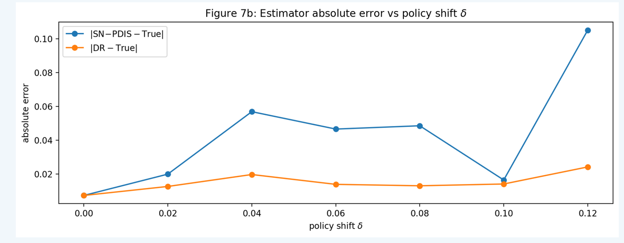
(i) Weight Degeneracy: $\log w_t$ drifts over time, implying exponential growth in raw weights. Figure B shows ESS *plummeting* as δ grows. **(ii) Accuracy:** SN-PDIS estimates degrade rapidly beyond small δ. DR remains accurate over a much wider range (Figure C). **(iii) Mechanism:** Contribution mass $w_t \ell_t$ concentrates on a few "spikes" (Figure D). **(iv) Robustness:** Over many runs, SN-PDIS has an order of magnitude higher variance than DR (Figure F).



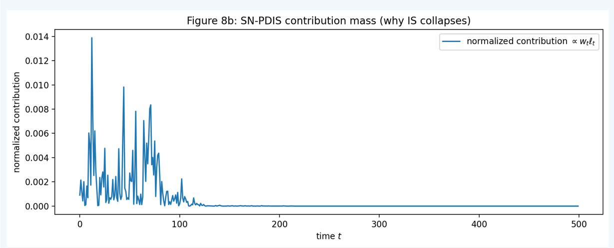
(A) Log-weight growth over time.



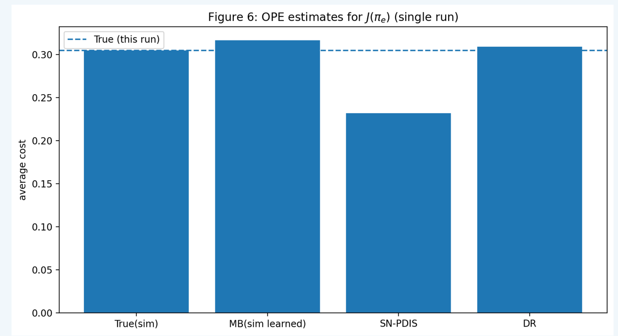
(B) ESS decays with policy shift.



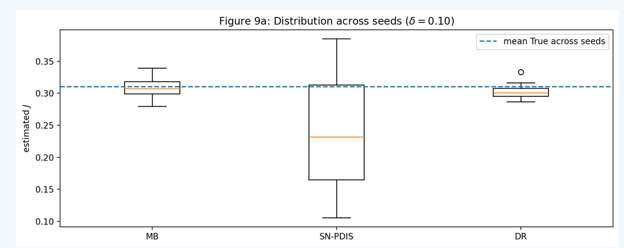
(C) OPE error: DR vs SN-PDIS vs MB.



(D) Mass concentration at few spikes.



(E) Single-run tracking performance.



(F) Variance across runs: DR is stable.

Significance

- **Probabilistic Modeling Innovation:** We formulate OPE under partial observability as latent-variable inference in a structural causal model, using Bayesian EM to estimate hidden dynamics from noisy trajectories.
- **Causal OPE Framework:** OPE is cast as evaluating the counterfactual intervention $\mathrm{do}(\pi_e)$ in a dynamic causal model. This framing clarifies identifiability assumptions and bridges reinforcement learning with modern causal inference.
- **Double Robustness Advantage:** Our estimator unifies model-based prediction and importance weighting into a control variate scheme that remains consistent if either component is accurate, achieving robustness beyond either method alone.
- **Safe RL Applications:** Reliable OPE is vital in safety-critical domains where online testing is risky. By addressing challenges like partial observability, model misspecification, and weight degeneracy, our method supports pre-deployment validation of new controllers using offline data.