Markovian Experimental Design under Concept Drift

Ayberk Yarkın Yıldız[1] **Lili Su**[1] **Carlee Joe-Wong**[2] **Edmund Yeh**[1] **Stratis Ioannidis**[1]
[1]Northeastern University, Boston, MA [2]Carnegie Mellon University, Pittsburgh, PA
{yildiz.ay,l.su,e.yeh,ioannidis}@northeastern.edu
cjoewong@andrew.cmu.edu

Abstract

We study how to optimally select training samples in the presence of concept drift. We propose a *Markovian experimental design* framework, in which a learner sequentially collects samples and re-estimates a model through Bayesian linear regression. We show that, when model drift is mean-reverting, i.e., governed by an unknown mean fixed-point coupled with Gaussian deviations, model posterior estimates are determined by a Kalman filter and the system is asymptotically stable. We also show that the steady-state model posterior can be computed as the solution of the so-called Discrete Algebraic Riccati Equation (DARE), and that D-optimal steady-state and myopic sample selection policies can be computed by solving tractable convex optimization problems. We do so by characterizing the curvature of the DARE solution, even though the latter generally has no closed form. Finally, we show that our approach generalizes to the non-linear model setting by replacing the Kalman filter with an extended Kalman filter.

1 Introduction

In many practical machine learning (ML) settings, algorithms trained over offline data and deployed in realistic conditions are subject to *concept drift* [45, 5]: the model, i.e., the relationship between output and input variables, changes as time progresses. Applications in which concept drift is observed, as well as causes for such drifts, are numerous: in recommender systems, concept drift arises due to changes in content popularity and temporal trends [11, 44], in wireless communications, due to slowly varying channel conditions [1, 57, 60], in environmental monitoring, due to seasonal and meteorological cycles [14], in financial forecasting, due to shifting market and volatility regimes [51], and so forth. In practice, concept drift is often addressed by continuously retraining models [33, 10]. The sheer volume of data streams available necessitates retraining over a subset of all data available at a given time. Privacy regulations and consequent data retention practices [15, 67] dictate that data is erased after some time, restricting retraining only to recently collected data. This gives rise to the following question: which subset of the data should one use to improve the quality of a predictive model, in light of (a) the presence of concept drift and (b) the data available at retraining time?

We propose a *Markovian experimental design* framework that allows us to study this question. In our proposed framework, a learner keeps track of a model drifting according to linear dynamics coupled with Gaussian noise. The learner sequentially selects samples to collect and subsequently re-estimates the model, e.g., via maximum a posteriori estimation over the new data. Our proposed mathematical framework allows us to address the data selection question in a precise and rigorous fashion; in particular, we make the following contributions:

- We model the drift as a specific, mean-reverting Gaussian process: in our drift dynamics, model evolution is characterized by a mean fixed-point coupled with Gaussian deviations. Concretely, the drifting model $\boldsymbol{\beta}_k \in \mathbb{R}^d$ evolves as

$$\boldsymbol{\beta}_{k+1} = \rho\, \boldsymbol{\beta}_k + (1-\rho)\, \boldsymbol{\beta}_* + \boldsymbol{w}_k, \qquad \text{where } \rho \in (0,1), \quad \boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{w}}^2 \mathbf{I}) \tag{1}$$

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

i.e., oscillating around the (unknown) mean fixed-point $\boldsymbol{\beta}_*$. These dynamics are simple but also relevant from a practical standpoint: in a real-life scenario, the mean fixed-point captures "on-average" or "status-quo" behavior, as it corresponds to the model learned if one ignored the presence of drift. In turn, Gaussian deviations correspond to temporal concept drifts that need to be accounted for.

- We first consider a Bayesian linear regression setting. We show that, under these model dynamics, (a) model posterior estimates are governed by a Kalman filter, (b) the mean fixed-point can be estimated arbitrarily well, and (c) the system is asymptotically stable under stationary selection policies. In particular, the system converges to a steady-state model posterior, whose covariance can be computed as the solution of the so-called Discrete Algebraic Riccati Equation (DARE) [4]. We also prove that D-optimal sample selection policies, both myopic/one-step lookahead as well as in steady state, can be computed efficiently by solving appropriately defined convex optimization problems.
- Finally, we extend our analysis to the non-linear case through the use of an extended Kalman filter (EKF) [8], which linearizes the model function.

From a technical standpoint, even though the DARE for arbitrary drift dynamics does not generally have a closed-form solution, we construct an explicit characterization of the spectrum of its solution under our dynamics. We accomplish this by coupling the eigen-decomposition of the covariance limit under our proposed mean fixed-point dynamics with the eigen-decomposition of the design matrix. This paves the way to optimizing the stationary policy in steady state: armed with this characterization, we prove that the D-optimal design objective is Schur-concave w.r.t. this spectrum, which in turn implies concavity over the positive semi-definite cone. Our mean fixed-point dynamics are also crucial to computing a steady-state policy in the EKF setting, as they enable us to compute the model Jacobian around the mean fixed-point limit.

2 Related Work

Experimental design [12, 9, 55, 28], i.e., the process of determining which samples to collect/experiments to conduct prior to model inference, has a long history in both machine learning and statistical inference. In the context of linear regression, the so-called "alphabet" criteria [55, 12] aim at minimizing different scalarizations of the model covariance; Bayesian versions of the latter also exist [12], and the D-optimality criterion we consider here falls under this category. The interest in sequential experimental design procedures, in which experiments are conducted sequentially, can be traced back to seminal works by Wald [68] and Robbins [58]. Bayesian assumptions naturally fit this setting, and many classic sequential experimental design works follow such an approach [12, 18, 48, 3, 17]. Significant computational challenges arise when estimating high-dimensional, non-linear models [23, 29, 30]. Experimental design has also been used in the estimation of model parameters of a dynamical system; see, e.g., [64, 65, 46]. Studied problems and approaches are orthogonal to the ones we consider here, namely, selecting which samples to collect; conversely, they would be useful for estimating parameters governing drift dynamics in our setting (e.g., matrix $\boldsymbol{A}$ in App. E). Nevertheless, both this and the classic regime assume that model parameters to be estimated are fixed, and not subject to concept drift across stages; as a result, our consideration of drift constitutes a significant departure from prior experimental design literature.

The prevalence of concept drifts in real-life settings has given rise to online learning algorithms designed to adapt a model in the presence of drift [45, 71]. This is also a rich research area that, not surprisingly, again traces its roots to Robbins & Monro [59] and includes classic stochastic approximation algorithms [7, 39], window-based and forgetting-factor approaches [72, 36, 43], and ensemble methods [35, 70]. Beyond model adaptation methods, techniques combining drift detection [5, 16] with online learning have been proposed [69, 42, 38, 6]. These methods primarily focus on adapting model parameters to track evolving data distributions. Though studying a rather simple setting (primarily linear regression) compared to the literature on concept drift, our work departs by focusing on the question of how to combine online learning (i.e., model adaptation) with experimental design (i.e., the choice of experiments to conduct/samples to collect).

A more closely related thread considers experimental design for Hidden Markov Models (HMMs), where the model parameters (the hidden state) take one out of finitely many possible values, and transitions are Markovian. Several experimental design techniques for HMMs use objectives akin to

ours (e.g., expected information gain [54] or entropy reduction in latent variable models [31]) and study myopic, one-step-ahead policies [54, 31, 56, 66]. Alali & Imani [2] study both steady-state and infinite-horizon policies using deep RL, to approximate a Bayesian lookahead policy for data collection in HMMs. Our approach differs from this line of work by departing from the finite hidden state assumption, considering a dynamical system evolution of drift instead. This leads to an infinite state space, but is also more realistic when model changes are smooth.

Another related thread studies sensor scheduling and selection for Kalman filtering, where one selects which sensors to activate so as to minimize the steady-state estimation error. Dutta et al. [19] give a unified full-horizon formulation solved by mixed-integer programming, without optimality guarantees; Ye et al. [73] study steady-state sensor selection under the trace (A-optimality) criterion, prove it is NP-hard and inapproximable, and resort to greedy heuristics. Our setting differs in two respects. First, our "state" is a drifting learned model with an unknown fixed point $\boldsymbol{\beta}_*$ estimated online, rather than a known-dynamics physical state. Second, we optimize a continuous, fractional design over the simplex rather than a combinatorial subset of sensors; together with our $\rho\mathbf{I}$ plus isotropic-noise structure, this yields a closed-form characterization of the DARE spectrum and a concave steady-state D-optimal objective. As a result, our design is a poly-time solvable convex program, circumventing the above hardness results. We include the greedy Kalman information-gain heuristic [61] as a competitor in our experiments.

To the best of our knowledge, the work closest to ours is a recent paper by Si Salem [62], which considers which samples to retain in an online learning setting. The value of data is captured by a transductive loss which relates to the expected prediction error (Eq. (33)) in our paper. Si Salem [62] provides a data retention algorithm with bounded regret guarantees in this setting. We depart by modeling drift in a Markovian fashion; this implies that estimation via a Kalman filter is optimal and, therefore, no data needs to be retained past the current stage. This also allows us to identify optimal (as opposed to bounded-regret) myopic and steady-state decisions in our setting.

3 Markovian Experimental Design under Concept Drift

3.1 Problem Setting

We consider a dynamic learning setting in which a learner tries to learn a model subject to *concept drift*. The learning process, illustrated in Fig. 1, proceeds in stages and, in each stage, the learner chooses a set of experiments to conduct and observes labels governed by the model active at this stage. The learner updates its model estimates using these collected labels and proceeds to the next stage, whereupon the model drifts.

Dynamic Label Collection. Given a candidate set of feature vectors $\boldsymbol{\mathcal{X}}_k \subset \mathbb{R}^d$ with $|\boldsymbol{\mathcal{X}}_k| = p_k \in \mathbb{N}$, the learner selects experiments to conduct at stage k by creating a vector $\boldsymbol{n}_k = [n_{ik}]_{i=1}^{p_k} \in \mathbb{N}^{p_k}$ with $\boldsymbol{n}_k^\top \mathbf{1} = n_k$. Under selection vector $\boldsymbol{n}_k$, the experiments to be conducted correspond to the design matrix $\boldsymbol{X}_k \equiv \boldsymbol{X}(\boldsymbol{n}_k) \in \mathbb{R}^{n_k \times d}$, where the map $\boldsymbol{X} : \mathbb{N}^{p_k} \to \mathbb{R}^{n_k \times d}$ stacks n_{ik} copies of vector $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}}_k$, for $i = 1, \ldots, p_k$. The learner collects:

$$\boldsymbol{y}_k = h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) + \boldsymbol{\epsilon}_k, \quad \text{where } \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{2}$$

and $h : \mathbb{R}^{n_k \times d} \times \mathbb{R}^{d_{\boldsymbol{\beta}}} \to \mathbb{R}^{n_k}$ is a function parameterized by $\boldsymbol{\beta}_k \in \mathbb{R}^{d_{\boldsymbol{\beta}}}$ (a.k.a. the *model*) at stage k. Our goal is to determine how the learner should (a) choose its *design selection policy* $\boldsymbol{n}_k \in \mathbb{N}^{p_k}$ and (b) *estimate* $\boldsymbol{\beta}_k \in \mathbb{R}^{d_{\boldsymbol{\beta}}}$ at each stage k, given that the model is subject to drift. We discuss drift next.

Concept Drift. A simple way to model drift is via: $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \boldsymbol{w}_k$, where $\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{w}}^2\mathbf{I})$, with the first model sampled from a known prior, i.e., $\boldsymbol{\beta}_1 \sim \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. Model parameters are thus subject to Gaussian perturbations of variance $\sigma_{\boldsymbol{w}}^2$, capturing temporally-localized trends. A problem with this simple model is that model variance drifts to infinity: this is not realistic in standard training tasks, where both labels and features stay bounded. We thus assume the following dynamics:

$$\boldsymbol{\beta}_{k+1} = \rho\boldsymbol{\beta}_k + (1-\rho)\,\boldsymbol{\beta}_* + \boldsymbol{w}_k, \tag{3}$$

where $\rho \in (0, 1)$ is a positive constant, $\boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{w}}^2\mathbf{I})$ are i.i.d. perturbations, and $\boldsymbol{\beta}_* \in \mathbb{R}^{d_{\boldsymbol{\beta}}}$ is an (unknown) mean fixed-point.[1] Intuitively, the dynamics in Eq. (3) capture a model that oscillates

[1] It is easy to see that $\lim_{k\to\infty} \mathbb{E}[\boldsymbol{\beta}_k] = \boldsymbol{\beta}_*$ under Eq. (3). Note that, in contrast, $\lim_{k\to\infty} \boldsymbol{\beta}_k$ *does not* converge in the classic sense: it does, however, converge *in distribution* to a Gaussian r.v. centered at $\boldsymbol{\beta}_*$.

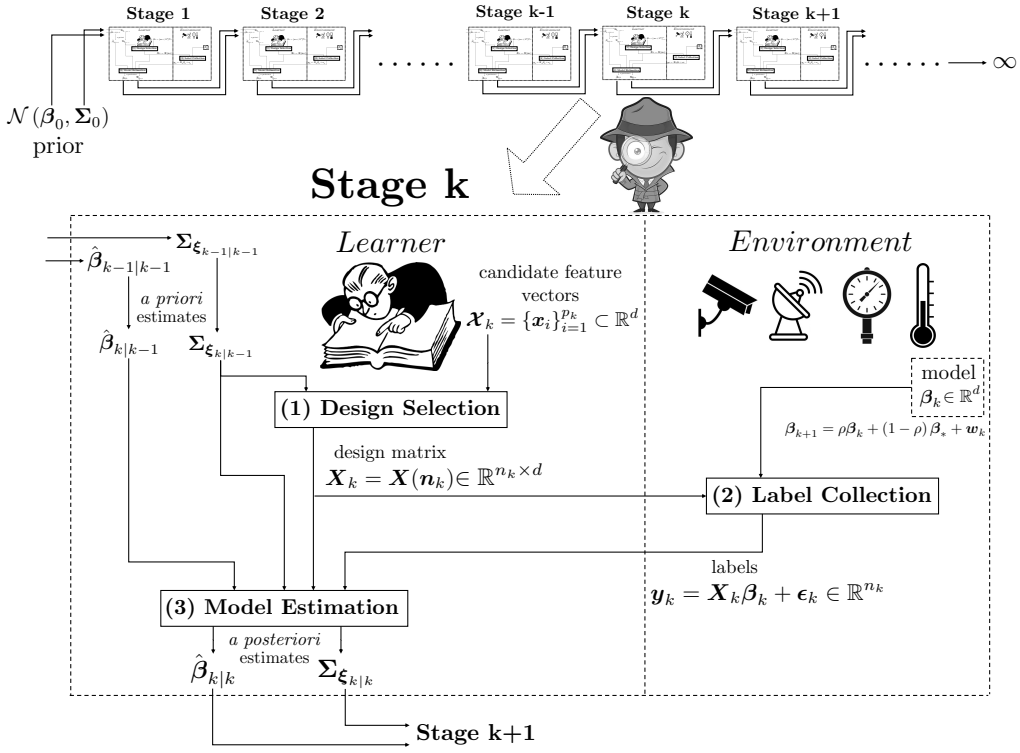


Figure 1: Markovian experimental design setting. At each stage k, the learner (1) selects a design $\boldsymbol{X}_k = \boldsymbol{X}(\boldsymbol{n}_k)$ from the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$, (2) collects labels $\boldsymbol{y}_k$ generated by the current model $\boldsymbol{\beta}_k$, and (3) updates its model estimates. Between stages, the model drifts via Eq. (3), so a Kalman filter (Thm. 1) propagates the *a posteriori* estimates of stage $k-1$ to the *a priori* estimates of stage k. We study how to perform step (1) optimally, both myopically and in steady state (Sec. 3.3).

around $\boldsymbol{\beta}_*$. The latter corresponds to mean behavior, or the "status-quo": it is the model the learner discovers when ignoring temporal drifts altogether (see also Thm. 2 below).

Linear Setting. Our main focus is the linear setting, where $d_{\boldsymbol{\beta}} = d$, $h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) \equiv \boldsymbol{X}_k\boldsymbol{\beta}_k$, and:

$$\boldsymbol{y}_k = \boldsymbol{X}_k\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k, \quad \text{where } \boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\mathbf{I}). \tag{4}$$

We show that, in the linear setting, Maximum a Posteriori (MAP) model estimation corresponds to a Kalman filter, and subsequently select designs using a classic experimental design criterion. For completeness, we review Kalman filters in App. A and experimental design in App. B.

3.2 Model Estimation in the Linear Setting

Before we optimize design selection, we first discuss how to estimate $\boldsymbol{\beta}_k$ under arbitrary design selection policies. We will subsequently reason about the quality of such estimates to build optimal design selection policies. As a warm-up (but also instructive) step, we momentarily consider estimation when the mean fixed-point $\boldsymbol{\beta}_*$ in Eq. (3) is known *a priori*.

Known $\boldsymbol{\beta}_*$. We use the common Kalman filter notation $(\cdot)_{k|k'}$ to denote a quantity estimated at stage k, having access to samples up to stage $k' \leq k$. Then, the Bayesian estimate of $\boldsymbol{\beta}_k$ is as follows:

Theorem 1. *Consider the system evolving via Eqs.* (3)–(4)*. Assume that the mean fixed-point $\boldsymbol{\beta}_*$ is known, and that design matrices $\boldsymbol{X}_k \in \mathbb{R}^{n_k \times d}$ are selected by an arbitrary (possibly non-stationary) policy. Then, at stage k, the posterior distribution of $\boldsymbol{\beta}_k$ after sample collection is Gaussian with mean and covariance given by:*

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_{k|k-1}\right), \quad \textit{and} \quad \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\mathbf{I} - \boldsymbol{K}_k\boldsymbol{X}_k\right)\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}, \tag{5}$$

respectively, where $\hat{\boldsymbol{\beta}}_{k|k-1}$, $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}$ *are the a priori estimates before label collection:*

$$\hat{\boldsymbol{\beta}}_{k|k-1} = \rho\hat{\boldsymbol{\beta}}_{k-1|k-1} + (1-\rho)\,\boldsymbol{\beta}_*, \qquad \textit{and} \qquad \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} = \rho^2\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k-1|k-1}} + \sigma_{\boldsymbol{w}}^2\mathbf{I}, \tag{6}$$

and $\boldsymbol{K}_k$ *is the so-called* Kalman gain*:*

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{X}_k^\top\left(\boldsymbol{X}_k\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{X}_k^\top + \sigma^2\mathbf{I}\right)^{-1}. \tag{7}$$

We prove this in App. C. Note that the presence of the $\boldsymbol{\beta}_*$ term in Eq. (3) implies that our setting slightly deviates from the classic Kalman filter setting (see App. A). We nevertheless reduce estimation to the classic setting through an appropriate transformation (namely, by subtracting $\boldsymbol{\beta}_*$).

Estimating $\boldsymbol{\beta}_*$. Next, we turn our attention to the case where $\boldsymbol{\beta}_*$ is not known. We show that $\boldsymbol{\beta}_*$ can be estimated arbitrarily well by averaging per-stage linear regression estimates:

Theorem 2. *Consider the system evolving via Eqs.* (3)–(4)*. Assume that the eigenvalues of matrices* $\{\boldsymbol{X}_k^\top\boldsymbol{X}_k\}_{k=1}^\infty$ *are bounded from above as well as bounded away from zero. Then:*

$$\hat{\boldsymbol{\beta}}_* = \frac{1}{k}\sum_{k'=1}^{k}\hat{\boldsymbol{\beta}}_{k'}^{\texttt{LR}}, \textit{ for } \hat{\boldsymbol{\beta}}_{k'}^{\texttt{LR}} \equiv \left(\boldsymbol{X}_{k'}^\top\boldsymbol{X}_{k'}\right)^{-1}\boldsymbol{X}_{k'}^\top\boldsymbol{y}_{k'} \tag{8}$$

converges both w.p. 1 and in expectation to the mean fixed-point $\boldsymbol{\beta}_*$ *as* $k \to \infty$*.*

The proof is in App. D. Combining Thms. 1 and 2, we can estimate the model at stage k using Eqs. (5)–(8), replacing $\boldsymbol{\beta}_*$ with $\hat{\boldsymbol{\beta}}_*$. Note that, in Thm. 1, $\boldsymbol{\beta}_*$ is used to obtain the posterior mean/MAP estimate $\hat{\boldsymbol{\beta}}_{k|k}$ *but not the posterior covariance* $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$. Hence, the resulting plug-in estimate $\hat{\boldsymbol{\beta}}_{k|k}$ is asymptotically unbiased and the covariance is correctly estimated for an arbitrary design selection policy, so long as design matrices are well-conditioned.[2] This also implies that *we can design experiments that minimize the model covariance without exact knowledge of* $\boldsymbol{\beta}_*$. We exploit this in identifying optimal myopic and steady-state design selection policies next.

3.3 Optimal Design Selection in the Linear Setting

We next turn our attention to the question of how the learner is to perform the experimental design selection (i.e., how the learner picks $\boldsymbol{n}_k$ at every stage). To that end, we use the so-called *D-optimality criterion* from classic experimental design (see App. B). We describe how to design D-optimal design selection policies in both the *myopic* and *steady-state* case.

3.3.1 Optimal Myopic Design Selection

We first focus on a *myopic* policy: the learner optimizes the design selection at stage k in a *stage-wise* (i.e., one stage at a time) fashion, having access to the past history at stage $k-1$. In D-optimal design [55, 9], the learner would select experiments that minimize the $\log\det$ of the model covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$. Intuitively, this minimizes the entropy of the (normal) distribution of $\boldsymbol{\beta}_k$; it acts as a proxy for the expected prediction error over an unseen sample (see App. B). As in classic experimental design [55, 9], we assume that n_k is large enough so that we can describe the selection through a fractional policy: our first result is that a D-optimal fractional stage-wise design policy can be obtained by solving a convex optimization problem.

Theorem 3. *Consider the system evolving via Eqs.* (3)–(4)*. The myopic D-optimal fractional policy at stage* k *can be computed as a solution to the following relaxed optimization problem:*

$$\textit{Maximize:} \quad G\left(\boldsymbol{\pi}_k; \sigma^2, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\right) \equiv \log\det\left[\tfrac{n_k}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}_k) + \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}^{-1}\right], \tag{9a}$$

$$\textit{s.t.:} \quad \boldsymbol{\pi}_k^\top\mathbf{1} \le 1, \boldsymbol{\pi}_k \ge \mathbf{0}, \tag{9b}$$

where $\boldsymbol{\pi}_k \in [0,1]^{p_k}$ *is the fractional policy at stage* $k \in \mathbb{N}$*,* $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^{p_k}\pi_i\boldsymbol{x}_i\boldsymbol{x}_i^\top$*, and* $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}^{-1} \in \mathbb{R}^{d\times d}$ *is the inverse of the a priori covariance given by Eq.* (6)*.*

[2] It also suffices that they are well-conditioned infinitely often, with $\boldsymbol{\beta}_*$ estimated only when they are. Note that this well-conditioning assumption holds throughout our experiments; nevertheless, it can also be enforced without affecting convexity guarantees by adding a small exploration component to the design (see App. E).

The proof is in App. E; in fact, we show that this even extends to more general model drift dynamics than Eq. (3). The convexity of Prob. (9) implies that it can be solved with standard convex optimization techniques [9]. Note that the monotonicity of G implies that the upper inequality constraint binds. Moreover, the fractional solution can be rounded to produce an approximate integral solution s.t. $\boldsymbol{n}_k \approx n_k \cdot \boldsymbol{\pi}_k$ via standard techniques; we review rounding schemes in App. B.

3.3.2 Optimal Steady-State Selection

Next, we turn our attention to stationary policies. To that end, we assume hereon that $\boldsymbol{\mathcal{X}}_k = \boldsymbol{\mathcal{X}}$ and $p_k = p$ (i.e., experiment options are fixed), and $\boldsymbol{n}_k = \boldsymbol{n} \in \mathbb{N}^p$, with $\boldsymbol{n}^\top \mathbf{1} = n$. We seek stationary policies $\boldsymbol{n} \in \mathbb{N}^p$ that are optimal in *steady state*. We first establish that a steady state indeed exists.

Convergence to Steady State. Under a stationary policy, the system determined by dynamics (3)–(4) is asymptotically stable:

Theorem 4. *Consider the system evolving via Eqs.* (3)–(4) *under a stationary policy $\boldsymbol{n} \in \mathbb{N}^p$, and assume that $\boldsymbol{X}(\boldsymbol{n}) \in \mathbb{R}^{n \times d}$ is rank d. Then, as $k \to \infty$, the error covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ converges to $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \in \mathbb{R}^{d \times d}$ that is a solution to the Discrete Algebraic Riccati Equation (DARE):*

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1} = \tfrac{1}{\sigma^2} \boldsymbol{X}(\boldsymbol{n})^\top \boldsymbol{X}(\boldsymbol{n}) + \left(\rho^2 \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1}, \tag{10}$$

while the Kalman gain $\boldsymbol{K}_k$ converges to:

$$\overline{\boldsymbol{K}} = \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \boldsymbol{X}(\boldsymbol{n})^\top \left(\boldsymbol{X}(\boldsymbol{n}) \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \boldsymbol{X}(\boldsymbol{n})^\top + \sigma^2 \mathbf{I}\right)^{-1}. \tag{11}$$

The proof is in App. F. Together, Thms. 2 and 4 imply the following. First, given a stationary policy $\boldsymbol{n} \in \mathbb{N}^p$, the posterior covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ converges to the solution of the DARE (10) *irrespective of whether $\boldsymbol{\beta}_*$ is known or not*. This is because neither $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ nor the DARE (10) depends on $\boldsymbol{\beta}_*$. Second, this convergence $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} \to \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ is *decoupled from the convergence $\hat{\boldsymbol{\beta}}_* \to \boldsymbol{\beta}_*$ in Thm. 2*: the covariance converges irrespective of whether $\hat{\boldsymbol{\beta}}_*$ converges (or is even used in the estimation of $\boldsymbol{\beta}_k$), while $\hat{\boldsymbol{\beta}}_*$ converges w.p. 1, for any stationary selection policy, as long as $\operatorname{rank}(\boldsymbol{X}(\boldsymbol{n})) = d$.

Optimal Stationary Policy. Armed with Thms. 1–4, we seek a stationary policy that minimizes the steady-state covariance, as given by the DARE (10). We stress again that covariance dynamics are identical regardless of whether the mean fixed-point $\boldsymbol{\beta}_*$ is known or estimated via Thm. 2. Formally, we wish to select $\boldsymbol{\pi} \in [0,1]^p$ that maximizes $\log\det\left[\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\right]$, where, by Thm. 4, $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi})$ is the solution to the (relaxed) DARE:

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1} = \frac{n}{\sigma^2} \boldsymbol{B}(\boldsymbol{\pi}) + \left(\rho^2 \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1}, \tag{12}$$

with $\boldsymbol{B}(\boldsymbol{\pi}) \equiv \sum_{i=1}^p \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$. Hence, finding the optimal fractional policy in steady state amounts to solving the following maximization problem:

$$\text{Max.: } G\left(\boldsymbol{\pi}; \sigma^2, \sigma_{\boldsymbol{w}}^2\right) \equiv \log\det\left[\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\right], \tag{13a}$$

$$\text{s.t.: } \boldsymbol{\pi}^\top \mathbf{1} \leq 1, \boldsymbol{\pi} \geq \mathbf{0}. \tag{13b}$$

Note that the map $\boldsymbol{\pi} \mapsto \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi})$ is defined *implicitly*, through the non-linear system of equations in Eq. (12), rather than through an explicit formula. To design a stationary policy that is optimal in steady state, we first try to relate how the steady-state covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ depends on the stationary fractional policy $\boldsymbol{\pi} \in [0,1]^p$. To that end, we prove an auxiliary lemma, establishing that matrices $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ and $\boldsymbol{B}(\boldsymbol{\pi})$ share the same eigenvectors.

Lemma 1. *Consider $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ that is a solution to*

$$\boldsymbol{\Sigma}^{-1} = \frac{n}{\sigma^2} \boldsymbol{B} + \left(\rho^2 \boldsymbol{\Sigma} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1}, \tag{14}$$

where $\boldsymbol{B} \in \mathbb{S}_+^d$ is a PSD matrix. For $i = 1, \ldots, d$, let λ_i, $\boldsymbol{e}_i$, be the eigenvalues and eigenvectors of $\boldsymbol{B}$, respectively. Then, $\boldsymbol{\Sigma} = \sum_{i=1}^d \lambda_i' \boldsymbol{e}_i \boldsymbol{e}_i^\top$, where, for $i = 1, \ldots, d$,

$$\lambda_i' = 2\sigma_{\boldsymbol{w}}^2 \frac{\sigma^2}{n} \cdot \left[\sqrt{\left(\tfrac{\sigma^2}{n}\left(1-\rho^2\right) + \sigma_{\boldsymbol{w}}^2 \lambda_i\right)^2 + 4\lambda_i \rho^2 \sigma_{\boldsymbol{w}}^2 \tfrac{\sigma^2}{n}} + \left(\tfrac{\sigma^2}{n}\left(1-\rho^2\right) + \sigma_{\boldsymbol{w}}^2 \lambda_i\right)\right]^{-1}. \tag{15}$$

We stress that Lemma 1, which we prove in App. G, *does not generally hold for DARE solutions of arbitrary drift dynamics (see Eq.* (29)*)*. One of our main technical contributions is thus to observe that, under dynamics (3), we can solve the DARE *explicitly*.

Lemma 1 also paves the way for determining the curvature of the D-optimal design objective in steady state. Leveraging this construction, we show that *Prob.* (13) *is a convex optimization problem*:

Theorem 5. *The objective function $G : \mathbb{R}^p \to \mathbb{R}$ defined in Eq.* (13a) *is non-decreasing and concave.*

The proof is in App. H. In light of the intricate relationship between $\boldsymbol{\pi}$ and $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ through Lemma 1, the proof is technically challenging; we provide a proof outline at the beginning of App. H. In short, we prove that the D-optimal design objective is non-decreasing and Schur-concave w.r.t. the eigenvalues of $\boldsymbol{B}(\boldsymbol{\pi})$, which in turn implies concavity over the positive semi-definite cone. The same machinery extends to the A-optimality (trace) criterion, which is likewise convex (App. I). An immediate consequence is that the optimal steady-state policy $\boldsymbol{\pi}$ can be computed efficiently via convex optimization techniques. A key challenge lies in computing the gradient ∇G, as this requires differentiating the steady-state covariance defined implicitly through the DARE system of equations (Eq. (12)). In App. J, we show how the gradient can be computed via the implicit function theorem [37], and solve Prob. (13) efficiently using the Frank–Wolfe algorithm [24].

3.4 Non-linear Setting

Our approach generalizes to the case when h is non-linear using the so-called *extended Kalman filter (EKF)* [8], which linearizes h around the current model estimate at each stage. Formally, the EKF linearizes h through its first-order Taylor approximation, so that

$$h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) \approx h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{H}_k(\boldsymbol{\beta}_k - \hat{\boldsymbol{\beta}}_{k|k-1}), \tag{16}$$

where $\boldsymbol{H}_k = \nabla_{\boldsymbol{\beta}} h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1}) \in \mathbb{R}^{n_k \times d_{\boldsymbol{\beta}}}$ is the Jacobian of h. In practice, this leads to the EKF adaptation equations that, compared to a standard Kalman filter, replace the design matrix $\boldsymbol{X}_k$ with the Jacobian $\boldsymbol{H}_k$. Our approach naturally extends to this setting, as discussed in App. K. A caveat is that, in the non-linear case, least-squares estimation is non-convex, so the mean fixed-point $\boldsymbol{\beta}_*$ may not be identifiable. Nevertheless, we can still produce a "status-quo" estimate, by minimizing the square-error loss ignoring dynamics. Just as in the linear case, this estimate does not play a role in the design of myopic policies. In contrast to the linear case, however, it does play a role in the design of a steady-state policy as, to compute the steady-state Jacobian, we exploit the fact that the model is centered around $\boldsymbol{\beta}_*$ under the dynamics in Eq. (3). As we discuss in the next section, both of these approximations (the linearization via Eq. (16) and our $\boldsymbol{\beta}_*$ estimate) work very well in practice.

4 Numerical Evaluation

4.1 Experimental Setup

Our experimental setup, including methods, datasets, and metrics, is described in detail in App. L. We briefly review it here. Our source code is publicly available[3].

Datasets. We use a synthetic dataset and six publicly available datasets, summarized in Table 1. We generate four versions of the synthetic dataset, all consistent with the dynamics in Eqs. (2)–(3), each using a different model function h: a linear model, ResNet [27], a convolutional neural network (CNN), and a multi-layer perceptron (MLP). The Beijing PM2.5 dataset [14] contains 5 years of hourly air quality measurements, where pollution dynamics are driven by seasonal weather patterns and policy interventions. The S&P dataset [51] contains daily stock prices for S&P 500 companies, which drift due to changing market regimes and volatility cycles. The Bike Sharing dataset [21] contains hourly rental counts from Washington, D.C., over 2011–2012; demand patterns drift with seasonal weather and growing ridership adoption. The California Housing dataset [52] contains aggregated 1990 US Census statistics in California; we induce drift by assigning different geographic regions to different stages. The AT&T Downlink and AT&T Uplink datasets contain a multi-carrier cellular study [13], collected via a cross-country drive (from Boston, MA to Los Angeles, CA) measurement campaign, traversing approximately 4,500 km across 13 US states.

[3] `https://github.com/neu-spiral/MarkovExp`

Table 1: Dataset features and model parameter dimensions. The dataset-feature block (p: candidate experiments per stage; n: training-set size per stage; d: feature dimension; T: number of stages) is uniform across methods. The non-linear $d_{\boldsymbol{\beta}}$ block reports the model parameter dimension per architecture (ResNet/CNN/MLP); in the linear setting $d_{\boldsymbol{\beta}} = d$.

Dataset	**Dataset Features**				**Non-linear $d_{\boldsymbol{\beta}}$**		
	p	n	d	T	ResNet	CNN	MLP
Synthetic	40	50	30	2000	97	98	97
Beijing PM2.5 [14]	19	30	7	1826	39	33	31
S&P 500 [51]	16	25	6	62	39	33	29
Bike Sharing [21]	19	30	8	724	39	33	33
California Housing [52]	40	55	6	412	39	33	29
AT&T Downlink [13]	16	25	12	335	39	33	41
AT&T Uplink [13]	16	25	8	328	39	33	33

Methods. We couple different *design selection* methods with different *model estimation* methods. For design selection, we consider five approaches: **Stage-wise Optimal Selection (STG)**, our optimal myopic design, described in Prob. (9); **Stationary Optimal Selection (STS)**, our optimal steady-state design policies for linear and non-linear settings, described in Prob. (13); **Greedy Kalman Information-Gain Selection (GKS)** [61], an external greedy sensor-selection baseline that sequentially selects the candidates of highest one-step information gain; **Adam Selection (ADM)**, which performs projected gradient ascent to solve Prob. (9) at each stage using the Adam optimizer [34]; and **Random Selection (RND)**, which samples feature vectors u.a.r. For model estimation, we adopt four approaches. **Adaptive Estimation (AE)** uses the Kalman filter and mean fixed-point estimates (Eqs. (5)–(8)), as well as their EKF variants (see App. K). We also consider a variant that assumes the mean fixed-point is known (**AE-K**). **Forgetting-factor Recursive Least Squares Estimation (RLS)** [43] is an external recursive baseline that tracks the drift with an exponential forgetting factor and a random-walk time update, without using the mean fixed-point $\boldsymbol{\beta}_*$. **Static Estimation (SE)** relies on priors and the design matrix along with the observed labels, without employing the recursive Kalman filter equations. Lastly, **Stochastic Gradient Descent Estimation (SGD)** employs a standard SGD update for the model estimation ignoring the impact of drift, run to convergence at each stage. Our methods are **STG-AE** and **STS-AE**.

Performance Metrics. We evaluate two metrics. In both synthetic and real datasets, we compute the **test RMSE** between predicted and test labels, evaluated over per-stage hold-out test sets. This metric is directly tied to our design objective: for a fresh input $\boldsymbol{x}$, the expected prediction error decomposes as $\mathrm{EPE}(\boldsymbol{x}) = \sigma^2 + \boldsymbol{x}^\top \boldsymbol{\Sigma}_{\boldsymbol{\xi}} \boldsymbol{x}$ (App. B, Eq. (33)), so a design that shrinks the posterior covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$, equivalently that attains a higher log-determinant objective, lowers the expected test error. This explains why, throughout our experiments, methods with a higher design objective attain a lower test RMSE (see App. M.2). In synthetic experiments, we also measure **Model Error** as the Euclidean distance between the true model $\boldsymbol{\beta}_k$ and its estimate $\hat{\boldsymbol{\beta}}_{k|k}$. All metrics are averaged over 100 runs with different random seeds, and we report mean and standard deviation.

4.2 Results

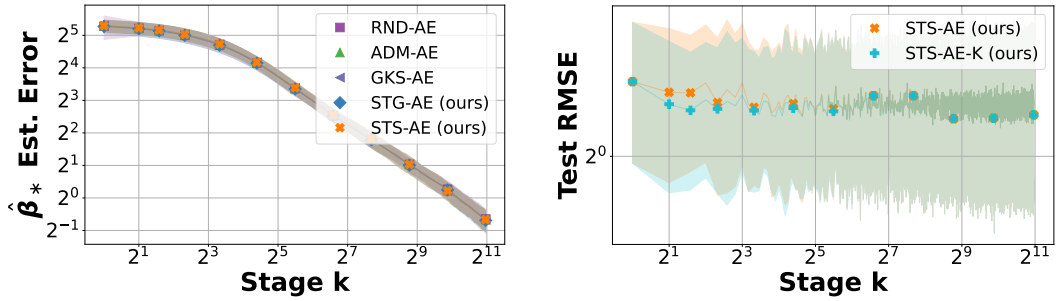
Table 2 reports the test RMSE for all methods across the four model architectures (Linear, ResNet, CNN, MLP) and seven datasets; to reduce clutter, we omit -K variants and report them in App. M. Across all settings, our methods STG-AE and STS-AE achieve the lowest test RMSE. Under our optimal selection, adaptive estimation (-AE) is superior to static (-SE), SGD-based, and forgetting-factor RLS (-RLS) estimation, confirming that the (extended) Kalman filter effectively tracks the drifting model. Among design selection strategies, our optimal policies (STG, STS) outperform greedy information-gain (GKS), Adam-based (ADM) and random (RND) selection, with the advantage most pronounced for non-linear models. Comparing architectures, the non-linear models achieve substantially lower error than the linear model on all real-world datasets, e.g., on Beijing PM2.5, the best non-linear method (STS-AE with ResNet, 87.28) improves over the best linear method (133.82) by over 34%. CNN and ResNet perform similarly, while MLP shows slightly higher variance on the synthetic scenario. Model error performance for synthetic scenarios is in App. M.

Table 2: Test RMSE on synthetic and real datasets for Linear, ResNet, CNN, and MLP. **Bold** indicates the best result per model and underline the second-best. Mean $\pm$ std reported over 100 random seeds.

Model	Method	Beijing PM2.5	S&P 500	Bike Sharing	Cal. Housing	AT&T Downlink	AT&T Uplink	Synthetic
Linear	RND-SGD	148.42 ± 1.20	2.95 ± 0.75	280.78 ± 2.18	4.11 ± 0.04	158.66 ± 1.85	25.41 ± 1.25	7.75 ± 0.09
	RND-SE	144.80 ± 0.87	2.77 ± 0.72	273.01 ± 0.57	3.81 ± 0.03	152.38 ± 1.27	23.07 ± 1.15	4.22 ± 0.18
	RND-AE	142.53 ± 0.50	2.71 ± 0.73	267.70 ± 1.00	3.72 ± 0.01	146.30 ± 1.69	20.77 ± 0.95	3.20 ± 0.05
	RND-RLS	142.48 ± 0.55	2.68 ± 0.71	265.57 ± 1.04	3.76 ± 0.03	147.43 ± 1.71	21.55 ± 0.92	5.48 ± 0.12
	ADM-SGD	147.91 ± 1.15	2.58 ± 0.68	280.63 ± 2.16	3.94 ± 0.05	153.44 ± 1.75	18.48 ± 1.35	7.57 ± 0.12
	ADM-SE	143.52 ± 0.76	2.43 ± 0.59	272.93 ± 0.42	3.76 ± 0.01	149.35 ± 1.25	13.12 ± 1.15	2.78 ± 0.10
	ADM-AE	141.46 ± 0.70	1.79 ± 0.57	265.40 ± 0.31	3.70 ± 0.02	142.34 ± 1.09	12.17 ± 0.58	2.74 ± 0.08
	ADM-RLS	142.15 ± 0.61	1.84 ± 0.62	264.42 ± 0.38	3.71 ± 0.02	145.31 ± 1.42	12.58 ± 0.74	4.92 ± 0.09
	GKS-SGD	146.05 ± 1.01	2.74 ± 0.71	278.97 ± 2.00	3.88 ± 0.06	156.18 ± 1.81	19.52 ± 1.31	6.98 ± 0.08
	GKS-SE	144.55 ± 0.69	2.61 ± 0.65	271.34 ± 0.48	3.78 ± 0.03	151.83 ± 1.32	17.63 ± 1.17	2.92 ± 0.10
	GKS-AE	141.65 ± 0.52	1.88 ± 0.55	263.01 ± 0.41	3.71 ± 0.02	143.51 ± 1.14	15.67 ± 0.93	2.79 ± 0.09
	GKS-RLS	141.13 ± 0.48	1.81 ± 0.51	264.44 ± 0.35	3.72 ± 0.02	145.57 ± 1.27	16.10 ± 1.08	7.15 ± 0.14
	STG-SGD	143.94 ± 1.01	2.51 ± 0.59	278.93 ± 2.00	3.89 ± 0.03	150.84 ± 1.03	17.85 ± 0.93	7.53 ± 0.11
	STG-SE	143.57 ± 0.69	2.55 ± 0.63	270.57 ± 0.47	3.88 ± 0.02	140.44 ± 1.30	13.18 ± 0.97	2.76 ± 0.07
	STG-RLS	140.53 ± 0.57	1.79 ± 0.55	260.73 ± 0.33	3.70 ± 0.02	139.14 ± 0.98	13.05 ± 0.84	5.59 ± 0.17
	STG-AE (ours)	138.61 ± 0.60	$\mathbf{1.68 \pm 0.51}$	252.97 ± 0.39	$\mathbf{3.66 \pm 0.01}$	138.97 ± 0.85	11.10 ± 0.81	2.66 ± 0.08
	STS-AE (ours)	$\mathbf{133.82 \pm 0.63}$	1.75 ± 0.53	$\mathbf{244.18 \pm 0.31}$	3.68 ± 0.01	$\mathbf{132.78 \pm 0.90}$	$\mathbf{10.51 \pm 0.78}$	$\mathbf{2.56 \pm 0.06}$
ResNet	RND-SGD	104.97 ± 1.23	1.78 ± 0.66	187.55 ± 1.11	2.51 ± 0.94	131.90 ± 0.93	13.46 ± 1.23	1.41 ± 0.09
	RND-SE	100.99 ± 3.00	0.80 ± 0.23	180.10 ± 2.45	1.85 ± 0.70	130.51 ± 0.98	8.30 ± 1.93	2.59 ± 0.50
	RND-AE	99.24 ± 0.39	0.76 ± 0.03	180.01 ± 0.35	1.32 ± 0.15	131.51 ± 0.33	7.66 ± 0.27	1.29 ± 0.10
	RND-RLS	99.89 ± 0.78	0.77 ± 0.05	182.88 ± 0.51	1.48 ± 0.23	129.76 ± 1.12	7.90 ± 0.39	2.18 ± 0.31
	ADM-SGD	107.95 ± 1.21	1.58 ± 0.66	187.55 ± 1.12	2.51 ± 0.94	131.89 ± 0.93	13.17 ± 1.23	1.37 ± 0.08
	ADM-SE	97.69 ± 3.02	0.77 ± 0.18	180.12 ± 2.46	1.85 ± 0.70	130.51 ± 0.99	8.21 ± 1.98	2.46 ± 0.47
	ADM-AE	96.31 ± 0.32	0.75 ± 0.02	177.88 ± 0.34	1.23 ± 0.12	127.44 ± 0.41	6.86 ± 0.23	1.29 ± 0.09
	ADM-RLS	97.50 ± 0.55	0.74 ± 0.05	175.46 ± 0.38	1.30 ± 0.21	129.98 ± 0.52	8.07 ± 0.37	2.14 ± 0.34
	GKS-SGD	107.56 ± 1.31	1.38 ± 0.66	187.34 ± 1.19	2.50 ± 0.94	131.72 ± 0.97	13.42 ± 1.22	1.34 ± 0.08
	GKS-SE	96.75 ± 1.19	1.03 ± 0.14	180.53 ± 2.37	1.69 ± 0.60	130.23 ± 0.96	12.14 ± 1.53	2.47 ± 0.29
	GKS-AE	94.89 ± 1.38	0.81 ± 0.05	179.93 ± 0.62	1.29 ± 0.19	129.45 ± 0.55	10.95 ± 0.86	1.11 ± 0.12
	GKS-RLS	93.42 ± 1.29	0.79 ± 0.11	178.13 ± 1.14	1.32 ± 0.18	130.63 ± 0.91	12.74 ± 0.95	2.57 ± 0.15
	STG-SGD	103.55 ± 1.29	1.48 ± 0.66	186.34 ± 1.18	2.50 ± 0.94	131.72 ± 0.97	13.45 ± 1.22	1.19 ± 0.08
	STG-SE	98.91 ± 2.99	0.75 ± 0.19	179.44 ± 2.47	1.96 ± 0.62	129.83 ± 0.97	8.40 ± 1.93	2.54 ± 0.45
	STG-RLS	95.26 ± 0.98	0.71 ± 0.03	175.95 ± 0.31	1.26 ± 0.19	128.35 ± 0.35	7.91 ± 0.43	2.24 ± 0.12
	STG-AE (ours)	92.22 ± 0.38	0.67 ± 0.03	$\mathbf{174.81 \pm 0.27}$	1.24 ± 0.17	123.32 ± 0.29	6.74 ± 0.32	0.98 ± 0.07
	STS-AE (ours)	$\mathbf{87.28 \pm 0.32}$	$\mathbf{0.60 \pm 0.04}$	177.88 ± 0.33	$\mathbf{1.19 \pm 0.16}$	$\mathbf{122.43 \pm 0.35}$	$\mathbf{6.60 \pm 0.34}$	$\mathbf{0.89 \pm 0.08}$
CNN	RND-SGD	106.66 ± 2.03	1.99 ± 1.17	187.14 ± 1.64	2.59 ± 1.05	132.06 ± 0.98	13.11 ± 1.16	1.46 ± 0.09
	RND-SE	97.66 ± 4.08	0.80 ± 0.11	181.95 ± 3.17	1.91 ± 0.59	130.27 ± 1.17	8.27 ± 1.98	2.78 ± 0.52
	RND-AE	96.32 ± 0.21	0.76 ± 0.02	180.92 ± 0.19	1.32 ± 0.10	131.46 ± 0.30	7.79 ± 0.27	1.39 ± 0.07
	RND-RLS	95.86 ± 0.58	0.79 ± 0.07	181.21 ± 0.28	1.28 ± 0.15	129.35 ± 0.47	7.95 ± 0.35	2.17 ± 0.26
	ADM-SGD	106.59 ± 2.03	1.99 ± 1.17	187.20 ± 1.58	2.59 ± 1.05	132.04 ± 0.99	13.11 ± 1.16	1.45 ± 0.09
	ADM-SE	97.64 ± 4.08	0.79 ± 0.09	180.99 ± 3.13	1.94 ± 0.56	130.27 ± 1.17	8.23 ± 2.01	3.01 ± 0.54
	ADM-AE	95.26 ± 0.24	0.74 ± 0.01	179.96 ± 0.16	1.25 ± 0.09	128.43 ± 0.25	6.76 ± 0.25	1.37 ± 0.07
	ADM-RLS	96.12 ± 0.39	0.77 ± 0.05	180.11 ± 0.19	1.27 ± 0.08	129.14 ± 0.53	7.51 ± 0.29	2.03 ± 0.23
	GKS-SGD	104.51 ± 2.11	1.98 ± 1.17	185.67 ± 2.17	2.58 ± 1.01	131.71 ± 0.99	13.05 ± 1.11	1.39 ± 0.08
	GKS-SE	97.15 ± 4.02	0.81 ± 0.08	181.70 ± 3.13	2.09 ± 0.41	130.43 ± 1.13	8.30 ± 1.70	2.61 ± 0.35
	GKS-AE	95.31 ± 0.22	0.73 ± 0.03	178.92 ± 0.18	1.26 ± 0.15	128.45 ± 0.24	6.89 ± 0.28	1.38 ± 0.08
	GKS-RLS	96.05 ± 0.61	0.76 ± 0.07	178.41 ± 0.18	1.25 ± 0.13	128.34 ± 0.35	7.38 ± 0.32	2.15 ± 0.28
	STG-SGD	104.52 ± 3.13	1.98 ± 1.17	184.87 ± 2.67	2.58 ± 1.05	131.71 ± 0.99	13.05 ± 1.10	1.42 ± 0.08
	STG-SE	98.12 ± 4.07	0.80 ± 0.14	181.63 ± 3.15	2.12 ± 0.48	130.42 ± 1.13	8.17 ± 1.93	2.97 ± 0.56
	STG-RLS	94.98 ± 0.25	0.71 ± 0.03	178.98 ± 0.20	1.24 ± 0.13	125.86 ± 0.21	6.42 ± 0.24	2.03 ± 0.21
	STG-AE (ours)	93.23 ± 0.17	0.66 ± 0.04	177.93 ± 0.15	$\mathbf{1.20 \pm 0.12}$	123.47 ± 0.26	5.86 ± 0.22	1.39 ± 0.12
	STS-AE (ours)	$\mathbf{88.23 \pm 0.13}$	$\mathbf{0.62 \pm 0.04}$	$\mathbf{173.93 \pm 0.17}$	1.24 ± 0.10	$\mathbf{121.39 \pm 0.27}$	$\mathbf{5.78 \pm 0.26}$	$\mathbf{1.35 \pm 0.10}$
MLP	RND-SGD	104.59 ± 1.43	1.53 ± 0.66	187.34 ± 1.32	2.46 ± 1.03	134.37 ± 1.49	13.06 ± 1.19	3.93 ± 0.57
	RND-SE	96.46 ± 5.37	0.82 ± 0.13	181.75 ± 2.22	1.38 ± 0.09	132.21 ± 1.96	9.95 ± 2.31	4.61 ± 0.74
	RND-AE	95.70 ± 0.65	0.78 ± 0.03	180.80 ± 0.26	1.35 ± 0.12	128.36 ± 0.35	9.74 ± 0.34	3.86 ± 0.62
	RND-RLS	96.10 ± 0.78	0.80 ± 0.04	180.79 ± 0.29	1.35 ± 0.18	128.89 ± 0.53	9.88 ± 0.36	4.27 ± 0.50
	ADM-SGD	104.59 ± 1.43	1.53 ± 0.66	186.35 ± 1.32	2.43 ± 1.03	132.19 ± 1.50	13.02 ± 1.09	3.70 ± 0.53
	ADM-SE	94.46 ± 5.37	0.79 ± 0.13	180.80 ± 2.14	1.37 ± 0.09	130.44 ± 1.88	8.89 ± 2.39	4.34 ± 0.67
	ADM-AE	93.65 ± 0.76	0.75 ± 0.02	179.99 ± 0.32	1.34 ± 0.11	127.46 ± 0.31	7.95 ± 0.41	3.61 ± 0.55
	ADM-RLS	93.18 ± 0.80	0.77 ± 0.03	179.46 ± 0.38	1.33 ± 0.14	128.12 ± 0.38	8.40 ± 0.39	4.06 ± 0.50
	GKS-SGD	104.51 ± 1.48	1.51 ± 0.62	187.15 ± 1.28	2.42 ± 1.01	132.07 ± 1.30	13.05 ± 1.16	3.91 ± 0.54
	GKS-SE	94.69 ± 5.30	0.92 ± 0.10	181.47 ± 2.14	1.38 ± 0.12	130.58 ± 1.81	8.91 ± 1.71	4.48 ± 0.71
	GKS-AE	93.88 ± 0.89	0.81 ± 0.08	179.85 ± 0.30	1.34 ± 0.11	128.10 ± 0.35	8.46 ± 0.35	3.59 ± 0.54
	GKS-RLS	94.12 ± 0.77	0.79 ± 0.06	178.92 ± 0.28	1.33 ± 0.17	127.85 ± 0.35	8.53 ± 0.31	4.18 ± 0.47
	STG-SGD	104.59 ± 1.43	1.53 ± 0.66	187.07 ± 1.47	2.41 ± 1.03	132.08 ± 1.29	12.89 ± 1.11	3.98 ± 0.57
	STG-SE	94.33 ± 5.34	0.76 ± 0.13	181.14 ± 2.14	1.37 ± 0.10	130.18 ± 1.87	8.98 ± 2.32	4.90 ± 0.82
	STG-RLS	92.98 ± 0.80	0.74 ± 0.03	178.12 ± 0.29	1.33 ± 0.15	126.65 ± 0.31	8.25 ± 0.30	4.06 ± 0.58
	STG-AE (ours)	91.65 ± 0.86	$\mathbf{0.69 \pm 0.02}$	177.97 ± 0.32	$\mathbf{1.28 \pm 0.12}$	124.29 ± 0.34	8.21 ± 0.33	2.98 ± 0.65
	STS-AE (ours)	$\mathbf{89.70 \pm 0.68}$	0.72 ± 0.03	$\mathbf{174.99 \pm 0.33}$	1.31 ± 0.16	$\mathbf{121.45 \pm 0.29}$	$\mathbf{7.90 \pm 0.29}$	$\mathbf{2.63 \pm 0.57}$

Fig. 2 shows the convergence of $\hat{\boldsymbol{\beta}}_*$ to $\boldsymbol{\beta}_*$ for the linear model in the synthetic scenario, under STS-AE, STG-AE, ADM-AE, RND-AE, and GKS-AE. We also compare the test RMSE for STS-AE and STS-AE-K, where $\boldsymbol{\beta}_*$ is known. We report additional comparisons of AE methods with their -K variants, as well as the convergence of $\hat{\boldsymbol{\beta}}_*$ to $\boldsymbol{\beta}_*$, in App. M. We observe that the methods closely track their -K variants, and that the estimator approximates $\boldsymbol{\beta}_*$.

In Fig. 3, over the ResNet model, we vary the system parameters ρ, σ, and $\sigma_{\boldsymbol{w}}$ for the synthetic dataset, each independently with the others fixed at their default values (Table 4). Across all parameter settings,



(a) Mean fixed-point estimator convergence

(b) Unknown vs. known β_*, test RMSE

Figure 2: Representative results for the linear model on the synthetic dataset: (a) estimation error between $\hat{\beta}_*$ and the true mean fixed-point β_*; (b) test RMSE for STS-AE when β_* is unknown (estimated online) vs. known. Lower is better; shaded regions represent ± 1 standard deviation.

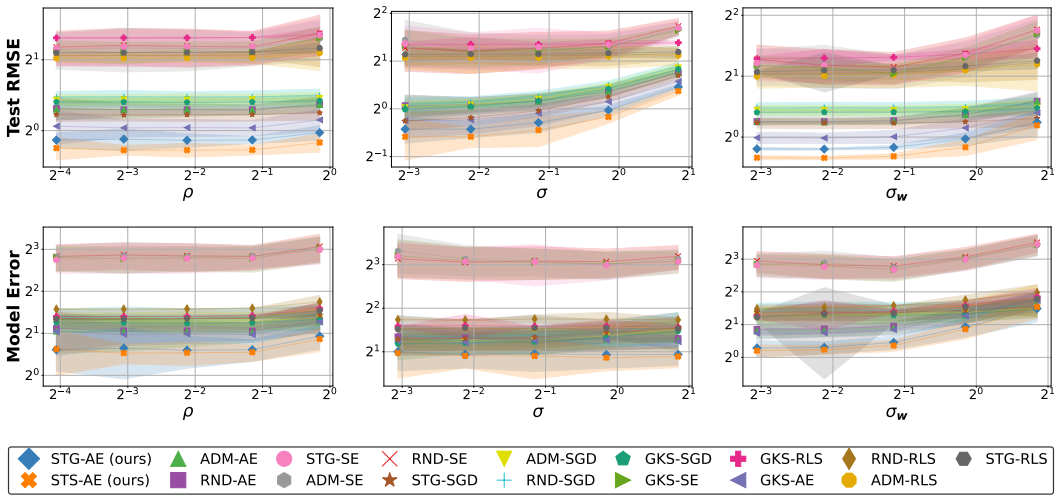


Figure 3: Test RMSE (Eq. (146), top row) and model error (Eq. (149), bottom row) comparisons with respect to different values of ρ, σ, and $\sigma_{\boldsymbol{w}}$ on the synthetic dataset for the ResNet model; other parameters as in Table 4. Lower is better. Shaded regions represent ± 1 standard deviation.

our proposed methods achieve lower test RMSE and model error than the baselines, demonstrating robustness under varying system dynamics. In particular, as ρ decreases (faster drift), the gap between adaptive (-AE) and the (-SE), (-SGD), and (-RLS) baselines widens, highlighting the importance of Kalman-based tracking when drift is strong. Similarly, as $\sigma_{\boldsymbol{w}}$ increases (larger drift noise), our optimal selection policies (STG, STS) maintain a clear advantage over GKS, ADM and RND, indicating that informed sample selection becomes increasingly valuable when the model changes rapidly. We report the impacts of these parameters on the other models in App. M.

5 Conclusion

We propose *Markovian Experimental Design*, to determine an optimal data selection strategy over a model that is drifting according to linear dynamics coupled with Gaussian noise. The underlying stage-wise and steady-state optimization problems can be solved using convex optimization techniques. Several interesting research questions remain open. More complex model drift dynamics, beyond Eq. (3), require reasoning about the concavity of the D-optimality objective, or another scalarization of the covariance, without explicitly solving the discrete algebraic Riccati equation. Another question of interest is whether the end-to-end optimization of the design selection admits a tractable, non-myopic, dynamic policy. Dynamic policies can also be studied in the adversarial setting, akin to Si Salem [62], removing the stochasticity of model dynamics. We discuss limitations in App. N.

Acknowledgments and Disclosure of Funding

We gratefully acknowledge the support from the National Science Foundation under grants 2106891, 2107062, 2112471, and the National Science Foundation CAREER award under grant 2340482.

References

[1] Amani Al-Shawabka, Francesco Restuccia, Salvatore D'Oro, Tong Jian, Bruno Costa Rendon, Nasim Soltani, Jennifer Dy, Stratis Ioannidis, Kaushik Chowdhury, and Tommaso Melodia. Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 646–655. IEEE, 2020.

[2] Mohammad Alali and Mahdi Imani. Deep reinforcement learning data collection for Bayesian inference of hidden Markov models. *IEEE transactions on artificial intelligence*, 2024.

[3] Billy Amzal, Frédéric Y Bois, Eric Parent, and Christian P Robert. Bayesian-optimal design via interacting particle systems. *Journal of the American Statistical association*, 101(474):773–785, 2006.

[4] Brian DO Anderson and John B Moore. *Optimal filtering*. Courier Corporation, 2005.

[5] Stephen Bach and Mark Maloof. A Bayesian approach to concept drift. *Advances in neural information processing systems*, 23, 2010.

[6] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86, 2006.

[7] Albert Benveniste, Michel Métivier, and Pierre Priouret. *Adaptive algorithms and stochastic approximations*, volume 22. Springer Science & Business Media, 2012.

[8] Gary Bishop, Greg Welch, et al. An introduction to the Kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

[9] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[10] Julian Bright, Alessandro Cerè, Georgios Schinas, and Theiss Heilker. Automate model retraining with Amazon SageMaker Pipelines when drift is detected. https://aws.amazon.com/blogs/machine-learning/automate-model-retraining-with-amazon-sagemaker-pipelines-when-drift-is-detected/, 2021.

[11] Huanhuan Cao, Enhong Chen, Jie Yang, and Hui Xiong. Enhancing recommender systems under volatile userinterest drifts. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1257–1266, 2009.

[12] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical science*, pages 273–304, 1995.

[13] Fukun Chen, Moinak Ghoshal, Enfu Nan, Phuc Dinh, Imran Khan, Z Jonny Kong, Y Charlie Hu, and Dimitrios Koutsonikolas. A large-scale study of the potential of multi-carrier access in the 5g era. In *International Conference on Passive and Active Network Measurement*, pages 469–484. Springer, 2025.

[14] Song Chen. Beijing PM2.5. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5JS49.

[15] Susan Corbett. The retention of personal information online: A call for international regulation of privacy law. *Computer Law & Security Review*, 29(3):246–254, 2013.

[16] Anton Dries and Ulrich Rückert. Adaptive concept drift detection. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):311–327, 2009.

[17] Hovav A Dror and David M Steinberg. Sequential experimental designs for generalized linear models. *Journal of the American Statistical Association*, 103(481):288–298, 2008.

[18] Christopher C Drovandi, James M McGree, and Anthony N Pettitt. Sequential monte carlo for Bayesian sequentially designed experiments for discrete data. *Computational Statistics & Data Analysis*, 57(1):320–335, 2013.

[19] Shamak Dutta, Nils Wilde, and Stephen L Smith. A unified approach to optimally solving sensor scheduling and sensor selection problems in kalman filtering. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 1170–1176. IEEE, 2023.

[20] Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

[21] Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2(2):113–127, 2014.

[22] William Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 1991.

[23] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential Bayesian experimental design. In *International conference on machine learning*, pages 3384–3395. PMLR, 2021.

[24] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[25] Robert G Gallager. *Stochastic processes: Theory for Applications*. Cambridge University Press, 2013.

[26] William W Hager. Updating the inverse of a matrix. *SIAM review*, 31(2):221–239, 1989.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Xun Huan, Jayanth Jagalur, and Youssef Marzouk. Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840, 2024.

[29] Xun Huan and Youssef M Marzouk. Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317, 2013.

[30] Xun Huan and Youssef M Marzouk. Sequential Bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320*, 2016.

[31] Aditi Jha, Zoe C Ashwood, and Jonathan W Pillow. Active learning for discrete latent variable models. *Neural computation*, 36(3):437–474, 2024.

[32] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[33] Jarek Kazmierczak, Khalid Salama, and Valentin Huerta. MLOps: Continuous delivery and automation pipelines in machine learning. https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning, 2024. Google Cloud Architecture center.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] J Zico Kolter and Marcus A Maloof. Dynamic weighted majority: An ensemble method for drifting concepts. *The Journal of Machine Learning Research*, 8:2755–2790, 2007.

[36] Ivan Koychev. Gradual forgetting for adaptation to concept drift. In *Proceedings of the ECAI 2000 Workshop on Current Issues in Spatio-Temporal Reasoning*, 2000.

[37] Steven George Krantz and Harold R Parks. *The implicit function theorem: history, theory, and applications*. Springer Science & Business Media, 2002.

[38] Ludmila I Kuncheva. Classifier ensembles for changing environments. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2004.

[39] Harold J Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*. Springer, 2003.

[40] Lap Chi Lau, Robert Wang, and Hong Zhou. Experimental design using interlacing polynomials. In *2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 452–464. SIAM, 2025.

[41] Lap Chi Lau and Hong Zhou. A local search framework for experimental design. *SIAM Journal on Computing*, 51(4):900–951, 2022.

[42] Jinpeng Li, Hang Yu, Zhenyu Zhang, Xiangfeng Luo, and Shaorong Xie. Concept drift adaptation by exploiting drift type. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–22, 2024.

[43] Lennart Ljung and Torsten Söderström. *Theory and practice of recursive identification*. MIT press, 1983.

[44] Andreas Lommatzsch, Benjamin Kille, and Sahin Albayrak. Incorporating context and trends in news recommender systems. In *Proceedings of the international conference on web intelligence*, pages 1062–1068, 2017.

[45] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

[46] Horia Mania, Ali Jadbabaie, Devavrat Shah, and Suvrit Sra. Time varying regression with hidden linear dynamics. In *Learning for Dynamics and Control Conference*, pages 858–869. PMLR, 2022.

[47] Albert W Marshall, Ingram Olkin, and Barry C Arnold. *Inequalities: theory of majorization and its applications*. Springer, 1979.

[48] Peter Müller, Don A Berry, Andy P Grieve, Michael Smith, and Michael Krams. Simulation-based sequential Bayesian design. *Journal of statistical planning and inference*, 137(10):3140–3150, 2007.

[49] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[50] Aleksandar Nikolov, Mohit Singh, and Uthaipon Tantipongpipat. Proportional volume sampling and approximation algorithms for a-optimal design. *Mathematics of Operations Research*, 47(2):847–877, 2022.

[51] Cam Nugent. S&P 500 stock data. Kaggle, 2018. `https://www.kaggle.com/datasets/camnugent/sandp500`.

[52] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[53] Thrasyvoulos Pappas, Alan Laub, and Nils Sandell. On the numerical solution of the discrete-time algebraic Riccati equation. *IEEE Transactions on Automatic Control*, 25(4):631–641, 1980.

[54] Thomas Parr, Karl Friston, and Peter Zeidman. Active data selection and information seeking. *Algorithms*, 17(3):118, 2024.

[55] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[56] Randy J Read, Robert D Oeffner, and Airlie J McCoy. Measuring and using information gained by observing diffraction data. *Biological Crystallography*, 76(3):238–247, 2020.

[57] Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Mauro Belgiovine, Luca Angioloni, Stratis Ioannidis, Kaushik Chowdhury, and Tommaso Melodia. Deepradioid: Real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms. In *Proceedings of the twentieth ACM international symposium on mobile ad hoc networking and computing*, pages 51–60, 2019.

[58] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

[59] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[60] Kunal Sankhe, Mauro Belgiovine, Fan Zhou, Shamnaz Riyaz, Stratis Ioannidis, and Kaushik Chowdhury. Oracle: Optimized radio classification through convolutional neural networks. In *IEEE INFOCOM 2019-IEEE conference on computer communications*, pages 370–378. IEEE, 2019.

[61] Manohar Shamaiah, Siddhartha Banerjee, and Haris Vikalo. Greedy sensor selection: Leveraging submodularity. In *49th IEEE conference on decision and control (CDC)*, pages 2572–2577. IEEE, 2010.

[62] Tareq Si Salem. Adaptive transductive inference via sequential experimental design with contextual retention. In *NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty*, 2024.

[63] Mohit Singh and Weijun Xie. Approximate positive correlated distributions and approximation algorithms for d-optimal design. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2240–2255. SIAM, 2018.

[64] Arno Strouwen, Bart M Nicolaï, and Peter Goos. Adaptive and robust experimental design for linear dynamical models using Kalman filter. *Statistical Papers*, 64(4):1209–1231, 2023.

[65] Dries Telen, Dominique Vercammen, Filip Logist, and Jan Van Impe. Robustifying optimal experiment design for nonlinear, dynamic (bio) chemical systems. *Computers & chemical engineering*, 71:415–425, 2014.

[66] Christopher Tosh, Mauricio Tec, and Wesley Tansey. Targeted active learning for probabilistic models. *arXiv preprint arXiv:2210.12122*, 2022.

[67] Alexander Tsesis. Data subjects' privacy rights: regulation of personal data retention and erasure. *U. Colo. L. Rev.*, 90:593, 2019.

[68] Abraham Wald. *Sequential analysis*. Courier Corporation, 1947.

[69] Ke Wan, Yi Liang, and Susik Yoon. Online drift detection with maximum concept discrepancy. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2924–2935, 2024.

[70] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235, 2003.

[71] Shuo Wang, Leandro L Minku, and Xin Yao. A systematic study of online class imbalance learning with concept drift. *IEEE transactions on neural networks and learning systems*, 29(10):4802–4821, 2018.

[72] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[73] Lintao Ye, Sandip Roy, and Shreyas Sundaram. On the complexity and approximability of optimal sensor selection for kalman filtering. In *2018 Annual American Control Conference (ACC)*, pages 5049–5054. IEEE, 2018.

Table 3: Summary of Notation

Markovian Experimental Design		Bayesian Linear Regression & Experimental Design	
$n_k \in \mathbb{N}$	# of samples at stage k	$n \in \mathbb{N}$	# of samples
$\boldsymbol{\mathcal{X}}_k \subset \mathbb{R}^d$	candidate feature vectors at stage k	$\boldsymbol{\mathcal{X}} \subset \mathbb{R}^d$	candidate feature vectors
$p_k \in \mathbb{N}$	# of candidate feature vectors $\lvert\boldsymbol{\mathcal{X}}_k\rvert$	$p \in \mathbb{N}$	# of candidate feature vectors $\lvert\boldsymbol{\mathcal{X}}\rvert$
$\boldsymbol{\beta}_k \in \mathbb{R}^d$	model at stage k	$\boldsymbol{\beta} \in \mathbb{R}^d$	model
$\boldsymbol{X}_k \in \mathbb{R}^{n_k \times d}$	design matrix at stage k	$\boldsymbol{X} \in \mathbb{R}^{n \times d}$	design matrix
$\boldsymbol{y}_k \in \mathbb{R}^{n_k}$	labels at stage k	$\boldsymbol{y} \in \mathbb{R}^n$	labels
$\boldsymbol{n}_k \in \mathbb{N}^{p_k}$	integral selection policy at stage k	$\boldsymbol{n} \in \mathbb{N}^p$	integral selection policy
$\boldsymbol{\pi}_k \in [0,1]^{p_k}$	fractional selection policy at stage k	$\boldsymbol{\pi} \in [0,1]^p$	fractional selection policy
$\hat{\boldsymbol{\beta}}_k \in \mathbb{R}^d$	MAP estimate at stage k ($\equiv \hat{\boldsymbol{\beta}}_{k\lvert k}$)	$\hat{\boldsymbol{\beta}} \in \mathbb{R}^d$	MAP estimate
$\boldsymbol{\xi}_k \in \mathbb{R}^d$	error $\hat{\boldsymbol{\beta}}_k - \boldsymbol{\beta}_k$	$\boldsymbol{\xi} \in \mathbb{R}^d$	error $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$
$\boldsymbol{\epsilon}_k \in \mathbb{R}^{n_k}$	additive i.i.d. noise at stage k	$\boldsymbol{\epsilon} \in \mathbb{R}^n$	additive i.i.d. noise
$\boldsymbol{w}_k \in \mathbb{R}^d$	Gaussian model drift at stage k	Common Notation	
$\boldsymbol{A} \in \mathbb{R}^{d \times d}$	state transition matrix	$d \in \mathbb{N}$	# of features
$\sigma_{\boldsymbol{w}}^2 \in \mathbb{R}_+$	Gaussian model drift variance	$\sigma^2 \in \mathbb{R}_+$	additive noise variance
$\hat{\boldsymbol{\beta}}_{k\lvert k-1} \in \mathbb{R}^d$	MAP estimate at stage k before observing $(\boldsymbol{X}_k, \boldsymbol{y}_k)$	$\boldsymbol{\beta}_0 \in \mathbb{R}^d$	prior mean
$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k\lvert k-1}} \in \mathbb{R}^{d \times d}$	covariance estimate at stage k before observing $(\boldsymbol{X}_k, \boldsymbol{y}_k)$	$\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d \times d}$	prior covariance
$\hat{\boldsymbol{\beta}}_{k\lvert k} \in \mathbb{R}^d$	MAP estimate at stage k after observing $(\boldsymbol{X}_k, \boldsymbol{y}_k)$	$\boldsymbol{X}(\cdot)$	design matrix map (Eq. (38))
$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k\lvert k}} \in \mathbb{R}^{d \times d}$	covariance estimate at stage k after observing $(\boldsymbol{X}_k, \boldsymbol{y}_k)$	$\boldsymbol{B}(\cdot) \in \mathbb{R}^{d \times d}$	weighted sum of outer products $\boldsymbol{x}_i \boldsymbol{x}_i^\top$
$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \in \mathbb{R}^{d \times d}$	steady-state covariance		
$\boldsymbol{\beta}_* \in \mathbb{R}^d$	mean fixed-point	$G(\cdot)$	D-optimal design objective
$\hat{\boldsymbol{\beta}}_* \in \mathbb{R}^d$	mean fixed-point estimate		
$\rho \in (0,1)$	interpolation parameter		
$\overline{\boldsymbol{K}} \in \mathbb{R}^{d \times n}$	steady-state Kalman gain		

A Dynamical System Estimation using Kalman Filters

For completeness, we review dynamical system estimation via Kalman filters. The topic is classic; we refer the interested reader to Gallager [25] and Anderson & Moore [4], whom we follow in the exposition below.

A.1 Dynamical System Model

Consider a dynamical system that consists of a sequence of multivariate Gaussian random variables $\{\boldsymbol{\beta}_k\}_{k=1}^{\infty} \in \mathbb{R}^d$ where $\boldsymbol{\beta}_1$ is sampled from a prior normal distribution with mean $\boldsymbol{\beta}_0 \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d \times d}$, i.e., $\boldsymbol{\beta}_1 \sim \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$, and $\boldsymbol{\beta}_{k+1}$ evolves from $\boldsymbol{\beta}_k$ for $k \geq 1$ as follows:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{A}_k \boldsymbol{\beta}_k + \boldsymbol{w}_k, \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}_k}), \tag{17}$$

where $\boldsymbol{A}_k \in \mathbb{R}^{d \times d}$ is a known state transition matrix and $\boldsymbol{w}_k \in \mathbb{R}^d$ is the process noise for $k \geq 1$ with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{w}_k} \in \mathbb{R}^{d \times d}$. In addition, observations $\boldsymbol{y}_k \in \mathbb{R}^n$ are given by:

$$\boldsymbol{y}_k = \boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{e}_k, \qquad \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{\boldsymbol{e}_k}), \tag{18}$$

where each measurement matrix yields $\boldsymbol{X}_k = \left[\boldsymbol{x}_{ik}^\top\right]_{i=1}^n \in \mathbb{R}^{n \times d}$ and $\boldsymbol{e}_k \in \mathbb{R}^n$ is the measurement noise for $k \geq 1$ with covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{e}_k} \in \mathbb{R}^{n \times n}$. Assuming the initial state $\boldsymbol{\beta}_1$ and the noise vectors at each stage $\boldsymbol{w}_k$ and $\boldsymbol{e}_k$ are all mutually independent, the Kalman filter computes the MAP estimate of $\boldsymbol{\beta}_k$ at each stage [32, 25]. The Kalman filter is a recursive algorithm for state estimation in systems with uncertain and noisy measurements, optimally combining past observations and a mathematical model to estimate the current state by minimizing the mean squared error between the estimated and true states. Being modeled on a Markov chain, it requires only the estimated state from the previous time step and the current measurement to compute the current estimate, unlike batch estimation techniques that need a history of observations and estimates [32, 25].

A.2 Estimation

Formally, the notation $\hat{\boldsymbol{\beta}}_{k|k'}$ defines each stage estimation of the model vector at stage k given observations up to and including stage $k' \leq k$. The respective errors are defined as $\boldsymbol{\xi}_{k|k'} = \hat{\boldsymbol{\beta}}_{k|k'} - \boldsymbol{\beta}_k$. Each stage is represented by $\hat{\boldsymbol{\beta}}_{k|k}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$, which are the *a posteriori* stage estimate mean and estimate covariance matrix at stage k given the observations up to and including stage k respectively. Kalman filter equations can be divided into two phases, which are *a priori* (predict) and *a posteriori* (update) phases. The *a priori* phase of the Kalman filter equations is defined as [25, 8]:

$$\hat{\boldsymbol{\beta}}_{k|k-1} = \boldsymbol{A}_k \hat{\boldsymbol{\beta}}_{k-1|k-1}, \tag{19}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} = \boldsymbol{A}_k \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k-1|k-1}} \boldsymbol{A}_k^\top + \boldsymbol{\Sigma}_{\boldsymbol{w}_k}. \tag{20}$$

After we see the evidence at stage k, we compute the Kalman gain as [8]:

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} \boldsymbol{X}_k^\top \left(\boldsymbol{X}_k \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} \boldsymbol{X}_k^\top + \boldsymbol{\Sigma}_{\boldsymbol{e}_k} \right)^{-1}, \tag{21}$$

and we update the state and covariance estimate as below, getting the *a posteriori* estimates at stage k [8]:

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{K}_k \left(\boldsymbol{y}_k - \boldsymbol{X}_k \hat{\boldsymbol{\beta}}_{k|k-1} \right), \tag{22}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\mathbf{I} - \boldsymbol{K}_k \boldsymbol{X}_k \right) \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}. \tag{23}$$

Here, the Kalman gain $\boldsymbol{K}_k$ balances the contribution of the measurement $\boldsymbol{X}_k$ to minimize the posterior covariance by adjusting the trade-off between model predictions and new observations. By using the Woodbury matrix identity [26] on Eq. (23), the *a posteriori* covariance is alternatively defined without utilizing the Kalman gain as:

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\boldsymbol{X}_k^\top \boldsymbol{\Sigma}_{\boldsymbol{e}_k}^{-1} \boldsymbol{X}_k + \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}^{-1} \right)^{-1}. \tag{24}$$

The *a priori* estimate at $k = 1$ and the respective covariance are given by [25, 8]:

$$\hat{\boldsymbol{\beta}}_{1|0} = \boldsymbol{\beta}_0, \tag{25}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{1|0}} = \boldsymbol{\Sigma}_0. \tag{26}$$

The Kalman filter provides an optimal estimation by assuming that the errors $\boldsymbol{w}_k$ and $\boldsymbol{e}_k$ have a normal distribution. Moreover, regardless of the normal distribution, if the noise covariances $\boldsymbol{\Sigma}_{\boldsymbol{w}_k}$ and $\boldsymbol{\Sigma}_{\boldsymbol{e}_k}$ are known, then the Kalman filter serves as the optimal linear estimator [8].

A.3 Steady-State Kalman Filter

We review in this section asymptotic stability results pertaining to Kalman filters. We briefly introduce several definitions first.

Definition 1. A Kalman filter described in Eqs. (17)–(26) is time invariant if the system parameters are independent from time, i.e., the process noise covariance $\boldsymbol{\Sigma}_{\boldsymbol{w}_k}$ and the measurement noise covariance $\boldsymbol{\Sigma}_{\boldsymbol{e}_k}$ are constant over time, the measurement matrix $\boldsymbol{X}_k$ and the state transition matrix $\boldsymbol{A}_k$ are constant or converge to constant matrices for all k.

Definition 2 ([4])**.** A pair of matrices $(\boldsymbol{F}, \boldsymbol{G})$ is stabilizable if there exists a matrix $\boldsymbol{H}$ such that:

$$|\lambda_i \left(\boldsymbol{F} + \boldsymbol{G}\boldsymbol{H} \right)| < 1, \tag{27}$$

for all i, i.e., the matrix $\boldsymbol{F} + \boldsymbol{G}\boldsymbol{H}$ is stable.

Definition 3 ([4])**.** A pair of matrices $(\boldsymbol{F}, \boldsymbol{G})$ is detectable if the dual system pair $\left(\boldsymbol{F}^\top, \boldsymbol{G} \right)$ is stabilizable (see Def. 2).

Armed with these definitions, we state here a classic convergence theorem regarding the steady state of Kalman filters:

Theorem 6 ([4]). *If the Kalman filter described in Eqs.* (17)–(26) *is time invariant (see Def. 1), and asymptotically stable, i.e.,* $|\lambda_i(\boldsymbol{A})| < 1$*, then the covariance and the Kalman gain converge to stationary matrices* $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ *and* $\overline{\boldsymbol{K}}$ *respectively. If the Kalman filter is time invariant (see Def. 1) but not necessarily asymptotically stable, i.e., possibly* $|\lambda_i(\boldsymbol{A})| \geq 1$ *for some* i*, but the pair* $\left(\boldsymbol{A}, \boldsymbol{X}^\top\right)$ *is detectable (Def. 3), and the pair* $(\boldsymbol{A}, \boldsymbol{G})$ *is stabilizable for any* $\boldsymbol{G}$ *with* $\boldsymbol{G}\boldsymbol{G}^\top = \boldsymbol{\Sigma}_{\boldsymbol{w}}$ *(Def. 2), then the covariance and the Kalman gain still converge to stationary matrices* $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ *and* $\overline{\boldsymbol{K}}$ *respectively.*

For both of the conditions, the Kalman filter gain and covariance matrices converge to stationary matrices as $k \to \infty$ [4]. The steady-state covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ converges to a constant matrix $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$, that satisfies the discrete algebraic Riccati equation (DARE) [4]. The equation is derived by equating $(\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k-1|k-1}})$ via Eqs. (20) and (23):

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \boldsymbol{A}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_{\boldsymbol{w}} - \boldsymbol{A}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{X}^\top \left(\boldsymbol{X}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{X}^\top + \boldsymbol{\Sigma}_{\boldsymbol{e}}\right)^{-1} \boldsymbol{X}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{A}^\top, \tag{28}$$

where $\boldsymbol{A}$, $\boldsymbol{X}$, and $\boldsymbol{\Sigma}_{\boldsymbol{e}}$ are constant at steady state. Using the Woodbury-applied alternative of the posterior covariance in Eq. (24), the DARE can be re-written as:

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \left[\left(\boldsymbol{A}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{A}^\top + \boldsymbol{\Sigma}_{\boldsymbol{w}}\right)^{-1} + \boldsymbol{X}^\top\boldsymbol{\Sigma}_{\boldsymbol{e}}^{-1}\boldsymbol{X}\right]^{-1}. \tag{29}$$

This equation provides the unique solution for the steady-state covariance [4, 53]. Additionally, the Kalman gain converges to:

$$\overline{\boldsymbol{K}} = \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{X}^\top \left(\boldsymbol{X}\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\boldsymbol{X}^\top + \boldsymbol{\Sigma}_{\boldsymbol{e}}\right)^{-1}. \tag{30}$$

B Bayesian Linear Regression and Experimental Design

In this section, we review the fundamentals of Bayesian Linear Regression and Experimental Design. We also formalize the connection between the expected prediction error (EPE) in Eq. (33) and the D-optimality criterion in Eq. (39). This topic is classic (see, e.g., Boyd & Vandenberghe [9]).

B.1 Bayesian Linear Regression

In Bayesian linear regression (see, e.g., Murphy [49] and Gallager [25]), a learner is given n samples $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i$ represents the feature vector and y_i is the corresponding label. There exists a model $\boldsymbol{\beta} \in \mathbb{R}^d$ sampled from prior $\mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$ such that:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma^2\mathbf{I}\right), \tag{31}$$

where $\boldsymbol{y} = [y_i]_{i=1}^n \in \mathbb{R}^n$ is the label vector, $\boldsymbol{X} = \left[\boldsymbol{x}_i^\top\right]_{i=1}^n \in \mathbb{R}^{n\times d}$ is the design matrix, and $\boldsymbol{\epsilon} = [\epsilon_i]_{i=1}^n \in \mathbb{R}^n$ is the noise vector. The maximum a posteriori (MAP) estimate $\hat{\boldsymbol{\beta}}^{\texttt{MAP}} = \arg\max_{\boldsymbol{\beta}\in\mathbb{R}^d} \mathbf{P}(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{y})$ is:

$$\hat{\boldsymbol{\beta}}^{\texttt{MAP}} = \boldsymbol{\beta}_0 + \boldsymbol{\Sigma}_0\boldsymbol{X}^\top(\boldsymbol{X}\boldsymbol{\Sigma}_0\boldsymbol{X}^\top + \sigma^2\mathbf{I})^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0). \tag{32}$$

Expected Prediction Error (EPE). Consider the parameter estimate $\hat{\boldsymbol{\beta}}$ generated via Bayesian linear regression (Eq. (32)). Consider also a prediction at a test point $\boldsymbol{x} \in \mathbb{R}^d$. Then, the expected prediction error (EPE) at this unseen sample point $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \mathbb{R}$ under estimate $\hat{\boldsymbol{\beta}}$ is:

$$\mathbb{E}\left[\left(y - \boldsymbol{x}^\top\hat{\boldsymbol{\beta}}\right)^2\right] = \sigma^2 + \boldsymbol{x}^\top\boldsymbol{\Sigma}_{\boldsymbol{\xi}}\boldsymbol{x}, \tag{33}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \in \mathbb{S}_+^d$ is the covariance of the error $\boldsymbol{\xi} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, i.e.,

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}} \equiv \operatorname{cov}(\boldsymbol{\xi}) = \left(\frac{1}{\sigma^2}\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}. \tag{34}$$

The first term σ^2 of the EPE corresponds to irreducible observation noise, while the second term captures the contribution of estimation uncertainty and depends on both the test direction $\boldsymbol{x}$ and the geometry of $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$. Hence, the eigenvalues of the error covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$ capture the general variability of the EPE across all directions; we elaborate on this below.

EPE and Eigenvalues of the Covariance. Let the eigen-decomposition of the error covariance be

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \qquad \boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_d), \tag{35}$$

where $\{\lambda_i\}_{i=1}^d$ are the eigenvalues of $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}$ and $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d]$ contains the corresponding orthonormal eigenvectors.

The test point $\boldsymbol{x}$ can then be written as a linear combination of the eigenvectors, i.e.,

$$\boldsymbol{x} = \sum_{i=1}^{d} c_i \boldsymbol{u}_i, \tag{36}$$

with the coefficients $c_i = \boldsymbol{u}_i^\top \boldsymbol{x}$. Substituting this representation into Eq. (33) yields

$$\boldsymbol{x}^\top \boldsymbol{\Sigma}_{\boldsymbol{\xi}} \boldsymbol{x} = \sum_{i=1}^{d} \lambda_i c_i^2. \tag{37}$$

Equation (37) shows that the EPE depends on how strongly the prediction direction $\boldsymbol{x}$ aligns with the eigenvectors of the error covariance and on the magnitude of the corresponding eigenvalues. Large eigenvalues correspond to directions of high predictive uncertainty, whereas small eigenvalues indicate well-estimated directions. As a result, the eigenvalues of the estimation error covariance capture the variability of the EPE across orthogonal prediction directions.

B.2 Experimental Design

In experimental design [12, 9, 55], a learner determines which $n \in \mathbb{N}$ specific experiments to conduct in order to accurately estimate the linear model $\boldsymbol{\beta}$ with prior $\mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. Formally, there exists a set of candidate feature vectors $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_i\}_{i=1}^p \subset \mathbb{R}^d$ corresponding to p possible experiment settings. The overall process consists of three steps, illustrated in Fig. 4. First, in the *design selection* step, the learner determines a vector $\boldsymbol{n} = [n_i]_{i=1}^p \in \mathbb{N}^p$ with $\boldsymbol{n}^\top \mathbf{1} \leq n$, comprising the number of times each experiment is performed. The design matrix is thus:

$$\boldsymbol{X}(\boldsymbol{n}) = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_p^\top \\ \vdots \\ \boldsymbol{x}_p^\top \end{bmatrix} \begin{matrix} \left.\vphantom{\begin{matrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_1^\top \end{matrix}}\right\} n_1 \\ \vdots \\ \left.\vphantom{\begin{matrix} \boldsymbol{x}_p^\top \\ \vdots \\ \boldsymbol{x}_p^\top \end{matrix}}\right\} n_p \end{matrix} \in \mathbb{R}^{n \times d}. \tag{38}$$

In the *label collection* step, the learner collects the label vector $\boldsymbol{y} = [y_i]_{i=1}^n \in \mathbb{R}^n$ corresponding to this design matrix, governed by Eq. (31). Finally, in the *model estimation* step, the learner estimates the model using Bayesian linear regression on the collected labels, i.e., via Eq. (32).

In *D-optimal design* [55], the selection $\boldsymbol{n} \in \mathbb{N}^p$ is determined by minimizing the $\log\det$ of the posterior covariance, i.e.,

$$\log\det\left[\boldsymbol{\Sigma}_{\boldsymbol{\xi}}(\boldsymbol{n})\right] = -\log\det\left(\frac{1}{\sigma^2}\boldsymbol{B}(\boldsymbol{n}) + \boldsymbol{\Sigma}_0^{-1}\right), \tag{39}$$

where $\boldsymbol{B}(\boldsymbol{n}) \equiv \sum_{i=1}^p n_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \in \mathbb{R}^{d \times d}$. Minimizing this criterion is equivalent to minimizing the entropy of the posterior, as well as the EPE in Eq. (33) in all directions. For large n, a standard approach for computing the optimal design is by solving the following convex relaxation [9]:

$$\text{Max.: } G(\boldsymbol{\pi}; \sigma^2, \boldsymbol{\Sigma}_0) \equiv \log\det\left[\frac{n}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}) + \boldsymbol{\Sigma}_0^{-1}\right] \tag{40a}$$

$$\text{s.t.: } \boldsymbol{\pi}^\top \mathbf{1} \leq 1, \boldsymbol{\pi} \geq \mathbf{0}, \tag{40b}$$

where $\boldsymbol{\pi} = [\pi_i]_{i=1}^p \in [0,1]^p$ is a *fractional policy* and $\boldsymbol{B}(\boldsymbol{\pi}) \equiv \sum_{i=1}^p \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top \in \mathbb{R}^{d \times d}$. The fractional solution can subsequently be rounded (see below), producing an approximate integral solution s.t. $\boldsymbol{n} \approx n\boldsymbol{\pi}$. As the objective G is concave in $\boldsymbol{\pi}$, Prob. (40) is a convex optimization problem that can be solved using standard techniques.

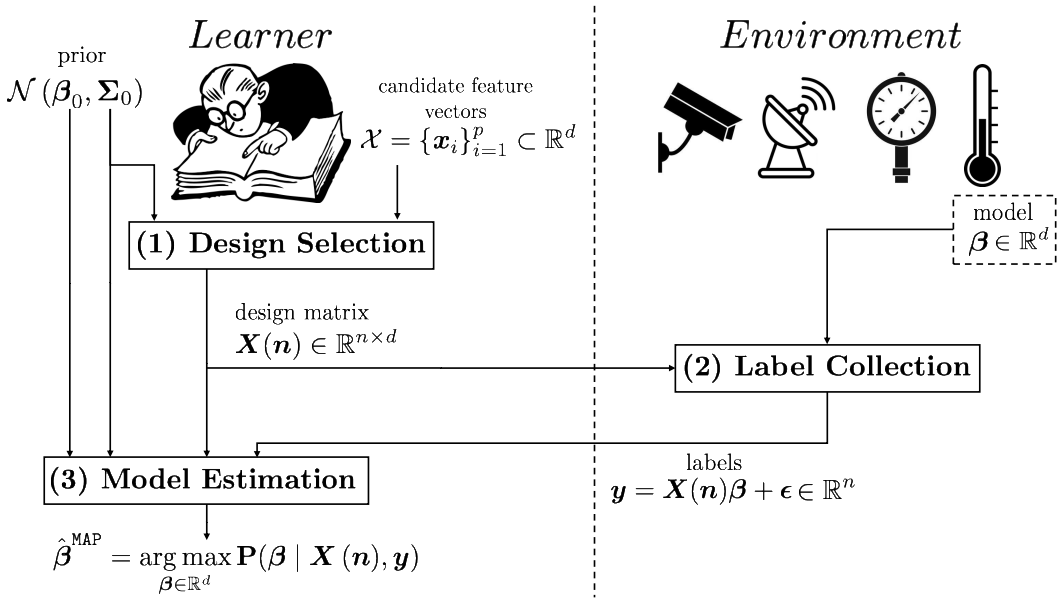


Figure 4: The classic experimental design setting. A learner having access to a prior distribution $\mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$ wishes to determine which experiments to conduct to estimate the (hidden) model $\boldsymbol{\beta}$. The entire process comprises three steps. In the *design selection* step, the learner selects n experiments to conduct, resulting in the design matrix $\boldsymbol{X}(\boldsymbol{n}) \in \mathbb{R}^{n \times d}$. In the *label collection* step, the corresponding experiments are conducted and labels $\boldsymbol{y} \in \mathbb{R}^n$ are revealed. In the *model estimation* step, the model is estimated via maximum a posteriori estimation. Only the covariance of the model prior is used in design selection, while the full prior is used in model estimation. In D-optimal design, the design selection is determined by selecting samples that minimize the entropy of the posterior.

Rounding Schemes. Classic experimental design (e.g., Boyd & Vandenberghe [9]) assumes rounding $\boldsymbol{\pi}^*$ to integer $\boldsymbol{n}$ is deterministic. For large budget n, the relaxation is tight: $n_i = \lfloor n\pi_i^* \rfloor$ satisfies the budget up to a unit deviation per coordinate, and the information matrix $\sum_i n_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is within $O(1/n)$ of the relaxed optimum. For moderate-to-large n, simple rounding is standard and sufficient. There are also more elaborate rounding schemes that preserve feasibility and objective quality. Singh and Xie [63] round via positively correlated distributions for D-design, and Nikolov et al. [50] use proportional volume sampling for A-design. Lau and Zhou [41] unify and improve both for D/A/E-design, including under additional knapsack constraints, and Lau et al. [40] provide deterministic rounding for D/A/E-design that satisfies cardinality constraints with provable approximation guarantees. Even randomized rounding procedures above produce an allocation with high-probability quality guarantees via martingale concentration bounds.

D-Optimality Criterion vs. EPE. Rather than optimizing the EPE for a single test direction, the D-optimality criterion aims to control prediction uncertainty uniformly across all directions. This is achieved by considering the determinant of the information matrix $\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^{-1}$. Since $\det(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^{-1}) = \prod_{i=1}^{d} \lambda_i^{-1}$, where λ_i are the eigenvalues of the error covariance in Eq. (35), it follows that

$$\log \det(\boldsymbol{\Sigma}_{\boldsymbol{\xi}}^{-1}) = -\sum_{i=1}^{d} \log \lambda_i. \tag{41}$$

Maximizing Eq. (41) thus penalizes large eigenvalues of the error covariance and promotes a balanced reduction of uncertainty across all eigen-directions. In this sense, the D-optimality criterion summarizes the variability of the expected prediction error over the entire input space, rather than along a single direction. By jointly controlling these eigenvalues through the log-determinant objective, D-optimality thus provides a principled scalar surrogate for minimizing predictive uncertainty when the test distribution is unknown or varies over time.

C Proof of Theorem 1

For $k \geq 1$, consider the following transformed model vector:

$$\tilde{\boldsymbol{\beta}}_k \equiv \boldsymbol{\beta}_k - \boldsymbol{\beta}_*. \tag{42}$$

Eq. (3) implies that:

$$\tilde{\boldsymbol{\beta}}_{k+1} = \rho\tilde{\boldsymbol{\beta}}_k + \boldsymbol{w}_k, \qquad \boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{w}}^2\mathbf{I}). \tag{43}$$

Consider also the following "differential" labels, defined as:

$$\tilde{\boldsymbol{y}}_k \equiv \boldsymbol{y}_k - \boldsymbol{X}_k\boldsymbol{\beta}_* \overset{\text{Eq. (60)}}{=} \boldsymbol{X}_k\boldsymbol{\beta}_k - \boldsymbol{X}_k\boldsymbol{\beta}_* + \boldsymbol{\epsilon}_k \overset{\text{Eq. (42)}}{=} \boldsymbol{X}_k\tilde{\boldsymbol{\beta}}_k + \boldsymbol{\epsilon}_k, \tag{44}$$

where $\boldsymbol{y}_k$ is the original vector of collected labels. Eqs. (43)–(44) thus define identical dynamics and observations, respectively, as Eqs. (59)–(60), with $\boldsymbol{A} = \rho\mathbf{I}$. Hence, the transformed model $\tilde{\boldsymbol{\beta}}_k$ can be estimated using the standard Kalman filter equations. Specifically, the *a priori* stage estimate of $\tilde{\boldsymbol{\beta}}_k$ is:

$$\hat{\tilde{\boldsymbol{\beta}}}_{k|k-1} = \rho\hat{\tilde{\boldsymbol{\beta}}}_{k-1|k-1}, \tag{45}$$

and the *a posteriori* estimate is:

$$\hat{\tilde{\boldsymbol{\beta}}}_{k|k} = \hat{\tilde{\boldsymbol{\beta}}}_{k|k-1} + \boldsymbol{K}_k\left(\tilde{\boldsymbol{y}}_k - \boldsymbol{X}_k\hat{\tilde{\boldsymbol{\beta}}}_{k|k-1}\right), \tag{46}$$

where the error covariance estimates and the Kalman gain are the same as in Eqs. (61b), (62b), and (63). Using Eq. (42), we re-write the *a priori* phase as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{k|k-1} - \boldsymbol{\beta}_* &= \rho\left(\hat{\boldsymbol{\beta}}_{k-1|k-1} - \boldsymbol{\beta}_*\right) \\ &= \rho\hat{\boldsymbol{\beta}}_{k-1|k-1} - \rho\boldsymbol{\beta}_*.\end{aligned} \tag{47}$$

Thus, we have

$$\hat{\boldsymbol{\beta}}_{k|k-1} = \rho\hat{\boldsymbol{\beta}}_{k-1|k-1} + (1-\rho)\,\boldsymbol{\beta}_*. \tag{48}$$

Furthermore, we can re-write the *a posteriori* phase as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{k|k} - \boldsymbol{\beta}_* &= \hat{\boldsymbol{\beta}}_{k|k-1} - \boldsymbol{\beta}_* + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\boldsymbol{\beta}_* - \boldsymbol{X}_k\left(\hat{\boldsymbol{\beta}}_{k|k-1} - \boldsymbol{\beta}_*\right)\right) \\ &= \hat{\boldsymbol{\beta}}_{k|k-1} - \boldsymbol{\beta}_* + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\boldsymbol{\beta}_* - \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{X}_k\boldsymbol{\beta}_*\right) \\ &= \hat{\boldsymbol{\beta}}_{k|k-1} - \boldsymbol{\beta}_* + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_{k|k-1}\right).\end{aligned} \tag{49}$$

Therefore, we get

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_{k|k-1}\right). \tag{50}$$

Thus, the parameter $\boldsymbol{\beta}_k$ in the system defined in Eq. (3) can be estimated using the regular Kalman filter equations in Eqs. (5)–(6). □

D Proof of Theorem 2

From Eq. (3), we can recursively write the model vector as:

$$\begin{aligned}\boldsymbol{\beta}_k &= \rho^{k-1}\boldsymbol{\beta}_1 + \left(1-\rho^{k-1}\right)\boldsymbol{\beta}_* + \sum_{j=1}^{k-1}\rho^{k-1-j}\boldsymbol{w}_j \\ &= \boldsymbol{\beta}_* + \rho^{k-1}\left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_*\right) + \sum_{j=1}^{k-1}\rho^{k-1-j}\boldsymbol{w}_j.\end{aligned} \tag{51}$$

Using the label collection in Eq. (4), the standard linear regression estimate is:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_k^{\texttt{LR}} &= \left(\boldsymbol{X}_k^\top \boldsymbol{X}_k\right)^{-1} \boldsymbol{X}_k^\top \boldsymbol{y}_k \\ &= \left(\boldsymbol{X}_k^\top \boldsymbol{X}_k\right)^{-1} \boldsymbol{X}_k^\top \left(\boldsymbol{X}_k \boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k\right) \\ &= \boldsymbol{\beta}_k + \left(\boldsymbol{X}_k^\top \boldsymbol{X}_k\right)^{-1} \boldsymbol{X}_k^\top \boldsymbol{\epsilon}_k \\ &= \boldsymbol{\beta}_* + \rho^{k-1}\left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_*\right) + \sum_{j=1}^{k-1} \rho^{k-1-j} \boldsymbol{w}_j + \left(\boldsymbol{X}_k^\top \boldsymbol{X}_k\right)^{-1} \boldsymbol{X}_k^\top \boldsymbol{\epsilon}_k\end{aligned} \tag{52}$$

Taking the sample mean of the estimator, we have:

$$\begin{aligned}\frac{1}{k}\sum_{k'=1}^{k} \hat{\boldsymbol{\beta}}_{k'}^{\texttt{LR}} = \boldsymbol{\beta}_* + \left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_*\right)\frac{1}{k}\sum_{k'=1}^{k} \rho^{k'-1} + \frac{1}{k}\sum_{k'=1}^{k}\sum_{j=1}^{k'-1} \rho^{k'-1-j}\boldsymbol{w}_j \\ + \frac{1}{k}\sum_{k'=1}^{k}\left(\boldsymbol{X}_{k'}^\top \boldsymbol{X}_{k'}\right)^{-1} \boldsymbol{X}_{k'}^\top \boldsymbol{\epsilon}_{k'}\end{aligned} \tag{53}$$

Consider the first summation term as

$$\frac{1}{k}\sum_{k'=1}^{k} \rho^{k'-1} = \frac{1-\rho^k}{k(1-\rho)}. \tag{54}$$

Thus the (deterministic) term

$$\left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_*\right)\frac{1}{k}\sum_{k'=1}^{k} \rho^{k'-1} = \left(\boldsymbol{\beta}_1 - \boldsymbol{\beta}_*\right)\frac{1-\rho^k}{k(1-\rho)} \tag{55}$$

converges to $\mathbf{0}$ as $k \to \infty$. Next, consider the drift noise term:

$$\frac{1}{k}\sum_{k'=1}^{k}\sum_{j=1}^{k'-1} \rho^{k'-1-j}\boldsymbol{w}_j = \frac{1}{k}\sum_{j=1}^{k-1} \boldsymbol{w}_j \sum_{k'=j+1}^{k} \rho^{k'-1-j} \tag{56}$$

$$= \frac{1}{k}\sum_{j=1}^{k-1} \boldsymbol{w}_j \frac{1-\rho^{k-j}}{1-\rho}. \tag{57}$$

Quantities $\boldsymbol{w}_j \frac{1-\rho^{k-j}}{1-\rho}$ are zero-mean random variables with bounded variance, as $\frac{1-\rho^{k-j}}{1-\rho} < \frac{1}{1-\rho}$. Hence, the drift noise term in Eq. (56) is the sum of independent, zero-mean, non-identically distributed random variables with bounded variance. It thus satisfies the Kolmogorov criterion for the strong law of large numbers (see Feller [22], p. 269) and thus converges w.p. 1 to zero. Convergence in expectation is trivial (as variables $\boldsymbol{w}_j$ are zero-mean).

Lastly, our assumptions on the boundedness of eigenvalues of $(\boldsymbol{X}_{k'}^\top \boldsymbol{X}_{k'})^{-1}$ imply that random variables $(\boldsymbol{X}_{k'}^\top \boldsymbol{X}_{k'})^{-1}\boldsymbol{X}_{k'}^\top \boldsymbol{\epsilon}_{k'}$ are also independent, zero-mean, and have bounded variance, so the term

$$\frac{1}{k}\sum_{k'=1}^{k}\left(\boldsymbol{X}_{k'}^\top \boldsymbol{X}_{k'}\right)^{-1} \boldsymbol{X}_{k'}^\top \boldsymbol{\epsilon}_{k'} \tag{58}$$

converges as $k \to \infty$ to $\mathbf{0}$, both w.p. 1 and in expectation, for the same reasons as above. □

E Optimal Myopic/One-Step Lookahead Policies under General Linear Dynamics

In this appendix, we provide an optimal myopic/one-step lookahead policy for general linear dynamics. These dynamics, as described by Eq. (59), generalize the dynamics in Eq. (3), assuming a general

linear evolution of drift through matrix $\boldsymbol{A}$. We show that we can design an optimal myopic/one-step lookahead policy for such general dynamics; the optimal myopic policy when the model evolves via Eq. (3) follows as a consequence (see App. E.1.2).

We note that, if matrix $\boldsymbol{A}$ is unknown, it can be estimated via traditional dynamical system identification techniques (see, e.g., [46]).

E.1 System Model

We consider a dynamic setting in which time progresses in stages $k = 1, 2, \ldots$, and the model vector $\boldsymbol{\beta}$ is subject to *drift* across stages, as illustrated in Fig. 1. Let $\{\boldsymbol{\beta}_k\}_{k=1}^{\infty} \subseteq \mathbb{R}^d$ denote the models at different stages. The first model $\boldsymbol{\beta}_1$ is sampled from a normal prior with mean $\boldsymbol{\beta}_0 \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{d\times d}$, i.e., $\boldsymbol{\beta}_1 \sim \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$, while, for $k \geq 1$, $\boldsymbol{\beta}_{k+1}$ evolves from $\boldsymbol{\beta}_k$ as follows:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{A}\boldsymbol{\beta}_k + \boldsymbol{w}_k, \qquad \boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\boldsymbol{w}}^2 \mathbf{I}), \tag{59}$$

where $\boldsymbol{A}$ is a known time-invariant state transition matrix, and $\boldsymbol{w}_k \in \mathbb{R}^d$ is the drift noise that is i.i.d. across stages. At each stage, the learner interacts with the environment via the three steps of the standard experimental design setting as presented in App. B.2. However, the process is repeated at each stage k, and both the design selection and model estimation steps are modified; we describe each of these steps in detail below.

Design Selection and Label Collection. Given a candidate set of feature vectors $\boldsymbol{\mathcal{X}}_k \subset \mathbb{R}^d$ with $|\boldsymbol{\mathcal{X}}_k| = p_k \in \mathbb{N}$, a learner selects which experiments to conduct at each stage k by creating a vector $\boldsymbol{n}_k = [n_{ik}]_{i=1}^{p_k} \in \mathbb{N}^{p_k}$ with $\boldsymbol{n}_k^\top \mathbf{1} = n_k$ at each respective stage. Under selection vector $\boldsymbol{n}_k$, the experiments to be conducted correspond to the design matrix $\boldsymbol{X}_k \equiv \boldsymbol{X}(\boldsymbol{n}_k)$, where $\boldsymbol{X}(\cdot)$ is as in Eq. (38).

After determining the design matrix, the learner collects labels, obtaining:

$$\boldsymbol{y}_k = \boldsymbol{X}_k\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\boldsymbol{0}, \sigma^2\mathbf{I}), \tag{60}$$

where $\boldsymbol{y}_k = [y_{ik}]_{i=1}^{n_k} \in \mathbb{R}^{n_k}$, $\boldsymbol{X}_k \in \mathbb{R}^{n_k\times d}$ are the vector of collected labels and the corresponding design matrix, respectively, and $\boldsymbol{\epsilon}_k \in \mathbb{R}^{n_k}$ is the (i.i.d.) measurement noise. Note that the initial state $\boldsymbol{\beta}_1$, the random drift vectors at each stage $\boldsymbol{w}_k$, and $\boldsymbol{\epsilon}_k$ are all assumed to be mutually independent.

Model Estimation. Finally, the learner estimates the model $\boldsymbol{\beta}_k$ at stage k *given the entire sequence of samples* $\{(\boldsymbol{X}_{k'}, \boldsymbol{y}_{k'})\}_{k'=1}^{k}$ *up to stage* k. Thus, departing from the standard experimental design setting, for each $k \geq 1$, *the MAP estimate of* $\boldsymbol{\beta}_k$ *now corresponds to a Kalman filter estimation* (see App. A for a review of Kalman filters). In particular, the model estimate at stage k can be adapted from the estimate at stage $k-1$, through a series of recursive equations. We use the common Kalman filter notation $(\cdot)_{k|k'}$ to denote a quantity estimated at stage k, having access to samples up to stage $k' \leq k$. Then, at stage k, before seeing the samples $(\boldsymbol{X}_k, \boldsymbol{y}_k)$, the *a priori* model estimate and its covariance of the model $\boldsymbol{\beta}_k$ at stage k are [25, 8]:

$$\hat{\boldsymbol{\beta}}_{k|k-1} = \boldsymbol{A}\hat{\boldsymbol{\beta}}_{k-1|k-1}, \tag{61a}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} = \boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k-1|k-1}}\boldsymbol{A}^\top + \sigma_{\boldsymbol{w}}^2\mathbf{I}. \tag{61b}$$

After the learner obtains the samples (i.e., the evidence) at stage k, the *a posteriori* estimate at stage k and its covariance are given by [8]:

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_{k|k-1}\right), \tag{62a}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\mathbf{I} - \boldsymbol{K}_k\boldsymbol{X}_k\right)\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}, \tag{62b}$$

where $\boldsymbol{K}_k \in \mathbb{R}^{d\times n_k}$ is the Kalman gain [8]:

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{X}_k^\top\left(\boldsymbol{X}_k\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{X}_k^\top + \sigma^2\mathbf{I}\right)^{-1}, \tag{63}$$

and initial estimates $(\cdot)_{1|0}$ are given by the prior.

Eqs. (61a)–(63) establish a recursion through which the learner can compute the model posterior mean (Eq. (62a)) and covariance (Eq. (62b)) at the conclusion of each stage; the former ($\hat{\boldsymbol{\beta}}_{k|k}$) is also the MAP estimate at stage k. Interestingly, no data retention is required in this setting: the full information obtained from past observations is encapsulated in the model posterior, which becomes a prior for the next stage.

E.1.1 Optimal Stage-wise Experimental Design

Having described our system model, we turn our attention to the question of how the learner is to perform the experimental design selection (i.e., how should $\boldsymbol{n}_k$ be picked at every stage). We first focus on a *greedy/myopic* setting: in this setting, the learner optimizes the design selection at stage k in a stage-wise fashion (i.e., one stage at a time), having access to the history of past dynamics through the model posterior at stage $k-1$.

Our first (straightforward) result is that a D-optimal fractional stage-wise design policy can again be obtained by solving an appropriate convex optimization problem.

Theorem 7. *The greedy D-optimal fractional policy at stage k can be computed as a solution to the following relaxed optimization problem:*

$$\textit{Maximize:} \quad G\left(\boldsymbol{\pi}_k; \sigma^2, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\right) \equiv \log\det\left[\frac{n_k}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}_k) + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right], \tag{64a}$$

$$\textit{s.t.:} \quad \boldsymbol{\pi}_k^\top \mathbf{1} \leq 1, \boldsymbol{\pi}_k \geq \mathbf{0}, \tag{64b}$$

where $\boldsymbol{\pi}_k \in [0,1]^{p_k}$ is the fractional policy at stage $k \in \mathbb{N}$, $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^{p_k} \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$, and $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}} \in \mathbb{R}^{d\times d}$ is the inverse of the a priori covariance given by Eq. (61b).

Proof. Using the Woodbury matrix identity [26], the *a posteriori* covariance in Eq. (62b) can be re-written as:

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\frac{1}{\sigma^2}\boldsymbol{X}_k^\top \boldsymbol{X}_k + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right)^{-1}, \tag{65}$$

where $\boldsymbol{X}_k \equiv \boldsymbol{X}(\boldsymbol{n}_k)$ is the design matrix at stage k and $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}$ is the inverse of the a priori covariance. Thus, given all decisions made by the learner so far, the optimal stage-wise design decision $\boldsymbol{n}_k$ can be found by maximizing the D-optimality criterion, given by:

$$\log\det\left[\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k}}(\boldsymbol{n}_k)\right] = \log\det\left[\frac{1}{\sigma^2}\boldsymbol{X}(\boldsymbol{n}_k)^\top \boldsymbol{X}(\boldsymbol{n}_k) + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right]. \tag{66}$$

The theorem thus follows by considering the relaxed version of this objective. □

Note that, given past decisions prior to stage k, $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}$ can be computed recursively, and is thus part of the input to Prob. (64), while the design is fully determined by $\boldsymbol{n}_k \approx n_k \boldsymbol{\pi}_k$. Finally, Prob. (64) has the same form as the classic D-optimal experimental design (cf. Prob. (40)), and can again be solved via standard convex optimization techniques [9].

E.1.2 Proof of Theorem 3

We conclude this section by determining the optimal myopic/one-step lookahead policies under the model drift described in Eq. (3). Observe that the *a posteriori* covariance (Eq. (62b)) of the model estimation under the general linear dynamics is exactly the same as the one (Eq. (5)) under the model drift in Eq. (3), from Thm. 1. Thus, the optimal stage-wise design for the dynamics in Eq. (3) can be found similarly by maximizing the D-optimality criterion, given by (66).

An immediate corollary is that Thm. 7 readily extends to this scenario: we can compute the optimal stage-wise policy by solving an instance of Prob. (64). Note that determining the optimal selection policy *does not require knowledge of $\boldsymbol{\beta}_*$*: Prob. (64) receives candidate samples $\boldsymbol{\mathcal{X}}$ and covariance $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}$ as input; the latter (by Eqs. (63), (5) and (6)) *does not depend on $\boldsymbol{\beta}_*$*.

E.1.3 Persistent Excitation

Provided that the set of candidate experiments $\boldsymbol{\mathcal{X}}_k \subset \mathbb{R}^d$ contains d vectors in general position, persistent excitation can be guaranteed by mixing in a small amount of exploration in the policy. This can be done by policies of the form:

$$\boldsymbol{\pi}^{\varepsilon} = (1-\varepsilon)\boldsymbol{\pi} + \varepsilon\boldsymbol{\mu} \tag{67}$$

with an exploration distribution $\boldsymbol{\mu}$ satisfying $\boldsymbol{B}(\boldsymbol{\mu}) \succ \mathbf{0}$. An alternative is to add the convex constraint $\boldsymbol{B}(\boldsymbol{\pi}) \succeq \delta\mathbf{I}$ to Probs. (9) and (13). As $\boldsymbol{B}(\boldsymbol{\pi})$ is affine in $\boldsymbol{\pi}$, in both approaches, the convexity of the corresponding optimization problems (Prob. (9) in the stage-wise case, and Prob. (13) in the steady-state case) is preserved.

Note that this is robust to rounding since $\boldsymbol{B}(\cdot)$ is affine and, hence, Lipschitz. In particular, for $R \geq \max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}_k} \|\boldsymbol{x}\|_2$, a rounded design $\widehat{\boldsymbol{\pi}}$ with $\|\widehat{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\varepsilon}\|_1 \leq \eta_\pi$ satisfies $\boldsymbol{B}(\widehat{\boldsymbol{\pi}}) \succeq (\delta - R^2\eta_\pi)\mathbf{I} \succ \mathbf{0}$ whenever $\delta > R^2\eta_\pi$. Hence, adding an exploration term implies that $\operatorname{rank}(\boldsymbol{X}_k) = d$ even under rounding, when the rounding error η_π is small enough.

F Proof of Theorem 4

The Markovian experimental design system denoted in Eq. (3) reaches a steady state as the system parameters satisfy the asymptotic conditions that are required to observe convergence in the system. Following Def. 1, the system parameters are defined as $\boldsymbol{\Sigma}_{\boldsymbol{w}_k} = \sigma_{\boldsymbol{w}}^2\mathbf{I}, \boldsymbol{\Sigma}_{\boldsymbol{\epsilon}_k} = \sigma^2\mathbf{I}$, where the state transition matrix is defined as $\boldsymbol{A} = \rho\mathbf{I}$. Additionally, assuming a stationary data collection policy in steady state, $\boldsymbol{X}(\boldsymbol{n})$ does not change over the stages. Therefore, the setting is time invariant. Following Thm. 6, the setting is asymptotically stable as $|\lambda_i(\boldsymbol{A})| < 1$. Then, these settings yield a stationary covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ and Kalman gain $\overline{\boldsymbol{K}}$.

Generalizing beyond the scope of Eq. (3), although the main setting in Eq. (59) is not necessarily asymptotically stable, i.e., $|\lambda_i(\boldsymbol{A})| < 1$ need not hold for all i, we still observe stability in covariance and Kalman gain as the detectability and stabilizability of $\left(\boldsymbol{A}, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ and $(\boldsymbol{A}, \boldsymbol{G})$, for any $\boldsymbol{G}$ with $\boldsymbol{G}\boldsymbol{G}^\top = \boldsymbol{\Sigma}_{\boldsymbol{w}_k}$, are satisfied, respectively. Following Def. 3, the detectability of $\left(\boldsymbol{A}, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ follows from the stabilizability of $\left(\boldsymbol{A}^\top, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ by duality. Having assumed $\boldsymbol{X}(\boldsymbol{n}) \in \mathbb{R}^{n\times d}$ to be rank d, $\left(\boldsymbol{A}^\top, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ is stabilizable if the matrix $\boldsymbol{H}$ in Eq. (27) is denoted as:

$$\boldsymbol{H} = -\left(\boldsymbol{X}(\boldsymbol{n})^\top\right)^\dagger \boldsymbol{A}^\top, \tag{68}$$

where $\left(\boldsymbol{X}(\boldsymbol{n})^\top\right)^\dagger$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{X}(\boldsymbol{n})^\top$. Substituting $\boldsymbol{H}$, the equation becomes:

$$\boldsymbol{A}^\top + \boldsymbol{X}(\boldsymbol{n})^\top\boldsymbol{H} = \boldsymbol{A}^\top - \boldsymbol{X}(\boldsymbol{n})^\top\left(\boldsymbol{X}(\boldsymbol{n})^\top\right)^\dagger \boldsymbol{A}^\top. \tag{69}$$

Since $\boldsymbol{X}(\boldsymbol{n})$ has rank d, the $d \times n$ matrix $\boldsymbol{X}(\boldsymbol{n})^\top$ has full row rank, so the projection term satisfies $\boldsymbol{X}(\boldsymbol{n})^\top\left(\boldsymbol{X}(\boldsymbol{n})^\top\right)^\dagger = \mathbf{I}$. Hence $\boldsymbol{A}^\top + \boldsymbol{X}(\boldsymbol{n})^\top\boldsymbol{H} = \boldsymbol{A}^\top - \boldsymbol{A}^\top = \mathbf{0}$, all of whose eigenvalues lie (trivially) strictly within the unit disk. Thus, Eq. (27) is always satisfied, i.e., $\left(\boldsymbol{A}^\top, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ is stabilizable, which by duality proves that $\left(\boldsymbol{A}, \boldsymbol{X}(\boldsymbol{n})^\top\right)$ is detectable. Moreover, following Def. 2, the stabilizability of $(\boldsymbol{A}, \boldsymbol{G})$ begins by constructing the matrix $\boldsymbol{G}$ as:

$$\boldsymbol{G} = \sigma_{\boldsymbol{w}}\mathbf{I}, \tag{70}$$

since $\boldsymbol{G}\boldsymbol{G}^\top = \sigma_{\boldsymbol{w}}^2\mathbf{I} = \boldsymbol{\Sigma}_{\boldsymbol{w}_k}$. As $\boldsymbol{G} = \sigma_{\boldsymbol{w}}\mathbf{I}$ is full rank where $\sigma_{\boldsymbol{w}} > 0$, $(\boldsymbol{A}, \boldsymbol{G})$ is stabilizable if the matrix $\boldsymbol{H}$ in Eq. (27) is denoted as:

$$\boldsymbol{H} = -\frac{1}{\sigma_{\boldsymbol{w}}}\boldsymbol{A}. \tag{71}$$

Substituting $\boldsymbol{H}$, the equation becomes:

$$\boldsymbol{A} + \boldsymbol{G}\boldsymbol{H} = \boldsymbol{A} + \sigma_{\boldsymbol{w}}\mathbf{I}\left(-\frac{1}{\sigma_{\boldsymbol{w}}}\boldsymbol{A}\right) = \mathbf{0}, \tag{72}$$

whose eigenvalues lie strictly within the unit disk. Thus, Eq. (27) is always satisfied, i.e., $(\boldsymbol{A}, \boldsymbol{G})$ is stabilizable. □

G Proof of Lemma 1

Consider the DARE-like equation:

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\tilde{\sigma}^2}\boldsymbol{B} + \left(\rho^2\boldsymbol{\Sigma} + \sigma_{\boldsymbol{w}}^2\mathbf{I}\right)^{-1}, \tag{73}$$

where $\tilde{\sigma} \equiv \frac{\sigma}{\sqrt{n}}$ and $\boldsymbol{B} \in \mathbb{S}_+^d$ is a PSD matrix. Let $\lambda_i, \boldsymbol{e}_i$ be the eigenvalues and eigenvectors of PSD matrix $\boldsymbol{B}$. We postulate that Eq. (15) is correct, and the eigen-decomposition of the covariance is:

$$\boldsymbol{\Sigma} = \sum_{i=1}^{d} \lambda_i' \boldsymbol{e}_i \boldsymbol{e}_i^\top. \tag{74}$$

We then show that this satisfies the DARE-like equation (73); the lemma thus follows from the uniqueness of the eigen-decomposition of a PSD matrix.

Assuming Eq. (74) holds, we get the following eigen-decompositions:

$$\left(\rho^2\boldsymbol{\Sigma} + \sigma_{\boldsymbol{w}}^2\mathbf{I}\right)^{-1} = \sum_{i=1}^{d} \left(\rho^2\lambda_i' + \sigma_{\boldsymbol{w}}^2\right)^{-1} \boldsymbol{e}_i \boldsymbol{e}_i^\top, \tag{75}$$

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^{d} \lambda_i'^{-1} \boldsymbol{e}_i \boldsymbol{e}_i^\top, \tag{76}$$

and the term involving $\boldsymbol{B}$ satisfies

$$\frac{1}{\tilde{\sigma}^2}\boldsymbol{B} = \frac{1}{\tilde{\sigma}^2} \sum_i \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top. \tag{77}$$

Re-arranging terms, we re-write Eq. (73) as:

$$\left(\rho^2\boldsymbol{\Sigma} + \sigma_{\boldsymbol{w}}^2\mathbf{I}\right)^{-1} - \boldsymbol{\Sigma}^{-1} + \frac{1}{\tilde{\sigma}^2}\boldsymbol{B} = \mathbf{0}. \tag{78}$$

Re-writing this using Eqs. (75)–(77), we get:

$$\begin{aligned}
\left(\rho^2\boldsymbol{\Sigma} + \sigma_{\boldsymbol{w}}^2\mathbf{I}\right)^{-1} - \boldsymbol{\Sigma}^{-1} + \frac{1}{\tilde{\sigma}^2}\boldsymbol{B} &= \sum_{i=1}^{d} \left(\rho^2\lambda_i' + \sigma_{\boldsymbol{w}}^2\right)^{-1} \boldsymbol{e}_i \boldsymbol{e}_i^\top - \sum_{i=1}^{d} \lambda_i'^{-1} \boldsymbol{e}_i \boldsymbol{e}_i^\top + \frac{1}{\tilde{\sigma}^2} \sum_{i=1}^{d} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top \\
&= \sum_{i=1}^{d} \frac{\lambda_i' - \rho^2\lambda_i' - \sigma_{\boldsymbol{w}}^2}{\rho^2\lambda_i'^2 + \sigma_{\boldsymbol{w}}^2\lambda_i'} \boldsymbol{e}_i \boldsymbol{e}_i^\top + \frac{1}{\tilde{\sigma}^2} \sum_{i=1}^{d} \lambda_i \boldsymbol{e}_i \boldsymbol{e}_i^\top \\
&= \sum_{i=1}^{d} \frac{\tilde{\sigma}^2\lambda_i' - \rho^2\tilde{\sigma}^2\lambda_i' - \sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2 + \rho^2\lambda_i\lambda_i'^2 + \sigma_{\boldsymbol{w}}^2\lambda_i\lambda_i'}{\rho^2\tilde{\sigma}^2\lambda_i'^2 + \sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2\lambda_i'} \boldsymbol{e}_i \boldsymbol{e}_i^\top \\
&= \mathbf{0}.
\end{aligned} \tag{79}$$

The last equation is equivalent to:

$$\tilde{\sigma}^2\lambda_i' - \rho^2\tilde{\sigma}^2\lambda_i' - \sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2 + \rho^2\lambda_i\lambda_i'^2 + \sigma_{\boldsymbol{w}}^2\lambda_i\lambda_i' = 0, \quad i = 1, \ldots, d. \tag{80}$$

Re-arranging terms, we get the quadratic equations w.r.t. λ_i':

$$\lambda_i'^2\rho^2\lambda_i + \lambda_i'\left(\tilde{\sigma}^2\left(1 - \rho^2\right) + \sigma_{\boldsymbol{w}}^2\lambda_i\right) - \sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2 = 0. \tag{81}$$

In order to solve the quadratic equation, we find Δ as:

$$\Delta = \left(\tilde{\sigma}^2\left(1 - \rho^2\right) + \sigma_{\boldsymbol{w}}^2\lambda_i\right)^2 + 4\rho^2\lambda_i\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2. \tag{82}$$

Observe that $\Delta \geq 0$ as the eigenvalues of PSD matrix $\boldsymbol{B}$ are always non-negative, i.e., $\lambda_i \geq 0$. Moreover, we are interested in positive solutions, as the covariance matrix is also PSD. Therefore, taking the positive root of the two distinct real roots, we write the eigenvalues λ_i' of covariance $\boldsymbol{\Sigma}$ as:

$$\lambda_i' = \frac{\sqrt{\left(\tilde{\sigma}^2\left(1-\rho^2\right)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)^2+4\lambda_i\rho^2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2}-\left(\tilde{\sigma}^2\left(1-\rho^2\right)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)}{2\rho^2\lambda_i}. \tag{83}$$

Rationalizing this expression by multiplying numerator and denominator by the conjugate $\sqrt{\Delta}+\left(\tilde{\sigma}^2(1-\rho^2)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)$, and using $\Delta-\left(\tilde{\sigma}^2(1-\rho^2)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)^2 = 4\lambda_i\rho^2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2$ gives the equivalent form

$$\lambda_i' = \frac{2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2}{\sqrt{\left(\tilde{\sigma}^2(1-\rho^2)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)^2+4\lambda_i\rho^2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2}+\left(\tilde{\sigma}^2(1-\rho^2)+\sigma_{\boldsymbol{w}}^2\lambda_i\right)}, \tag{84}$$

which is Eq. (15) (recall $\tilde{\sigma}^2 = \sigma^2/n$). The lemma follows by the uniqueness of the eigen-decomposition of $\boldsymbol{\Sigma} \in \mathbb{S}_+^d$. Note that Eq. (15) is well-defined for every $\lambda_i \geq 0$: at $\lambda_i = 0$ it equals $\sigma_{\boldsymbol{w}}^2/(1-\rho^2) > 0$. □

H Proof of Theorem 5

We begin by giving an outline of the proof. Observe that:

$$G\left(\boldsymbol{\pi}\right) = \log\det\left[\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\right] = \sum_{i=1}^{d}\log g\left(\lambda_i\left(\boldsymbol{\pi}\right)\right), \tag{85}$$

where $g:\mathbb{R}_+\to\mathbb{R}_+$ is the map from $\lambda_i \mapsto \lambda_i'^{-1}$ and λ_i' is given by Eq. (15), and $\lambda_i(\boldsymbol{\pi})$, $i=1,\ldots,d$ are the eigenvalues of $\boldsymbol{B}(\boldsymbol{\pi})$. The proof proceeds as follows:

1. We first show that the map $g:\mathbb{R}_+\to\mathbb{R}_+$ is non-decreasing and concave.
2. As a result, the map $\boldsymbol{\lambda}\mapsto\sum_{i=1}^d\log g(\lambda_i)$ is Schur-concave (see Definition 4).
3. This implies that function $\Phi:\mathbb{S}_+^d\to\mathbb{R}$, defined as
$$\Phi(\boldsymbol{A}) = \textstyle\sum_{i=1}^d\log g(\lambda_i(\boldsymbol{A})), \tag{86}$$
is concave over the positive semi-definite cone $\mathbb{S}_+^d$; moreover, as g is non-decreasing, Φ is non-decreasing in the Loewner order.
4. The theorem then follows from the fact that $G(\boldsymbol{\pi}) = \Phi(\boldsymbol{B}(\boldsymbol{\pi}))$, and $\boldsymbol{B}$ is affine and non-decreasing (in the Loewner order) in $\boldsymbol{\pi}$.

We prove each of these four constituent steps below.

Concavity of g. Our first step is to show that g is non-decreasing and concave (see Lemma 2). Let $\lambda_i, \boldsymbol{e}_i$ be the eigenvalues and eigenvectors of the PSD matrix $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^p\pi_i\boldsymbol{x}_i\boldsymbol{x}_i^\top$, and λ_i' (given by Eq. (15)) the eigenvalues of the steady-state covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$; correspondingly $g(\lambda_i) = 1/\lambda_i'$ are the eigenvalues of $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}$. We can thus re-write the objective in Eq. (13a) as:

$$G\left(\boldsymbol{\pi}\right) \equiv \log\det\left[\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\right] \tag{87}$$

$$= \log\left(\prod_{i=1}^{d} g\left(\lambda_i\left(\boldsymbol{\pi}\right)\right)\right) \tag{88}$$

$$= \sum_{i=1}^{d}\log g\left(\lambda_i\left(\boldsymbol{\pi}\right)\right), \tag{89}$$

where the function $g:\mathbb{R}_+\to\mathbb{R}_+$ is

$$g\left(\lambda\right) = 1/\lambda'(\lambda), \tag{90}$$

that defines the eigenvalues of the inverse covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})$ inside the determinant of the objective.

Lemma 2. *Function $g : \mathbb{R}_+ \to \mathbb{R}_+$ in Eq.* (90) *that is defined as:*

$$g(\lambda) = 1/\lambda'(\lambda), \tag{91}$$

is non-decreasing and concave.

Proof. Using the form of λ' in Eq. (15) gives

$$g(\lambda) = \frac{\sqrt{Q(\lambda)} + L(\lambda)}{2\sigma_{\boldsymbol{w}}^2 \tilde{\sigma}^2}, \tag{92}$$

where $L(\lambda) = \tilde{\sigma}^2(1-\rho^2) + \sigma_{\boldsymbol{w}}^2 \lambda$ and $Q(\lambda) = L(\lambda)^2 + 4\lambda\rho^2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2$. Here L is affine, and $Q(\lambda) = \sigma_{\boldsymbol{w}}^4\lambda^2 + 2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2(1+\rho^2)\lambda + \tilde{\sigma}^4(1-\rho^2)^2$ is a quadratic with positive leading coefficient, strictly positive for $\lambda \geq 0$, with discriminant

$$\Delta_Q = \left(2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2(1+\rho^2)\right)^2 - 4\sigma_{\boldsymbol{w}}^4\tilde{\sigma}^4(1-\rho^2)^2 = 16\rho^2\sigma_{\boldsymbol{w}}^4\tilde{\sigma}^4 > 0. \tag{93}$$

Therefore $(\sqrt{Q})''(\lambda) = -\Delta_Q/\left(4Q(\lambda)^{3/2}\right) < 0$, i.e., $\sqrt{Q}$ is concave; being a positive multiple of the sum of the concave $\sqrt{Q}$ and the affine L, g is concave. Finally $L'(\lambda) = \sigma_{\boldsymbol{w}}^2 > 0$ and $Q'(\lambda) = 2\sigma_{\boldsymbol{w}}^4\lambda + 2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2(1+\rho^2) > 0$, so $g'(\lambda) = \left(Q'(\lambda)/(2\sqrt{Q(\lambda)}) + \sigma_{\boldsymbol{w}}^2\right)/(2\sigma_{\boldsymbol{w}}^2\tilde{\sigma}^2) > 0$; hence g is non-decreasing. Both properties hold for every $\lambda \geq 0$, including the boundary $\lambda = 0$. □

Schur-concavity of map $\boldsymbol{\lambda} \mapsto \sum_{i=1}^d \log g(\lambda_i)$. A consequence of Lemma 2 is that the map $\boldsymbol{\lambda} \mapsto \sum_{i=1}^d \log g(\lambda_i)$ is Schur-concave; to show this, we introduce first the definition of Schur-convex and Schur-concave functions:

Definition 4. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we say that $\boldsymbol{x}$ majorizes $\boldsymbol{y}$, i.e., $\boldsymbol{x} \succ \boldsymbol{y}$, if and only if $\boldsymbol{y}$ is in the convex hull of all vectors obtained by permuting the coordinates of $\boldsymbol{x}$. By the Birkhoff–von Neumann theorem [47], this is equivalent to the following statement: there exists a doubly stochastic matrix $\boldsymbol{D}$ such that $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}$. A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is *Schur-convex* if $\boldsymbol{x} \succ \boldsymbol{y}$ implies [47]:

$$\phi(\boldsymbol{x}) \geq \phi(\boldsymbol{y}), \tag{94}$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$. Similarly, a function ϕ is *Schur-concave* if $\boldsymbol{x} \succ \boldsymbol{y}$ implies $\phi(\boldsymbol{x}) \leq \phi(\boldsymbol{y})$.

Armed with this definition, we have the following result:

Lemma 3. *Function $\phi : \mathbb{R}^d \to \mathbb{R}$ defined as*

$$\phi(\boldsymbol{\lambda}) = \sum_{i=1}^d \log g(\lambda_i) \tag{95}$$

is Schur-concave.

Proof. Using convexity composition rules (see Boyd & Vandenberghe [9]), $\log g$ is concave as $\log$ is concave and non-decreasing and g is concave from Lemma 2. We make use of the following lemma:

Lemma 4 (Marshall et al. [47])**.** *A function $\phi : \mathbb{R}^n \to \mathbb{R}$ that is defined as*

$$\phi(\boldsymbol{x}) = \sum_{i=1}^n g(x_i) \tag{96}$$

is Schur-concave *if the function $g : \mathbb{R} \to \mathbb{R}$ is concave.*

Following this lemma, $\phi = \sum_{i=1}^d \log g(\lambda_i)$ is Schur-concave as $\log g$ is concave. □

Concavity and monotonicity of Φ over the PSD cone. Next, we define a function $\Phi : \mathbb{S}_+^d \to \mathbb{R}$ as

$$\Phi(\boldsymbol{A}) = \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{A})), \tag{97}$$

where $\boldsymbol{A} \in \mathbb{S}_+^d$. Recall that $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^{p} \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$. Thus, we can write the objective function as

$$G(\boldsymbol{\pi}) = \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{\pi})) = \Phi(\boldsymbol{B}(\boldsymbol{\pi})). \tag{98}$$

Next, we establish the monotonicity and concavity properties of map Φ.

Lemma 5. *Function $\Phi : \mathbb{S}_+^d \to \mathbb{R}$ in Eq.* (97) *that is defined as*

$$\Phi(\boldsymbol{A}) = \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{A})), \tag{99}$$

is non-decreasing in the Loewner order (i.e., $\Phi(\boldsymbol{A}) \geq \Phi(\boldsymbol{B})$ whenever $\boldsymbol{A} \succeq \boldsymbol{B}$, that is, $\boldsymbol{A} - \boldsymbol{B} \in \mathbb{S}_+^d$) and concave over the positive semi-definite cone $\mathbb{S}_+^d$.

Proof. Since function ϕ, defined in Eq. (95), is Schur-concave by Lemma 3, for all $\boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \mathbb{R}^d$ with $\boldsymbol{\lambda} \prec \boldsymbol{\lambda}'$, we have:

$$\boldsymbol{\lambda} \prec \boldsymbol{\lambda}' \quad \Rightarrow \quad \phi(\boldsymbol{\lambda}) \geq \phi(\boldsymbol{\lambda}') \quad \Rightarrow \quad \sum_{i=1}^{d} \log g(\lambda_i) \geq \sum_{i=1}^{d} \log g(\lambda_i'), \tag{100}$$

where the majorization '$\succ$' is defined in Definition 4. Moreover, Fan's theorem [47, 20] states that, for all matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}_+^d$, we have:

$$\boldsymbol{\lambda}(\boldsymbol{A} + \boldsymbol{B}) \prec \boldsymbol{\lambda}(\boldsymbol{A}) + \boldsymbol{\lambda}(\boldsymbol{B}). \tag{101}$$

Therefore, the following inequality is satisfied

$$\sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{A} + \boldsymbol{B})) \geq \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{A}) + \lambda_i(\boldsymbol{B})) \tag{102}$$

for all matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}_+^d$. Thus, the function Φ is concave as:

$$\begin{aligned}
\Phi(\theta \boldsymbol{A} + (1-\theta)\boldsymbol{B}) &= \sum_{i=1}^{d} \log g(\lambda_i(\theta \boldsymbol{A} + (1-\theta)\boldsymbol{B})) \\
&\overset{(102)}{\geq} \sum_{i=1}^{d} \log g(\theta \lambda_i(\boldsymbol{A}) + (1-\theta)\lambda_i(\boldsymbol{B})) \\
&\overset{\text{Lemma 2}}{\geq} \sum_{i=1}^{d} (\theta \log g(\lambda_i(\boldsymbol{A})) + (1-\theta) \log g(\lambda_i(\boldsymbol{B}))) \\
&= \theta \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{A})) + (1-\theta) \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{B})) \\
&= \theta \Phi(\boldsymbol{A}) + (1-\theta)\Phi(\boldsymbol{B}),
\end{aligned} \tag{103}$$

where the second inequality follows from the concavity of g. Finally, Φ is non-decreasing in the Loewner order: for $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}_+^d$ with $\boldsymbol{A} \succeq \boldsymbol{B}$, write $\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{P}$, where $\boldsymbol{P} = \boldsymbol{A} - \boldsymbol{B} \in \mathbb{S}_+^d$. Then

$$\Phi(\boldsymbol{A}) = \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{B} + \boldsymbol{P})) \overset{(102)}{\geq} \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{B}) + \lambda_i(\boldsymbol{P})) \geq \sum_{i=1}^{d} \log g(\lambda_i(\boldsymbol{B})) = \Phi(\boldsymbol{B}), \tag{104}$$

where the second inequality holds as $\lambda_i(\boldsymbol{P}) \geq 0$ and $\log g$ is non-decreasing (g is positive and non-decreasing by Lemma 2). □

Concavity and monotonicity of G. Lemma 5 allows us to prove that the steady-state objective function $G : \mathbb{R}^p \to \mathbb{R}$ in Eq. (98) is indeed non-decreasing and concave. To see this, recall from Eq. (98) that we can write the objective as $G(\boldsymbol{\pi}) = \Phi(\boldsymbol{B}(\boldsymbol{\pi}))$, where $\Phi : \mathbb{S}_+^d \to \mathbb{R}$ is defined in Eq. (97) and $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^p \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$. Observe that $\boldsymbol{B} : \mathbb{R}^p \to \mathbb{S}_+^d$ is affine in $\boldsymbol{\pi}$. Since, by Lemma 5, Φ is concave, function G is concave as a composition of a concave and an affine function. Moreover, $\boldsymbol{B}$ is non-decreasing in the Loewner order: for $\mathbf{0} \leq \boldsymbol{\pi} \leq \boldsymbol{\pi}'$ (componentwise), $\boldsymbol{B}(\boldsymbol{\pi}') - \boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^p (\pi_i' - \pi_i)\, \boldsymbol{x}_i \boldsymbol{x}_i^\top \succeq \mathbf{0}$, as each $\boldsymbol{x}_i \boldsymbol{x}_i^\top \succeq \mathbf{0}$. Since, by Lemma 5, Φ is non-decreasing in the Loewner order, $G(\boldsymbol{\pi}') = \Phi(\boldsymbol{B}(\boldsymbol{\pi}')) \geq \Phi(\boldsymbol{B}(\boldsymbol{\pi})) = G(\boldsymbol{\pi})$, i.e., G is non-decreasing. □

I Extension to A-Optimality

The D-optimality criterion we adopt already tracks the test RMSE closely (the expected prediction error decomposes as $\sigma^2 + \boldsymbol{x}^\top \boldsymbol{\Sigma}_{\boldsymbol{\xi}} \boldsymbol{x}$, cf. Eq. (33)). Nonetheless, the same machinery shows that the *A-optimality* criterion, minimizing the trace (i.e., the total posterior variance) of the steady-state covariance, also yields a convex program.

Using the closed-form spectrum of Lemma 1, the A-optimal steady-state objective is

$$\operatorname{tr} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi}) \;=\; \sum_{i=1}^{d} \lambda_i'(\boldsymbol{\pi}) \;=\; \sum_{i=1}^{d} \phi(\lambda_i(\boldsymbol{B}(\boldsymbol{\pi}))), \qquad \phi(\lambda) \;:=\; \lambda'(\lambda) \;=\; \frac{1}{g(\lambda)}, \tag{105}$$

where λ_i' are the eigenvalues of $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ and $g = 1/\lambda'$ is the positive, non-decreasing, concave map of Lemma 2.

Lemma 6. *The map $\phi(\lambda) = 1/g(\lambda)$ is non-increasing and convex on $\mathbb{R}_+$.*

Proof. Since $g > 0$ is non-decreasing, $\phi = 1/g$ is non-increasing. Since $g > 0$ is concave, $\phi = 1/g$ is convex: $\phi'' = \big(2(g')^2 - g\, g''\big)/g^3 \geq 0$, as $g'' \leq 0$ and $g > 0$. □

Because ϕ is convex, the separable symmetric map $\boldsymbol{\lambda} \mapsto \sum_{i=1}^d \phi(\lambda_i)$ is *Schur-convex* [47], the exact mirror of the Schur-concavity of $\boldsymbol{\lambda} \mapsto \sum_i \log g(\lambda_i)$ used for D-optimality (Definition 4). Repeating the argument in the proof of Theorem 5 (Fan's theorem together with the affinity of $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_i \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$ in $\boldsymbol{\pi}$), the objective $\boldsymbol{\pi} \mapsto \operatorname{tr} \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi}) = \sum_{i=1}^d \phi(\lambda_i(\boldsymbol{B}(\boldsymbol{\pi})))$ is convex over the PSD cone. Hence *minimizing* the A-optimal objective is a convex program, exactly mirroring the D-optimal case of App. H. Moreover, the implicit-gradient machinery of App. J carries over verbatim, with the log-determinant gradient $\partial \log\det$ replaced by the trace gradient $\partial \operatorname{tr}$.

J Solving Prob. (13)

In our numerical evaluations, we find the optimal steady-state policy by solving Prob. (13). As established in Thm. 5, this is a convex optimization problem and thus admits efficient standard techniques; in practice, we implemented a solver using the Frank–Wolfe algorithm [24].

Both in this solver and in any optimization approach that uses a gradient evaluation (e.g., projected gradient descent, primal-dual methods, etc.), a key challenge lies in computing the gradient of the objective function in Eq. (13a). This requires differentiating the steady-state covariance defined implicitly through the DARE system of equations (Eq. (12)). We first describe how this gradient is computed efficiently via the implicit function theorem [37], then explain how it is used within the Frank–Wolfe iterations, and evaluate the computational complexity of each iteration within the algorithm.

J.1 Estimating the Gradient of the Steady-State Objective (13a) via the Implicit Function Theorem

Recall that the steady-state objective is written as $G(\boldsymbol{\pi}) = \log\det\big(\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\big)$. Define the inverse steady-state covariance

$$\overline{\boldsymbol{Q}}(\boldsymbol{\pi}) \triangleq \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi}) \in \mathbb{S}_{++}^d, \tag{106}$$

so that $G(\boldsymbol{\pi}) = \log\det(\overline{\boldsymbol{Q}}(\boldsymbol{\pi}))$. The matrix $\overline{\boldsymbol{Q}}(\boldsymbol{\pi})$ is not available in closed form. Instead, it is defined implicitly as the unique solution to a Riccati-type fixed-point equation. To formalize this dependence, we define the mapping

$$F(\boldsymbol{Q}, \boldsymbol{\pi}) := \boldsymbol{Q} - \left(\rho^2 \boldsymbol{Q}^{-1} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1} - \frac{n}{\sigma^2} \sum_{i=1}^{p} \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top. \tag{107}$$

This formulation corresponds to the information-form steady-state Riccati equation and is algebraically equivalent to Eq. (12). We then note that the inverse steady-state covariance $\overline{\boldsymbol{Q}}(\boldsymbol{\pi})$ satisfies $F(\overline{\boldsymbol{Q}}(\boldsymbol{\pi}), \boldsymbol{\pi}) = \mathbf{0}$. For a fixed allocation $\boldsymbol{\pi}$, we compute the matrix $\overline{\boldsymbol{Q}}(\boldsymbol{\pi})$ numerically via fixed-point iteration applied to the above equation until convergence. Since $F(\boldsymbol{Q}, \boldsymbol{\pi})$ is continuously differentiable and its Jacobian with respect to $\boldsymbol{Q}$ is invertible at the steady state, the implicit function theorem [37] guarantees that $\overline{\boldsymbol{Q}}(\boldsymbol{\pi})$ is differentiable with respect to $\boldsymbol{\pi}$. Moreover, for each coordinate π_j,

$$\frac{\partial \overline{\boldsymbol{Q}}}{\partial \pi_j} = -\left(\frac{\partial F}{\partial \boldsymbol{Q}}\Big|_{\overline{\boldsymbol{Q}}}\right)^{-1} \frac{\partial F}{\partial \pi_j}, \tag{108}$$

where $\frac{\partial F}{\partial \boldsymbol{Q}} : \mathbb{S}^d \to \mathbb{S}^d$ denotes a linear operator acting on the space $\mathbb{S}^d$ of symmetric $d \times d$ matrices. The partial derivative with respect to π_j is given explicitly by $\frac{\partial F}{\partial \pi_j} = -\frac{n}{\sigma^2} \boldsymbol{x}_j \boldsymbol{x}_j^\top \in \mathbb{S}^d$. On the other hand, the Jacobian $\frac{\partial F}{\partial \boldsymbol{Q}} : \mathbb{S}^d \to \mathbb{S}^d$ admits the following Fréchet representation: for any perturbation $\boldsymbol{\Delta} \in \mathbb{S}^d$,

$$\boldsymbol{\Delta} \;\mapsto\; \boldsymbol{\Delta} - \rho^2 \boldsymbol{M}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{\Delta}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{M}, \qquad \boldsymbol{M} = \left(\rho^2 \overline{\boldsymbol{Q}}^{-1} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1} \in \mathbb{S}^d_{++}. \tag{109}$$

Consequently, evaluating $\left(\frac{\partial F}{\partial \boldsymbol{Q}}\right)^{-1} \boldsymbol{C} \in \mathbb{S}^d$ for a given $\boldsymbol{C} \in \mathbb{S}^d$ amounts to solving the two-sided Sylvester (Lyapunov-type) linear matrix equation

$$\boldsymbol{\Delta} - \rho^2 \boldsymbol{M}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{\Delta}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{M} = \boldsymbol{C}, \qquad \boldsymbol{\Delta} \in \mathbb{S}^d, \tag{110}$$

which admits a unique solution under standard detectability conditions. Each sensitivity matrix $\frac{\partial \overline{\boldsymbol{Q}}}{\partial \pi_j} \in \mathbb{S}^d$ is obtained by solving this equation numerically. Finally, applying the chain rule yields the gradient of the steady-state objective:

$$\frac{\partial G(\boldsymbol{\pi})}{\partial \pi_j} = \operatorname{tr}\left(\overline{\boldsymbol{Q}}^{-1} \frac{\partial \overline{\boldsymbol{Q}}}{\partial \pi_j}\right) = \operatorname{tr}\left(\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \frac{\partial \overline{\boldsymbol{Q}}}{\partial \pi_j}\right) \in \mathbb{R}. \tag{111}$$

This procedure provides an exact first-order characterization of $\nabla G(\boldsymbol{\pi}) \in \mathbb{R}^p$, which is subsequently used by the Frank–Wolfe solver.

Efficient gradient via a single adjoint solve. Evaluating Eq. (108) separately for each coordinate solves the Lyapunov-type equation above p times, costing $O(p\, d^3)$. This reduces to a *single* solve. The operator $\boldsymbol{\Delta} \mapsto \boldsymbol{\Delta} - \rho^2 \boldsymbol{M}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{\Delta}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{M}$ is self-adjoint with respect to the trace inner product (as $\boldsymbol{M}, \overline{\boldsymbol{Q}}^{-1} \in \mathbb{S}^d$), so by the chain rule and $\frac{\partial F}{\partial \pi_j} = -\frac{n}{\sigma^2} \boldsymbol{x}_j \boldsymbol{x}_j^\top$,

$$\frac{\partial G(\boldsymbol{\pi})}{\partial \pi_j} = \operatorname{tr}\left(\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\, \frac{\partial \overline{\boldsymbol{Q}}}{\partial \pi_j}\right) = -\operatorname{tr}\left(\boldsymbol{\Lambda}\, \frac{\partial F}{\partial \pi_j}\right) = \frac{n}{\sigma^2}\, \boldsymbol{x}_j^\top \boldsymbol{\Lambda}\, \boldsymbol{x}_j, \tag{112}$$

where the adjoint matrix $\boldsymbol{\Lambda} \in \mathbb{S}^d$ is the unique solution of the *single* Lyapunov-type equation

$$\boldsymbol{\Lambda} - \rho^2 \boldsymbol{M}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{\Lambda}\, \overline{\boldsymbol{Q}}^{-1} \boldsymbol{M} = \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} = \overline{\boldsymbol{Q}}^{-1}. \tag{113}$$

The entire gradient $\nabla G(\boldsymbol{\pi}) \in \mathbb{R}^p$ is thus obtained from one $d \times d$ solve for $\boldsymbol{\Lambda}$ ($O(d^3)$) followed by p quadratic forms $\boldsymbol{x}_j^\top \boldsymbol{\Lambda} \boldsymbol{x}_j$ ($O(p\, d^2)$), reducing the per-iteration cost from $O(p\, d^3)$ to $O(d^3 + p\, d^2)$.

J.2 Solving Prob. (13) Using the Frank–Wolfe Algorithm

Using the gradient characterization above, we solve Prob. (13) via the Frank–Wolfe algorithm [24] over the probability simplex

$$\Delta_p := \left\{ \boldsymbol{\pi} \in \mathbb{R}^p_{\geq 0} : \sum_{i=1}^{p} \pi_i = 1 \right\}. \tag{114}$$

The algorithm is initialized from a feasible point $\boldsymbol{\pi}_0 \in \Delta_p$, which we take to be the uniform allocation in our experiments. At each Frank–Wolfe iteration t, the objective is linearized around the current iterate $\boldsymbol{\pi}_t$, yielding the linear subproblem

$$\boldsymbol{s}_t \in \arg\max_{\boldsymbol{s} \in \Delta_p} \nabla G(\boldsymbol{\pi}_t)^\top \boldsymbol{s}. \tag{115}$$

Since the feasible set is a simplex, this subproblem admits a closed-form solution: $\boldsymbol{s}_t$ is the unit vector corresponding to the largest component of $\nabla G(\boldsymbol{\pi}_t) \in \mathbb{R}^p$. That is, the algorithm selects the coordinate $j^\star = \arg\max_j [\nabla G(\boldsymbol{\pi}_t)]_j$, and assigns all mass to that index. The iterate is then updated via the convex combination

$$\boldsymbol{\pi}_{t+1} = (1 - \gamma_t)\boldsymbol{\pi}_t + \gamma_t \boldsymbol{s}_t, \tag{116}$$

where the step size $\gamma_t \in [0, 1]$ is selected by performing a one-dimensional line search along the Frank–Wolfe direction $\boldsymbol{s}_t - \boldsymbol{\pi}_t$. In our implementation, we employ exact line search on the original objective $G(\boldsymbol{\pi})$, which we compute using the *minimize_scalar* method from *scipy.optimize* with box constraints $[0, 1]$.

We emphasize, however, that exact line search is not required for the convergence of the Frank–Wolfe algorithm. In practice, comparable performance can be achieved using inexact line search rules (e.g., backtracking or Armijo line search) or standard diminishing step-size schedules. The use of exact line search in our implementation is primarily for numerical stability and ease of implementation, rather than algorithmic necessity. The algorithm terminates when the Frank–Wolfe duality gap

$$g_t = \nabla G(\boldsymbol{\pi}_t)^\top (\boldsymbol{s}_t - \boldsymbol{\pi}_t) \tag{117}$$

falls below a tolerance, indicating approximate first-order stationarity.

J.3 Computational Complexity

Each Frank–Wolfe iteration is dominated by the cost of evaluating the gradient $\nabla G(\boldsymbol{\pi}_t) \in \mathbb{R}^p$. Computing the steady-state covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi}_t)$ requires solving a Riccati-type fixed-point equation, which in practice is obtained via fixed-point iteration and involves repeated matrix inversions of dimension d, incurring a cost of $O(d^3)$ per iteration.

Given $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi}_t)$, each coordinate of the gradient requires solving a Lyapunov/Sylvester-type linear matrix equation of dimension $d \times d$, yielding a total cost of $O(p\, d^3)$ per gradient evaluation. Using the adjoint formulation of Eqs. (112)–(113), this drops to $O(d^3 + p\, d^2)$: a single $d \times d$ solve for the adjoint $\boldsymbol{\Lambda}$ followed by p quadratic forms. The remaining $O(d^3)$ term is the $d \times d$ covariance/DARE operation and becomes the bottleneck only for very high-dimensional (e.g., deep) models; there, standard low-rank or diagonal covariance approximations from Bayesian deep learning keep the *effective* d small. By contrast, the Frank–Wolfe linear subproblem admits a closed-form solution and amounts to identifying the largest component of the gradient, which costs $O(p)$.

When exact line search is used, each step-size selection requires multiple evaluations of the steady-state objective along a single direction. Since each such evaluation involves recomputing the steady-state covariance, the total cost of exact line search introduces an additional multiplicative factor. In practice, this overhead can be avoided by using inexact line search rules or predefined step-size schedules, which preserve convergence guarantees while substantially reducing computational cost.

Overall, the computational complexity of each Frank–Wolfe iteration is dominated by gradient evaluation, while the linear subproblem and update steps incur negligible additional overhead.

K Non-linear Setting

In this appendix, we provide the full details of the non-linear generalization summarized in Sec. 3.4. Throughout, we consider the system defined by the drift dynamics in Eq. (3) and the non-linear observation model in Eq. (2):

$$\boldsymbol{\beta}_{k+1} = \rho\boldsymbol{\beta}_k + (1-\rho)\boldsymbol{\beta}_* + \boldsymbol{w}_k, \quad \boldsymbol{w}_k \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{w}}^2\mathbf{I}), \tag{118}$$

$$\boldsymbol{y}_k = h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}), \tag{119}$$

where $h : \mathbb{R}^{n_k \times d} \times \mathbb{R}^{d_{\boldsymbol{\beta}}} \to \mathbb{R}^{n_k}$ is a function parameterized by $\boldsymbol{\beta}_k \in \mathbb{R}^{d_{\boldsymbol{\beta}}}$.

K.1 Model Estimation in the Non-linear Setting

As the measurement function h is non-linear, the standard Kalman filter no longer applies. We instead employ the *extended Kalman filter (EKF)* [8], which linearizes h around the current model estimate at each stage. Formally, the EKF linearizes h through its first-order Taylor approximation, so that

$$h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) \approx h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{H}_k(\boldsymbol{\beta}_k - \hat{\boldsymbol{\beta}}_{k|k-1}), \tag{120}$$

where

$$\boldsymbol{H}_k = \frac{\partial h(\boldsymbol{X}_k, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{k|k-1}} \in \mathbb{R}^{n_k \times d_{\boldsymbol{\beta}}} \tag{121}$$

is the Jacobian matrix. In practice, this leads to the EKF adaptation equations that, compared to a standard Kalman filter, involve replacing the design matrix $\boldsymbol{X}_k$ with the Jacobian $\boldsymbol{H}_k$. Our approach naturally extends to this setting, as discussed below.

Known $\boldsymbol{\beta}_*$. We can again show that, under our dynamics as well as the approximation in Eq. (120), estimation reduces to an EKF when the mean fixed-point model $\boldsymbol{\beta}_*$ is known.

Theorem 8. *Consider the system evolving via Eqs.* (2)–(3)*. Assume that the mean fixed-point $\boldsymbol{\beta}_*$ is known. Then, the extended Kalman filter maintains a Gaussian approximation $\mathcal{N}(\hat{\boldsymbol{\beta}}_{k|k}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}})$ to $\boldsymbol{\beta}_k$, with the posterior mean and covariance:*

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{y}_k - h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1})\right), \tag{122a}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = (\mathbf{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\,\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}, \tag{122b}$$

respectively, where $\hat{\boldsymbol{\beta}}_{k|k-1}$, $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}$ are the a priori estimates at stage k given in Eq. (6)*, $\boldsymbol{K}_k$ is the Kalman gain:*

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{H}_k^\top\left(\boldsymbol{H}_k\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}\boldsymbol{H}_k^\top + \sigma^2\mathbf{I}\right)^{-1}, \tag{123}$$

and $\boldsymbol{H}_k$ is the Jacobian matrix:

$$\boldsymbol{H}_k = \frac{\partial h(\boldsymbol{X}_k, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}_{k|k-1}} \in \mathbb{R}^{n_k \times d_{\boldsymbol{\beta}}}. \tag{124}$$

Proof. The prediction step follows directly from the dynamics in Eq. (3), which is identical to the linear case (Eq. (6)). For the posterior, we linearize the observation model around $\hat{\boldsymbol{\beta}}_{k|k-1}$:

$$h(\boldsymbol{X}_k, \boldsymbol{\beta}_k) \approx h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{H}_k(\boldsymbol{\beta}_k - \hat{\boldsymbol{\beta}}_{k|k-1}). \tag{125}$$

Under this first-order approximation, the observation model becomes locally linear:

$$\boldsymbol{y}_k \approx h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{H}_k(\boldsymbol{\beta}_k - \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{\epsilon}_k. \tag{126}$$

Defining the *non-linear innovation* $\boldsymbol{\nu}_k \equiv \boldsymbol{y}_k - h(\boldsymbol{X}_k, \hat{\boldsymbol{\beta}}_{k|k-1})$ and applying the standard Kalman filter update with $\boldsymbol{H}_k$ in place of $\boldsymbol{X}_k$ yields Eqs. (122a)–(122b). The proof of Thm. 1 (App. C) carries over directly, since the transformation $\tilde{\boldsymbol{\beta}}_k = \boldsymbol{\beta}_k - \boldsymbol{\beta}_*$ yields the same structure with $\boldsymbol{A} = \rho\mathbf{I}$ in the state equation and $\boldsymbol{H}_k$ replacing $\boldsymbol{X}_k$ in the linearized observation equation. □

As in the linear setting, the covariance recursion does not depend on $\boldsymbol{\beta}_*$; we can compute the optimal myopic policy without knowing $\boldsymbol{\beta}_*$. We describe the optimal myopic policies for the non-linear setting in App. K.2.1.

Estimating $\boldsymbol{\beta}_*$. As in the linear setting, we need an estimate of the mean fixed-point $\boldsymbol{\beta}_*$. We obtain this via cumulative non-linear least squares:

$$\hat{\boldsymbol{\beta}}_* \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \sum_{t=1}^{k} \|\boldsymbol{y}_t - h(\boldsymbol{X}_t, \boldsymbol{\beta})\|_2^2 . \tag{127}$$

Unlike the linear case (Thm. 2), where the least-squares problem is convex and the estimator provably converges to $\boldsymbol{\beta}_*$, this non-linear estimation is generally non-convex, so the minimizer may not be unique. In practice, we observe that $\hat{\boldsymbol{\beta}}_*$ stabilizes reliably across all model architectures considered in our experiments (see also Fig. 6 in App. M). We use this estimate as the linearization point for the steady-state design selection below.

K.2 Optimal Design Selection in the Non-linear Setting

We now describe how to perform D-optimal design selection in the non-linear setting. The key idea is that, under the linearization (120), the Jacobian vectors play the role of the feature vectors in the linear setting: the optimization problems retain the same convex structure, with the Jacobian evaluated at an appropriate linearization point.

K.2.1 Optimal Myopic Design Selection

At each stage k, the EKF has computed the a priori estimate $\hat{\boldsymbol{\beta}}_{k|k-1}$ and covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}$ in Eq. (6). We linearize the observation model around $\hat{\boldsymbol{\beta}}_{k|k-1}$, which is the best available estimate before observing the data at stage k. For each candidate $\boldsymbol{x}_{ik} \in \boldsymbol{\mathcal{X}}_k$, define the Jacobian vector:

$$\boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{k|k-1}) = \left.\frac{\partial h(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}_{k|k-1}} \in \mathbb{R}^{d_{\boldsymbol{\beta}}}. \tag{128}$$

Applying the Woodbury matrix identity to the a posteriori covariance (122b) yields:

$$\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}} = \left(\frac{1}{\sigma^2}\boldsymbol{H}_k^\top \boldsymbol{H}_k + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right)^{-1}, \tag{129}$$

where, under design vector $\boldsymbol{n}_k$, we have $\boldsymbol{H}_k^\top \boldsymbol{H}_k = \sum_{i=1}^{p} n_{ik}\, \boldsymbol{g}_i \boldsymbol{g}_i^\top$. The D-optimal myopic design thus amounts to solving:

$$\text{Maximize:} \quad G\left(\boldsymbol{\pi}_k; \sigma^2, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}, \hat{\boldsymbol{\beta}}_{k|k-1}\right) \equiv \log\det\left[\frac{n_k}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}_k; \hat{\boldsymbol{\beta}}_{k|k-1}) + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right], \tag{130a}$$

$$\text{s.t.:} \quad \boldsymbol{\pi}_k^\top \mathbf{1} \le 1, \boldsymbol{\pi}_k \ge \mathbf{0}, \tag{130b}$$

where $\boldsymbol{B}(\boldsymbol{\pi}_k; \hat{\boldsymbol{\beta}}_{k|k-1}) = \sum_{i=1}^{p_k} \pi_{ik} \boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{k|k-1}) \boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_{k|k-1})^\top$ is the linearized Fisher information. Since $\boldsymbol{g}_i$ is fixed at the current estimate $\hat{\boldsymbol{\beta}}_{k|k-1}$, the objective is $\log\det$ of an affine function of $\boldsymbol{\pi}_k$ and is therefore concave. Prob. (130) is thus a convex optimization problem of the same form as the linear case (Prob. (9)), and can be solved using the same techniques. We stress again that, as in the linear case, the myopic policy *does not require knowledge of* $\boldsymbol{\beta}_*$.

K.2.2 Optimal Steady-State Selection

For the steady-state analysis, recall that under the dynamics (3), the model $\boldsymbol{\beta}_k$ oscillates around the mean fixed-point $\boldsymbol{\beta}_*$ in steady state. This motivates us to analyze the steady state using the following approximations: first, we linearize the observation model around $\hat{\boldsymbol{\beta}}_*$, the estimate of the mean fixed-point obtained from (127); second, we treat this linearization as fixed for the purpose of the steady-state analysis. Concretely, we assume that the Jacobian $\boldsymbol{H}_k \approx \boldsymbol{H} \equiv \partial h(\boldsymbol{X}(\boldsymbol{n}), \boldsymbol{\beta})/\partial\boldsymbol{\beta}\big|_{\hat{\boldsymbol{\beta}}_*}$ is approximately constant across stages. Under this assumption, the EKF covariance recursion becomes a time-invariant linear system, to which the standard convergence theory applies.

Convergence to Steady State. Under a stationary selection policy $\boldsymbol{n} \in \mathbb{N}^p$ and the linearizability assumption above, the EKF covariance recursion becomes time-invariant with state transition matrix $\boldsymbol{A} = \rho\mathbf{I}$, observation matrix $\boldsymbol{H}$, process noise $\sigma_{\boldsymbol{w}}^2\mathbf{I}$, and measurement noise $\sigma^2\mathbf{I}$. Assuming $\boldsymbol{H}$

has rank $d_{\boldsymbol{\beta}}$, Thm. 4 applies directly (with $\boldsymbol{H}$ replacing $\boldsymbol{X}(\boldsymbol{n})$): the pair $(\rho\mathbf{I}, \boldsymbol{H})$ is detectable and $(\rho\mathbf{I}, \sigma_{\boldsymbol{w}}\mathbf{I})$ is stabilizable, following from the same arguments as in App. F. Hence, the EKF covariance converges to $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \in \mathbb{R}^{d_{\boldsymbol{\beta}} \times d_{\boldsymbol{\beta}}}$ satisfying the DARE:

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1} = \frac{1}{\sigma^2}\boldsymbol{H}^\top \boldsymbol{H} + \left(\rho^2 \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1}, \tag{131}$$

and the Kalman gain converges to $\boldsymbol{K} = \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}} \boldsymbol{H}^\top (\boldsymbol{H}\, \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}\, \boldsymbol{H}^\top + \sigma^2 \mathbf{I})^{-1}$. Unlike the linear setting, where the covariance dynamics are fully decoupled from $\boldsymbol{\beta}_*$, the non-linear DARE (131) depends on $\boldsymbol{\beta}_*$ through the Jacobian $\boldsymbol{H}$. We use $\hat{\boldsymbol{\beta}}_*$ for two reasons. First, $\boldsymbol{\beta}_*$ is the point around which the model oscillates in steady state (cf. Eq. (3)), making it the most representative linearization point for the long-run behavior. Second, as $\hat{\boldsymbol{\beta}}_*$ stabilizes, the Jacobian $\boldsymbol{H}_k \to \boldsymbol{H} \equiv \partial h / \partial \boldsymbol{\beta}\big|_{\boldsymbol{\beta}_*}$, and the DARE solution converges accordingly.

Optimal Stationary Policy. Under the linearizability assumption, the steady-state design optimization takes the same form as in the linear case. For each candidate $\boldsymbol{x}_i$, define the Jacobian vector:

$$\boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_*) = \left.\frac{\partial h(\boldsymbol{x}_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}_*} \in \mathbb{R}^{d_{\boldsymbol{\beta}}}, \tag{132}$$

and the linearized Fisher information $\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*) = \sum_{i=1}^p \pi_i \boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_*) \boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_*)^\top$. The optimal stationary policy solves:

$$\text{Maximize:} \quad G\left(\boldsymbol{\pi}; \sigma^2, \sigma_{\boldsymbol{w}}^2, \hat{\boldsymbol{\beta}}_*\right) \equiv \log\det\left[\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi})\right], \tag{133a}$$

$$\text{s.t.:} \quad \boldsymbol{\pi}^\top \mathbf{1} \le 1, \boldsymbol{\pi} \ge \mathbf{0}, \tag{133b}$$

where $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi})$ solves the relaxed DARE:

$$\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}^{-1}(\boldsymbol{\pi}) = \frac{n}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*) + \left(\rho^2 \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi}) + \sigma_{\boldsymbol{w}}^2 \mathbf{I}\right)^{-1}. \tag{134}$$

The map $\boldsymbol{\pi} \mapsto \overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}(\boldsymbol{\pi})$ is defined implicitly through Eq. (134), precisely as in the linear case. The crucial observation is that, under the linearizability assumption, $\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*) = \sum_{i=1}^p \pi_i\, \boldsymbol{g}_i \boldsymbol{g}_i^\top$ has the same algebraic structure as $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^p \pi_i\, \boldsymbol{x}_i \boldsymbol{x}_i^\top$ in the linear setting. Thus Lemma 1 applies: the eigenvectors of $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ and $\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*)$ are shared, and the eigenvalues of $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ are given by Eq. (15) as a function of the eigenvalues of $\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*)$. This paves the way for establishing concavity:

Corollary 1 (Concavity of the Non-linear Steady-State Objective)**.** *Under the linearizability assumption, the objective $G : \mathbb{R}^p \to \mathbb{R}$ defined in* (133a) *is concave. Consequently, Prob.* (133) *is a convex optimization problem.*

Proof. Under the linearizability assumption, $\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*)$ is positive semi-definite and affine in $\boldsymbol{\pi}$, with the same structure as in the linear case. Since Lemma 1 holds for any positive semi-definite matrix $\boldsymbol{B}$, the eigenvalue characterization in Eq. (15) applies. The proof of Thm. 5 (App. H) then carries over verbatim: the map $g : \lambda_i \mapsto \lambda_i'^{-1}$ remains non-decreasing and concave, the function $\Phi(\boldsymbol{A}) = \sum_{i=1}^{d_{\boldsymbol{\beta}}} \log g(\lambda_i(\boldsymbol{A}))$ is Schur-concave, and $G(\boldsymbol{\pi}) = \Phi(\boldsymbol{B}(\boldsymbol{\pi}; \hat{\boldsymbol{\beta}}_*))$ inherits concavity from the affinity of $\boldsymbol{B}$ in $\boldsymbol{\pi}$. □

As a result, under the linearizability assumption, the non-linear steady-state design problem follows the same steps as Thm. 5 in the linear setting, with the Jacobian vectors $\boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_*)$ replacing the feature vectors $\{\boldsymbol{x}_i\}$. Consequently, Prob. (133) is a convex optimization problem that can be solved using the optimization approach described in App. J, namely, the Frank–Wolfe algorithm, coupled with implicit differentiation to compute the gradient of the DARE solution w.r.t. the policy $\boldsymbol{\pi}$.

L Experimental Setup

L.1 Datasets

We perform numerical evaluations over a synthetic dataset as well as over six publicly available datasets spanning environmental monitoring, finance, transportation, housing, and wireless networks,

summarized in Table 1. For the real datasets, to enable re-sampling during the label collection stage, we reveal the ground truth label with additive zero-mean Gaussian noise with variance 10^{-1}. For the real datasets, in both the linear and non-linear settings, the system parameters $(\rho, \sigma, \sigma_{\boldsymbol{w}})$ are estimated directly from data following the procedure described in App. L.3; the resulting values are summarized in Table 4. In all experiments across all datasets, we run every method 100 times, with a different seed each time, and report the mean and standard deviation of the metrics described in App. L.5.

L.1.1 Synthetic Dataset

We generate four synthetic datasets that are consistent with the dynamics in Eqs. (2)–(3), using different model functions h. We simulated four functions, namely a linear model, a ResNet model, a convolutional neural network (CNN), and a Multi-Layer Perceptron (MLP); the three non-linear architectures are described in App. L.2. Note that, in the linear case, $d_{\boldsymbol{\beta}} = d$. In all cases, the set of candidate experiments $\boldsymbol{\mathcal{X}}_k \subset \mathbb{R}^d$, i.e., the pool of feature vectors from which the learner selects, varies with time. We generate these in an i.i.d. fashion per stage by setting $d = 30$, $p = 40$, and $n = 50$, and sampling $\boldsymbol{x}_{ik} \in \mathbb{R}^d$ in $\boldsymbol{\mathcal{X}}_k$ from a normal distribution with zero mean and unit variance, i.e., $\boldsymbol{x}_{ik} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We generate the ground truth labels via Eq. (2) under a model $\boldsymbol{\beta}_k$ drifting according to Eq. (3). We set the parameters in the model dynamics (Eq. (3)) as $\rho = 0.9$, and the noise parameters to $\sigma = 0.9$ and $\sigma_{\boldsymbol{w}} = 0.9$ (see Table 4). The prior mean is randomly sampled as $\boldsymbol{\beta}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the prior covariance is constructed as $\boldsymbol{\Sigma}_0 = c\mathbf{I}$, where $c = 50$. The first stage model is sampled as $\boldsymbol{\beta}_1 \sim \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. Moreover, the mean fixed-point is randomly sampled as $\boldsymbol{\beta}_* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Beyond the aforementioned set of candidate feature vectors, we create another synthetic set of feature vectors $\boldsymbol{X}_k^{\text{test}} = \{\boldsymbol{x}_{ik}\}_{i=1}^{n/4} \subset \mathbb{R}^d$, coupled with labels $\boldsymbol{y}^{\text{test}} \in \mathbb{R}^{n/4}$ generated via Eq. (2). We use this test set to measure the predictive performance of our estimates (see also App. L.5). We run these synthetic experiments for a total of $T = 2000$ stages.

L.1.2 Beijing PM2.5 [14].

This dataset consists of hourly air quality measurements collected at a single monitoring station (the US Embassy in Beijing, China) from January 1, 2010 through December 31, 2014, spanning a total of 5 years (60 months). The regression task is to predict the PM2.5 concentration given the remaining air quality and meteorological measurements. Each sample corresponds to a single hourly measurement with $d = 12$ features that include date-time information (year, month, day, hour) and meteorological variables. The dataset contains 43824 samples in total, corresponding to roughly 8760 hourly observations per year over the 5-year span.

We pre-process the data by applying ordinal encoding to the categorical feature "combined wind direction". We handled the missing PM2.5 values (4.72% of the whole data) by imputation using linear interpolation. We group together daily measurements as feature vectors, resulting in 24-hour long stages with 24 samples each. Having a total of 43824 samples in the data, we end up with $T = 1826$ stages in this scenario. After utilizing the daily measurements, we exclude the date-time information from the data. For each stage, we randomly split 20% of the total measurements as a test set $(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{y}^{\text{test}})$, and the rest are used as the training set: their respective features are utilized as the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$. Thus, the number of experiments becomes $p = 19$ at each stage.

L.1.3 S&P 500 [51].

This dataset contains daily historical stock-price records for all companies listed on the S&P 500 index of the US public equity market, covering a 5-year window from approximately February 2013 through February 2018. The regression task is to predict the next-day closing price given the current day's trading features. Each sample corresponds to a single trading day for a single stock and includes the trading date, the stock's ticker name, and $d = 6$ daily price/volume features: open price, high price, low price, close price, adjusted close price, and trading volume. The dataset contains 619040 stock-day records in total across 505 S&P 500 constituents, corresponding to roughly 1260 trading days per stock over the 5-year span. We pre-process the data by selecting a representative subset of stocks and a window of approximately 14 months (62 trading weeks) from the original 5-year coverage, and computing daily returns and normalized volume. We group together weekly trading measurements, resulting in stages of approximately 5 trading days. Each stage pools all available stock measurements for that week. The selected window yields $T = 62$ stages. For each stage,



Figure 5: Illustration of the AT&T Downlink/Uplink dataset's measurement route (source: [13]). The trace corresponds to a cross-country drive from Boston, MA to Los Angeles, CA in August 2022, spanning approximately 4,500 km and traversing 13 US states. As the vehicle traverses different regions, cellular network conditions change due to varying base-station density, available technology (4G LTE vs. 5G low/mid/mmWave bands), and terrain, inducing geographic concept drift in the feature–throughput relationship.

we randomly split 20% of the measurements as a test set $(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{y}^{\text{test}})$, and the rest are used as the training set: their respective features are utilized as the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$. The number of candidate experiments is $p = 16$ at each stage.

L.1.4 Bike Sharing Demand [21].

This dataset contains hourly bike rental counts from the Capital Bikeshare system in Washington, D.C., USA, spanning two full years from January 1, 2011 through December 31, 2012 (24 months). The regression task is to predict the total number of bike rentals in a given hour given weather and calendar features. Each sample corresponds to a single hourly measurement with $d = 8$ features, including season, holiday indicator, working day indicator, weather situation, normalized temperature, normalized feeling temperature, normalized humidity, and normalized wind speed. The dataset contains 17379 hourly records in total, corresponding to roughly 8690 hourly observations per year over the 2-year span.

We pre-process the data by applying ordinal encoding to the categorical features (season, weather situation). We group together daily measurements as feature vectors, resulting in 24-hour long stages with 24 samples each. After grouping the 17379 hourly records, we obtain $T = 724$ stages. We exclude the date-time information from the feature set. For each stage, we randomly split 20% of the measurements as a test set $(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{y}^{\text{test}})$, and the rest are used as the training set: their respective features are utilized as the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$. The number of candidate experiments is $p = 19$ at each stage.

L.1.5 California Housing [52].

This dataset contains aggregated demographic and housing statistics from the 1990 US Decennial Census, covering 20640 census block groups distributed across the entire state of California (all 58 counties). The regression task is to predict the median house value for a block group given the remaining census-derived features. Each sample corresponds to a single block group with $d = 6$ features: median income, housing median age, average number of rooms, average number of bedrooms, block-group population, and average household size. The raw dataset additionally provides each block group's latitude and longitude, which support geographic ordering. The dataset contains 20640 samples in total, all collected from the 1990 decennial census, yielding an average of 356 block-group observations per California county.

Table 4: Dynamics parameter estimates $(\rho, \sigma, \sigma_{\boldsymbol{w}})$ fitted to each model architecture's residuals on each dataset (see App. L.3). For each real dataset, the underlined value in the ResNet/CNN/MLP columns marks the median across the three non-linear architectures per parameter; this median is the shared estimate used for all non-linear models on that dataset in our experiments.

Dataset	Linear			ResNet			CNN			MLP		
	ρ	σ	$\sigma_{\boldsymbol{w}}$	ρ	σ	$\sigma_{\boldsymbol{w}}$	ρ	σ	$\sigma_{\boldsymbol{w}}$	ρ	σ	$\sigma_{\boldsymbol{w}}$
Synthetic	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.90
Beijing PM2.5 [14]	0.92	80.29	0.98	0.95	134.48	0.03	0.93	131.68	0.02	0.92	135.93	0.02
S&P 500 [51]	0.66	0.78	0.22	0.96	0.75	0.01	0.91	0.78	0.02	0.91	0.89	0.02
Bike Sharing [21]	0.93	156.98	29.08	0.95	263.28	0.03	0.93	261.16	0.03	0.96	261.75	0.02
California Housing [52]	0.94	0.79	0.01	0.95	1.19	0.02	0.91	1.22	0.01	0.91	1.21	0.02
AT&T Downlink [13]	0.53	183.58	684.94	0.94	363.75	0.04	0.92	364.47	0.04	0.94	360.51	0.03
AT&T Uplink [13]	0.74	13.32	644.75	0.91	20.04	0.02	0.94	20.56	0.05	0.90	19.57	0.03

We induce a natural ordering by sorting block groups geographically (by latitude, north-to-south) and partitioning them into stages of 50 consecutive block groups each. This geographic ordering introduces the drift, as the relationship between housing features and price varies across regions (e.g., the income–price mapping differs substantially between the San Francisco Bay Area and the Central Valley). The dataset yields $T = 412$ stages. For each stage, we randomly split 20% of the block groups as a test set $(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{y}^{\text{test}})$, and the rest are used as the training set: their respective features are utilized as the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$. The number of candidate experiments is $p = 40$ at each stage.

L.1.6 AT&T Downlink and Uplink [13].

This multi-carrier cellular measurement dataset was collected during a cross-country driving campaign from Boston, MA to Los Angeles, CA in August 2022, traversing approximately 4,500 km across 13 US states (Massachusetts, New York, Pennsylvania, Ohio, Indiana, Illinois, Iowa, Nebraska, Wyoming, Colorado, Utah, Nevada, and California) over a multi-day trip and passing through major cities including Buffalo, Cleveland, Columbus, Indianapolis, Chicago, Des Moines, Omaha, Denver, Salt Lake City, and Las Vegas. The corresponding measurement route is illustrated in Fig. 5. Throughout the trip, three smartphones each connected to one of the three major US mobile operators (AT&T, Verizon, and T-Mobile) concurrently logged downlink/uplink throughput at 500 ms intervals together with per-sample signal-quality and cellular-technology indicators (RSRP, RSRQ, SINR, serving frequency band, and 4G/5G technology type). We extract two separate prediction tasks from the AT&T traces: predicting *downlink* throughput and predicting *uplink* throughput, yielding the AT&T Downlink and AT&T Uplink datasets, with approximately 6,700 and 6,560 raw measurement samples, respectively, collected over the course of the cross-country trip (roughly 500 samples per traversed state on average).

For the downlink task, each sample has $d = 12$ features comprising the cellular signal-quality and technology indicators listed above; for the uplink task, each sample has $d = 8$ features (the uplink direction does not expose a subset of the per-channel KPIs that are available on the downlink). We induce a stage structure by grouping measurements into road segments of approximately equal geographic length, with each segment constituting one stage; as the vehicle traverses regions with varying base station density, technology availability (4G LTE vs. 5G low/mid/mmWave bands), and terrain, the network conditions change substantially across stages, creating natural concept drift in the feature–throughput relationship. The downlink task spans $T = 335$ stages and the uplink task spans $T = 328$ stages. For each stage, we randomly split 20% of the measurements as a test set $(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{y}^{\text{test}})$, and the rest are used as the training set: their respective features are utilized as the candidate feature vectors $\boldsymbol{\mathcal{X}}_k$. The number of candidate experiments is $p = 16$ for both tasks at each stage.

L.2 Architectures

The three non-linear architectures are parameterized as follows. ResNet is a custom 1D residual network based on the ResNet architecture [27] with 5 convolutional layers, organized as an initial convolution of kernel size 3 from 1 to 2 channels, followed by two residual blocks with 2 and 3 channels, respectively, each containing two convolutions of kernel size 3 (the second block also

Table 5: Methods

Method	Description
STS-AE (ours)	Stationary optimal selection, Adaptive estimation
STS-AE-K (ours)	Stationary optimal selection, Adaptive estimation, Known $\boldsymbol{\beta}_*$
STG-AE (ours)	Stage-wise optimal selection, Adaptive estimation
STG-AE-K (ours)	Stage-wise optimal selection, Adaptive estimation, Known $\boldsymbol{\beta}_*$
STG-SE	Stage-wise optimal selection, Static estimation
STG-SGD	Stage-wise optimal selection, SGD estimation
ADM-AE	Adam optimized selection, Adaptive estimation
ADM-AE-K	Adam optimized selection, Adaptive estimation, Known $\boldsymbol{\beta}_*$
ADM-SE	Adam optimized selection, Static estimation
ADM-SGD	Adam optimized selection, SGD estimation
RND-AE	Random selection, Adaptive estimation
RND-AE-K	Random selection, Adaptive estimation, Known $\boldsymbol{\beta}_*$
RND-SE	Random selection, Static estimation
RND-SGD	Random selection, SGD estimation
GKS-AE	Greedy Kalman information-gain selection, Adaptive estimation
GKS-AE-K	Greedy Kalman information-gain selection, Adaptive estimation, Known $\boldsymbol{\beta}_*$
GKS-SE	Greedy Kalman information-gain selection, Static estimation
GKS-SGD	Greedy Kalman information-gain selection, SGD estimation
GKS-RLS	Greedy Kalman information-gain selection, Forgetting-factor RLS estimation
RND-RLS	Random selection, Forgetting-factor RLS estimation
ADM-RLS	Adam optimized selection, Forgetting-factor RLS estimation
STG-RLS	Stage-wise optimal selection, Forgetting-factor RLS estimation

uses a 1×1 convolutional shortcut, as its channel count changes), an adaptive average pooling layer reducing the spatial dimension to 1, and a final linear scalar readout. The CNN consists of two 1D convolutional layers with output channels $(3, 5)$ and kernel sizes $(3, 5)$, each followed by a $\tanh$ activation, and uses the same adaptive-pool-plus-linear-head readout as the ResNet. Since both convolutional architectures expect a sequence input but our datasets provide d-dimensional feature vectors $\boldsymbol{x} \in \mathbb{R}^d$, we feed each vector to the network by reshaping it into a single-channel 1D signal of length d before the convolutions are applied. The MLP, in contrast, takes the feature vector directly and consists of three hidden fully-connected layers of width 2 each with $\tanh$ activations followed by a linear scalar output. All three non-linear architectures use $\tanh$ activations and bias terms in every linear/convolutional layer; per-architecture parameter counts are reported in Table 1.

L.3 Parameter Estimation

Note that, in the real datasets, there is no ground truth dynamical model $\boldsymbol{\beta}_k$. As we need ρ, σ^2, and $\sigma_{\boldsymbol{w}}^2$ in our algorithms, we measure these directly from data as follows. In the linear setting, we produce an estimate $\hat{\boldsymbol{\beta}}_*$ of the mean fixed-point $\boldsymbol{\beta}_*$ by performing linear regression on the burn-in prefix only (the first $\lfloor 0.1\,T \rfloor$ stages), and estimate a per-stage $\hat{\boldsymbol{\beta}}_k$ through the linear regression using the p training samples. In the non-linear setting, we instead fit the non-linear model h on the same burn-in prefix via non-linear least squares:

$$\hat{\boldsymbol{\beta}}_* = \underset{\boldsymbol{\beta} \in \mathbb{R}^{d_{\boldsymbol{\beta}}}}{\arg\min} \sum_{k=1}^{K} \|\boldsymbol{y}_k - h(\boldsymbol{X}_k, \boldsymbol{\beta})\|_2^2 \,, \tag{135}$$

and estimate a per-stage $\hat{\boldsymbol{\beta}}_k$ by fitting the non-linear model on each stage's p training samples:

$$\hat{\boldsymbol{\beta}}_k = \underset{\boldsymbol{\beta} \in \mathbb{R}^{d_{\boldsymbol{\beta}}}}{\arg\min} \|\boldsymbol{y}_k - h(\boldsymbol{X}_k, \boldsymbol{\beta})\|_2^2 \,, \quad k = 1, \ldots, K. \tag{136}$$

Then, for both of the settings, considering our dynamics in Eq. (3), observe that residual errors satisfy

$$\boldsymbol{w}_k \equiv \boldsymbol{\beta}_{k+1} - \rho \boldsymbol{\beta}_k - (1 - \rho)\, \boldsymbol{\beta}_* \tag{137}$$

Exploiting this relation, we first estimate $\hat{\rho}$ by minimizing the average squared residual via:

$$\hat{\rho} = \underset{\rho \in [0,1]}{\arg\min} \left[\frac{1}{K-1} \sum_{k=1}^{K-1} \left\| \hat{\boldsymbol{\beta}}_{k+1} - \rho \hat{\boldsymbol{\beta}}_k - (1 - \rho)\, \hat{\boldsymbol{\beta}}_* \right\|^2 \right], \tag{138}$$

Table 6: Parameters for Sub-Synthetic Scenarios.

Param.	Sub-Syn. I	Sub-Syn. II	Description
p	40	15	# possible candidate $\boldsymbol{x}_i \in \mathbb{R}^d$
n	50	20	# selected $\boldsymbol{x}_i$ each stage
d	30	30	Dimension of $\boldsymbol{x}_i$
ρ	0.9	0.9	State transition constant
σ	0.9	0.9	Measurement noise standard deviation
$\sigma_{\boldsymbol{w}}$	0.9	0.9	Drift noise standard deviation
T	2000	2000	Number of stages

where $K = \lfloor 0.1\,T \rfloor$ is the number of burn-in stages (the first 10% of the T total stages), which are excluded from every reported metric; all estimates in this subsection are computed causally on this prefix, using no information from the evaluated stages. After obtaining $\hat{\rho}$, we estimate the drift noise variance $\sigma^2_{\boldsymbol{w}}$ as the average of the mean square error of these estimated residuals:

$$\hat{\sigma}^2_{\boldsymbol{w}} = \frac{1}{(K-1)d_{\boldsymbol{\beta}}} \sum_{k=1}^{K-1} \left\|\hat{\boldsymbol{w}}_k\right\|^2, \quad \text{where} \quad \hat{\boldsymbol{w}}_k = \hat{\boldsymbol{\beta}}_{k+1} - \hat{\rho}\hat{\boldsymbol{\beta}}_k - (1-\hat{\rho})\,\hat{\boldsymbol{\beta}}_*,\ k \geq 1. \tag{139}$$

Finally, we estimate the measurement noise variance σ^2 as the average of the corresponding noise residuals:

$$\hat{\sigma}^2 = \frac{1}{(K-1)p} \sum_{k=1}^{K-1} \left\|\hat{\boldsymbol{e}}_k\right\|^2, \qquad \text{where} \qquad \hat{\boldsymbol{e}}_k = \boldsymbol{y}_k - \boldsymbol{\mathcal{X}}_k\hat{\boldsymbol{\beta}}_k,\ k \geq 1. \tag{140}$$

These parameters are then provided to the design selection algorithms we tested. Note that $(\hat{\rho}, \hat{\sigma}, \hat{\sigma}_{\boldsymbol{w}})$ characterize the pair (dataset, model class), and not the dataset alone. By Eq. (137), $\boldsymbol{w}_k$ is the drift of the *model parameters*, so $\hat{\sigma}_{\boldsymbol{w}}$ lives in the parameter space of the architecture that is fitted. In addition, Eqs. (138)–(139) plug in the per-stage fits $\hat{\boldsymbol{\beta}}_k$ in place of the unobserved $\boldsymbol{\beta}_k$, so the estimation error of these fits is absorbed into $\hat{\boldsymbol{w}}_k$ along with the true drift. This is what produces the large $\hat{\sigma}_{\boldsymbol{w}}$ that the linear model attains in Table 4 on the datasets where a linear fit is poor, namely 644.75 and 684.94 on the two AT&T tasks and 29.08 on Bike Sharing, against 0.02–0.05 for the non-linear architectures. Fitting a linear model to a strongly non-linear signal yields unstable per-stage coefficients, which inflate $\|\hat{\boldsymbol{w}}_k\|$. The same noisy $\hat{\boldsymbol{\beta}}_k$ also act as regressors in Eq. (138), which in turn attenuates $\hat{\rho}$ towards zero, as observed on the AT&T tasks (0.74 and 0.53, against ≈ 0.9 for the non-linear architectures). These gaps thus reflect how well each model class fits the signal rather than a different underlying data drift.

L.4 Methods

We explored five design selection methods combined with four estimation methods, as described in Table 5. All of them were used on our datasets with the four model functions we discussed in the context of our synthetic dataset (see App. L.1.1): i.e., the linear, ResNet, CNN, and MLP models.

L.4.1 Design Selection Methods

We consider the following design selection methods:

- **Stage-wise Optimal Selection (STG):** This selection method refers to our optimal stage-wise design policy in App. E.1.1, which is determined by solving the convex optimization Prob. (64). We solve this problem via the *cvxpy* convex optimization toolbox.
- **Stationary Optimal Selection (STS):** This selection method refers to our optimal steady-state design policies for linear and non-linear settings, described in Sec. 3.3.2 and App. K.2.2. This is computed by solving Probs. (13) and (133). Contrary to the stage-wise optimization problem, computing the objective involves finding a solution to the DARE system of equations (Eqs. (12) and (134)). This prevents us from applying convex optimization

Table 7: Model error on synthetic data for Linear, ResNet, CNN, and MLP models. **Bold** indicates the best result per column and underline the second-best. Values are mean ± std over experiment seeds.

Method	Linear	ResNet	CNN	MLP
RND-SGD	12.43 ± 1.10	2.75 ± 0.45	3.06 ± 0.43	13.19 ± 2.01
RND-SE	14.20 ± 1.18	8.38 ± 1.87	9.91 ± 1.92	15.62 ± 2.61
RND-AE	8.18 ± 0.96	2.47 ± 0.42	2.47 ± 0.34	12.97 ± 2.22
RND-RLS	9.46 ± 0.98	3.36 ± 0.38	3.61 ± 0.38	13.25 ± 2.18
ADM-SGD	12.33 ± 1.07	2.58 ± 0.44	2.85 ± 0.36	12.41 ± 1.90
ADM-SE	11.45 ± 1.12	8.05 ± 1.72	9.75 ± 1.95	14.66 ± 2.38
ADM-AE	7.31 ± 0.87	2.43 ± 0.37	2.39 ± 0.33	12.11 ± 1.97
ADM-RLS	8.89 ± 0.81	2.98 ± 0.31	3.11 ± 0.35	12.38 ± 1.95
GKS-SGD	12.36 ± 1.17	2.51 ± 0.38	2.87 ± 0.37	12.94 ± 2.05
GKS-SE	11.18 ± 1.02	8.11 ± 1.68	9.58 ± 1.69	14.49 ± 2.45
GKS-AE	7.46 ± 0.81	2.41 ± 0.35	2.45 ± 0.34	12.08 ± 1.92
GKS-RLS	8.71 ± 0.91	3.01 ± 0.29	3.09 ± 0.31	12.55 ± 2.03
STG-SGD	11.36 ± 1.11	2.71 ± 0.41	2.88 ± 0.34	13.39 ± 2.03
STG-SE	10.47 ± 0.84	7.96 ± 1.67	9.62 ± 2.03	16.63 ± 2.90
STG-RLS	6.92 ± 0.61	2.92 ± 0.39	2.95 ± 0.34	11.96 ± 2.14
STG-AE (ours)	$\underline{6.32 \pm 0.49}$	$\underline{1.91 \pm 0.37}$	$\underline{2.03 \pm 0.36}$	$\underline{11.42 \pm 2.31}$
STS-AE (ours)	$\mathbf{6.16 \pm 0.47}$	$\mathbf{1.82 \pm 0.34}$	$\mathbf{1.94 \pm 0.38}$	$\mathbf{11.13 \pm 2.05}$

solvers, like *cvxpy*, directly. Thus, we solve this problem with the *Frank–Wolfe* algorithm by calculating the gradients of the objective functions in Eqs. (13a) and (133a), which involves finding a solution to the DARE system of equations (Eqs. (12) and (134)), using the implicit function theorem [37]. Details of these computations are provided in App. J. Since the objective in Eq. (133a) depends on $\hat{\boldsymbol{\beta}}_*$, in our experiments, we re-solve Prob. (133) at each stage to improve the selection as $\hat{\boldsymbol{\beta}}_*$ is refined at each stage.

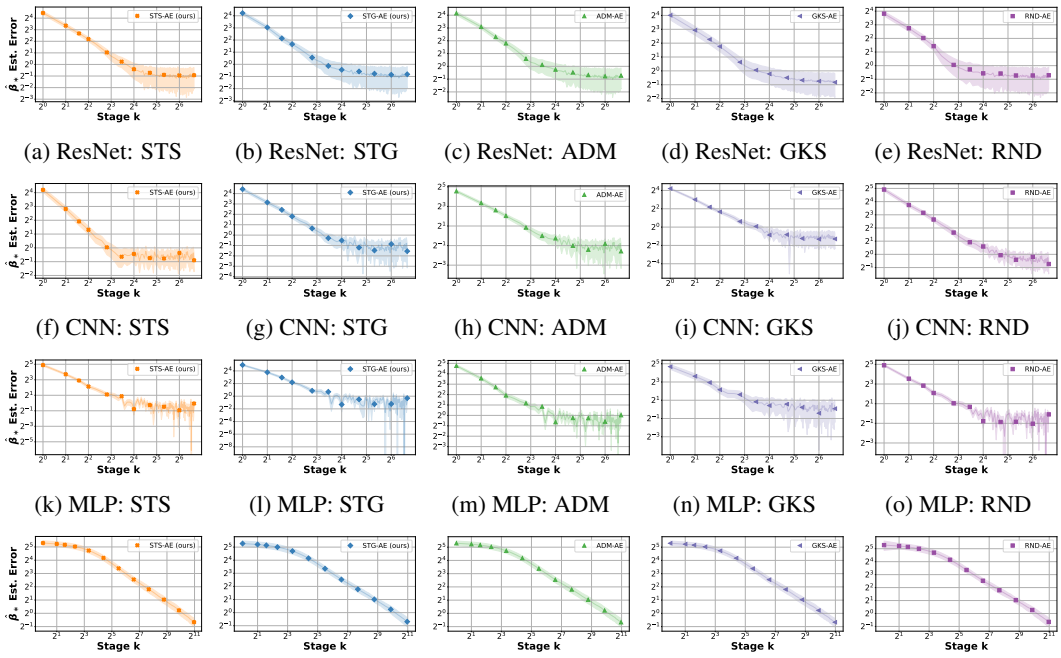
- **Greedy Kalman Information-Gain Selection (GKS):** This is an external design-selection baseline [61] that sequentially selects the n samples that maximize the one-step information gain. Starting from the *a priori* precision $\boldsymbol{M} = \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}^{-1}$, at each of the n picks it selects the candidate that maximizes the leverage score $\boldsymbol{x}_i^\top \boldsymbol{M}^{-1} \boldsymbol{x}_i$ (in the non-linear setting, the Jacobian row $\boldsymbol{g}(\boldsymbol{x}_i, \hat{\boldsymbol{\beta}}_*)$ replaces $\boldsymbol{x}_i$), increments $\boldsymbol{n}_k$ at that index, and updates $\boldsymbol{M}^{-1}$ through a rank-one Sherman–Morrison update. Unlike our convex D-optimal designs (STG, STS), this greedy sensor-selection heuristic comes with no optimality guarantee, and thus isolates the benefit of the proposed design criterion.
- **Adam Selection (ADM):** This selection method performs projected gradient ascent to solve Prob. (9) using the Adam optimizer [34]. At each Adam iteration t, the method forms the gradient of the objective in Eq. (9a) with respect to the i-th design variable as

$$g_k^{t,i} = \frac{n_k}{\sigma^2} \boldsymbol{x}_i^\top \boldsymbol{M}(\boldsymbol{\pi}_k^t)^{-1} \boldsymbol{x}_i, \tag{141}$$

  where $\boldsymbol{M}(\boldsymbol{\pi}_k) = \frac{n_k}{\sigma^2} \boldsymbol{B}(\boldsymbol{\pi}_k) + \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}}^{-1}$ from Eq. (9a). The iterate $\boldsymbol{\pi}_k^t$ is then updated using an Adam ascent step followed by projection onto the feasible simplex, and this procedure is repeated until convergence.
- **Random Selection (RND):** This selection method refers to random selection, which randomly selects n feature vectors from the candidates to construct the design matrix at each stage. The procedure is similar to the "balls-and-bins" method, by initializing the selection vector $\boldsymbol{n}_k \in \mathbb{N}^p$ with zero elements, selecting an index uniformly at random, and incrementing the corresponding entry by 1. After repeating this n times, the selection vector is determined.

L.4.2 Model Estimation

We also consider the following model estimation policies:



(a) ResNet: STS (b) ResNet: STG (c) ResNet: ADM (d) ResNet: GKS (e) ResNet: RND

(f) CNN: STS (g) CNN: STG (h) CNN: ADM (i) CNN: GKS (j) CNN: RND

(k) MLP: STS (l) MLP: STG (m) MLP: ADM (n) MLP: GKS (o) MLP: RND

(p) Linear: STS (q) Linear: STG (r) Linear: ADM (s) Linear: GKS (t) Linear: RND

Figure 6: Estimation error between the estimation $\hat{\boldsymbol{\beta}}_*$ and the true mean fixed-point $\boldsymbol{\beta}_*$ for the AE methods on the synthetic dataset, across four model architectures (rows, top to bottom: ResNet, CNN, MLP, Linear) and five AE methods (columns, left to right: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE). Shaded regions represent ± 1 standard deviation from the mean.

- **Adaptive Estimation (AE):** This estimation method refers to estimating the full posterior via Kalman filter and EKF equations in the linear and non-linear settings in Eqs. (5)–(6) and (122a)–(122b) respectively, for both the model $\hat{\boldsymbol{\beta}}_{k|k}$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ at each stage. In adaptive estimation, we assume by default that the mean fixed point $\boldsymbol{\beta}_*$ in Eq. (3) is unknown, and therefore use the estimator in Eq. (8); we also consider a sub-case in which the mean fixed point is assumed to be known, denoted by K.
- **Forgetting-factor Recursive Least Squares Estimation (RLS):** This is an external recursive-estimation baseline [43] that tracks the drifting model using a classic exponential forgetting factor $\gamma \in (0, 1]$. Its measurement update coincides with the (E)KF update in Eq. (5), but its time update models the parameters as a plain random walk with multiplicative covariance inflation,

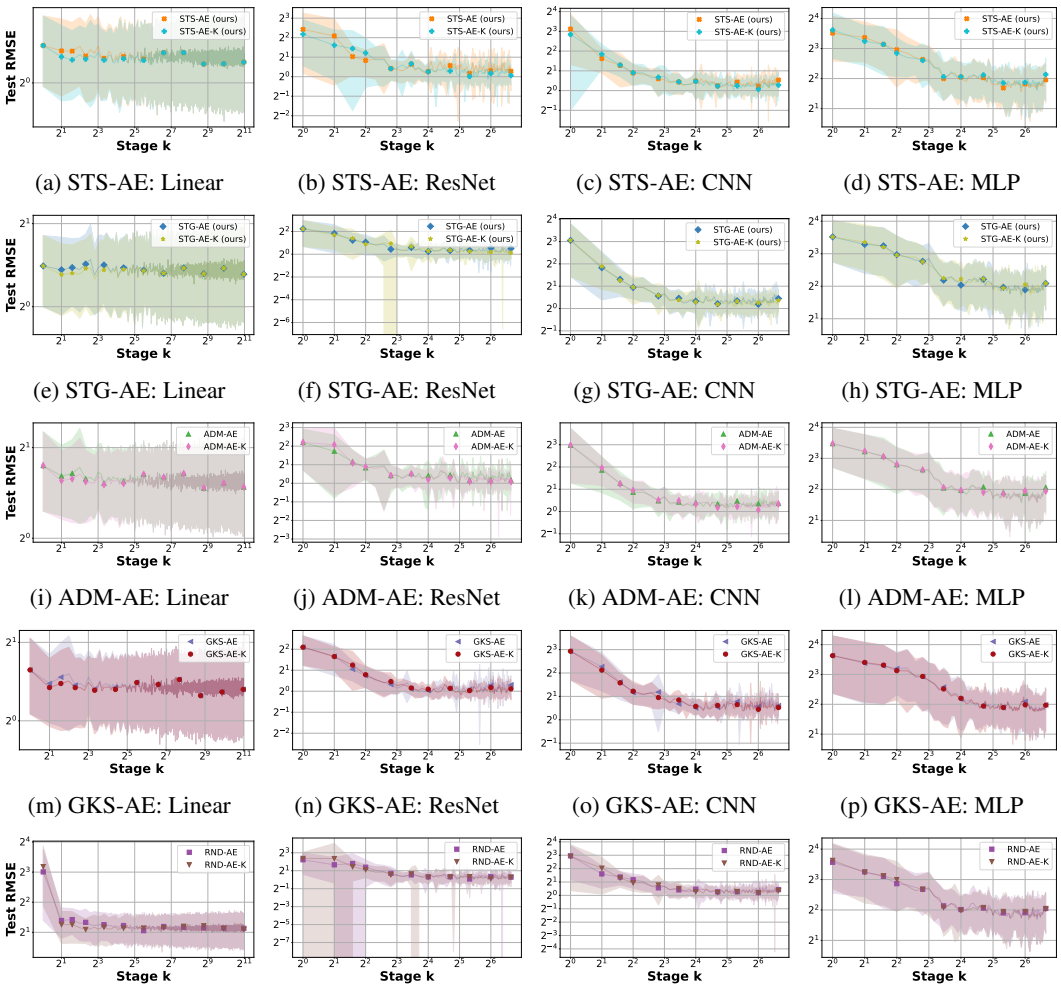
$$\hat{\boldsymbol{\beta}}_{k|k-1} = \hat{\boldsymbol{\beta}}_{k-1|k-1}, \qquad \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k-1}} = \frac{1}{\gamma}\, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k-1|k-1}}, \tag{142}$$

  rather than the mean-reverting dynamics of Eq. (3). It therefore discounts past observations geometrically and does not use the mean fixed-point $\boldsymbol{\beta}_*$. We set $\gamma = 0.98$.
- **Static Estimation (SE):** This estimation method refers to estimating the model using priors [12] and the design matrix $\boldsymbol{X}_k$ along with the collected labels $\boldsymbol{y}_k$ at each stage, using Eq. (32).
- **Stochastic Gradient Descent Estimation (SGD):** This estimation method employs a standard stochastic gradient descent (SGD) update [59] for the model estimation. Given the design matrix $\boldsymbol{X}_k \in \mathbb{R}^{n \times d}$ and the observations $\boldsymbol{y}_k \in \mathbb{R}^n$ at stage k, the empirical squared loss is:

$$\mathcal{L}_k(\boldsymbol{\beta}) = \frac{1}{2n} \left\| \boldsymbol{y}_k - \boldsymbol{X}_k \boldsymbol{\beta} \right\|_2^2 . \tag{143}$$

  The method updates the model estimation at each stage via the SGD step:

$$\hat{\boldsymbol{\beta}}_{k|k} = \hat{\boldsymbol{\beta}}_{k-1|k-1} - \eta_k \nabla \mathcal{L}_k(\hat{\boldsymbol{\beta}}_{k-1|k-1}), \tag{144}$$



(a) STS-AE: Linear (b) STS-AE: ResNet (c) STS-AE: CNN (d) STS-AE: MLP

(e) STG-AE: Linear (f) STG-AE: ResNet (g) STG-AE: CNN (h) STG-AE: MLP

(i) ADM-AE: Linear (j) ADM-AE: ResNet (k) ADM-AE: CNN (l) ADM-AE: MLP

(m) GKS-AE: Linear (n) GKS-AE: ResNet (o) GKS-AE: CNN (p) GKS-AE: MLP

(q) RND-AE: Linear (r) RND-AE: ResNet (s) RND-AE: CNN (t) RND-AE: MLP

Figure 7: Unknown vs. known $\boldsymbol{\beta}_*$ test RMSE comparisons across the five AE methods on the synthetic dataset; lower is better. Rows (top to bottom) correspond to the five AE methods: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE. Columns (left to right) correspond to the four model architectures: Linear, ResNet, CNN, MLP. Shaded regions represent ± 1 standard deviation from the mean.

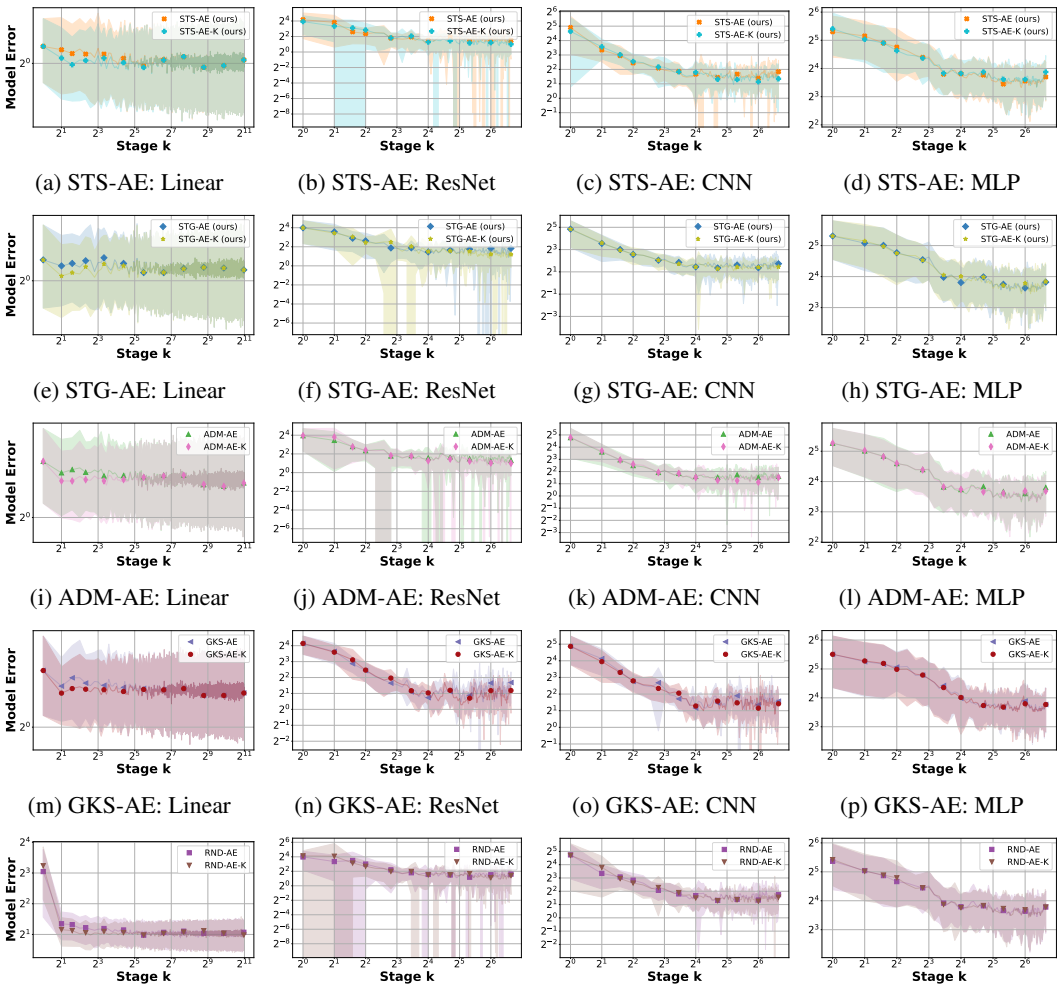
until convergence, where the gradient is $\nabla \mathcal{L}_k(\hat{\boldsymbol{\beta}}_{k-1|k-1}) = -\frac{1}{n}\boldsymbol{X}_k^\top \left(\boldsymbol{y}_k - \boldsymbol{X}_k \hat{\boldsymbol{\beta}}_{k-1|k-1}\right)$ and $\eta_k > 0$ is the learning rate. We run at most 200 epochs per stage, stopping early once the relative change in the loss (or the gradient norm) falls below 10^{-6}.

L.5 Performance Metrics

We evaluate the quality of each method (i.e., a combined design selection and estimation process) using the following metrics. In order to display the per-stage metrics into a single number per experiment in Table 2, we take the average over the number of stages T in Table 1.

Test RMSE. In both synthetic and real data scenarios, over the test set, we measure the root mean square error (RMSE) between the collected test labels and predicted labels. In particular, at each stage, we collect labels for the test set as:

$$\boldsymbol{y}_k^{\text{test}} = f(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{\beta}_k) + \boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \tag{145}$$



(a) STS-AE: Linear (b) STS-AE: ResNet (c) STS-AE: CNN (d) STS-AE: MLP

(e) STG-AE: Linear (f) STG-AE: ResNet (g) STG-AE: CNN (h) STG-AE: MLP

(i) ADM-AE: Linear (j) ADM-AE: ResNet (k) ADM-AE: CNN (l) ADM-AE: MLP

(m) GKS-AE: Linear (n) GKS-AE: ResNet (o) GKS-AE: CNN (p) GKS-AE: MLP

(q) RND-AE: Linear (r) RND-AE: ResNet (s) RND-AE: CNN (t) RND-AE: MLP

Figure 8: Unknown vs. known $\boldsymbol{\beta}_*$ model error comparisons across the five AE methods on the synthetic dataset; lower is better. Rows (top to bottom) correspond to the five AE methods: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE. Columns (left to right) correspond to the four model architectures: Linear, ResNet, CNN, MLP. Shaded regions represent ± 1 standard deviation from the mean.

where $f(\boldsymbol{X}, \boldsymbol{\beta}) = \boldsymbol{X}\boldsymbol{\beta}$ in the linear setting and $f = h$ in the non-linear setting. Predictions are computed as $\boldsymbol{y}_k^{\text{pred}} = f(\boldsymbol{X}_k^{\text{test}}, \hat{\boldsymbol{\beta}}_{k|k})$. Then, we measure the test RMSE as:

$$\begin{aligned} \text{Test RMSE}_k &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{ik}^{\text{test}} - y_{ik}^{\text{pred}}\right)^2} \\ &= \sqrt{\frac{1}{n}\left\|\boldsymbol{y}_k^{\text{test}} - \boldsymbol{y}_k^{\text{pred}}\right\|_2^2}. \end{aligned} \tag{146}$$

To report the average measurement per experiment, we discard the first τ stages as a warm-up (burn-in) period, where $\tau = \lfloor 0.1\,T \rfloor$ corresponds to 10% of the total number of stages, and average the per-stage test RMSE over the remaining $T - \tau$ stages. Specifically, the mean test RMSE per experiment is:

$$\overline{\text{Test RMSE}} = \frac{1}{T-\tau}\sum_{k=\tau+1}^{T} \text{Test RMSE}_k. \tag{147}$$

Table 8: Test RMSE on synthetic and real datasets, comparing each -AE method (unknown $\boldsymbol{\beta}_*$) with its -K variant (known $\boldsymbol{\beta}_*$) for the Linear, ResNet, CNN, and MLP models. **Bold** indicates the best result per model and underline the second-best. The -AE values are reproduced from Table 2. Mean $\pm$ std reported over 100 trials with different random seeds.

Model	Method	Beijing PM2.5	S&P 500	Bike Sharing	Cal. Housing	AT&T Downlink	AT&T Uplink	Synthetic
Linear	RND-AE	142.53 ± 0.50	2.71 ± 0.73	267.70 ± 1.00	3.72 ± 0.01	146.30 ± 1.69	20.77 ± 0.95	3.20 ± 0.05
	RND-AE-K	142.39 ± 0.52	2.68 ± 0.61	265.95 ± 1.07	3.66 ± 0.02	142.12 ± 1.59	20.64 ± 0.83	3.16 ± 0.06
	ADM-AE	141.46 ± 0.70	1.79 ± 0.57	265.40 ± 0.31	3.70 ± 0.02	142.34 ± 1.09	12.17 ± 0.58	2.74 ± 0.08
	ADM-AE-K	141.01 ± 0.46	1.74 ± 0.51	261.40 ± 0.38	3.48 ± 0.01	138.98 ± 0.92	11.19 ± 0.53	2.66 ± 0.06
	GKS-AE	141.65 ± 0.52	1.88 ± 0.55	263.01 ± 0.41	3.71 ± 0.02	143.51 ± 1.14	15.67 ± 0.93	2.79 ± 0.09
	GKS-AE-K	141.21 ± 0.53	1.85 ± 0.53	261.86 ± 0.40	3.62 ± 0.02	139.18 ± 1.08	14.88 ± 0.87	2.68 ± 0.08
	STG-AE (ours)	138.61 ± 0.60	1.68 ± 0.51	252.97 ± 0.39	3.66 ± 0.01	138.97 ± 0.85	11.10 ± 0.81	2.66 ± 0.08
	STG-AE-K (ours)	138.50 ± 0.41	$\mathbf{1.65 \pm 0.52}$	250.31 ± 0.34	$\mathbf{3.32 \pm 0.02}$	135.78 ± 0.90	10.88 ± 0.69	2.62 ± 0.07
	STS-AE (ours)	133.82 ± 0.63	1.75 ± 0.53	244.18 ± 0.31	3.68 ± 0.01	132.78 ± 0.90	10.51 ± 0.78	2.56 ± 0.06
	STS-AE-K (ours)	$\mathbf{133.14 \pm 0.55}$	1.71 ± 0.49	$\mathbf{242.70 \pm 0.25}$	3.38 ± 0.01	$\mathbf{129.83 \pm 0.87}$	$\mathbf{10.39 \pm 0.71}$	$\mathbf{2.51 \pm 0.04}$
ResNet	RND-AE	99.24 ± 0.39	0.76 ± 0.03	180.01 ± 0.35	1.32 ± 0.15	131.51 ± 0.33	7.66 ± 0.27	1.29 ± 0.10
	RND-AE-K	97.26 ± 0.46	0.71 ± 0.02	177.91 ± 0.32	1.26 ± 0.11	128.14 ± 0.35	7.51 ± 0.24	1.21 ± 0.09
	ADM-AE	96.31 ± 0.32	0.75 ± 0.02	177.88 ± 0.34	1.23 ± 0.12	127.44 ± 0.41	6.86 ± 0.23	1.29 ± 0.09
	ADM-AE-K	95.28 ± 0.36	0.69 ± 0.02	173.49 ± 0.31	1.21 ± 0.10	123.63 ± 0.35	6.68 ± 0.24	1.18 ± 0.09
	GKS-AE	94.89 ± 1.38	0.81 ± 0.05	179.93 ± 0.62	1.29 ± 0.19	129.45 ± 0.55	10.95 ± 0.86	1.11 ± 0.12
	GKS-AE-K	93.12 ± 1.31	0.78 ± 0.05	176.98 ± 0.59	1.21 ± 0.20	126.39 ± 0.51	10.24 ± 0.71	1.06 ± 0.11
	STG-AE (ours)	92.22 ± 0.38	0.67 ± 0.03	174.81 ± 0.27	1.24 ± 0.17	123.32 ± 0.29	6.74 ± 0.32	0.98 ± 0.07
	STG-AE-K (ours)	91.48 ± 0.20	0.63 ± 0.02	$\mathbf{171.48 \pm 0.28}$	1.18 ± 0.18	121.61 ± 0.30	6.63 ± 0.31	0.95 ± 0.06
	STS-AE (ours)	87.28 ± 0.32	0.60 ± 0.04	177.88 ± 0.33	1.19 ± 0.16	122.43 ± 0.35	6.60 ± 0.34	0.89 ± 0.08
	STS-AE-K (ours)	$\mathbf{86.58 \pm 0.31}$	$\mathbf{0.57 \pm 0.03}$	172.63 ± 0.28	$\mathbf{1.15 \pm 0.15}$	$\mathbf{120.51 \pm 0.32}$	$\mathbf{6.56 \pm 0.32}$	$\mathbf{0.86 \pm 0.05}$
CNN	RND-AE	96.32 ± 0.21	0.76 ± 0.02	180.92 ± 0.19	1.32 ± 0.10	131.46 ± 0.30	7.79 ± 0.27	1.39 ± 0.07
	RND-AE-K	94.36 ± 0.18	0.73 ± 0.01	178.96 ± 0.15	1.28 ± 0.09	128.89 ± 0.29	7.19 ± 0.29	1.32 ± 0.08
	ADM-AE	95.26 ± 0.24	0.74 ± 0.01	179.96 ± 0.16	1.25 ± 0.09	128.43 ± 0.25	6.76 ± 0.25	1.37 ± 0.07
	ADM-AE-K	93.88 ± 0.20	0.69 ± 0.01	175.91 ± 0.13	1.23 ± 0.08	126.94 ± 0.26	6.18 ± 0.23	1.32 ± 0.06
	GKS-AE	95.31 ± 0.22	0.73 ± 0.03	178.92 ± 0.18	1.26 ± 0.15	128.45 ± 0.24	6.89 ± 0.28	1.38 ± 0.08
	GKS-AE-K	94.11 ± 0.20	0.69 ± 0.02	175.58 ± 0.15	1.23 ± 0.11	125.84 ± 0.21	6.02 ± 0.25	1.31 ± 0.07
	STG-AE (ours)	93.23 ± 0.17	0.66 ± 0.04	177.93 ± 0.15	1.20 ± 0.12	123.47 ± 0.26	5.86 ± 0.22	1.39 ± 0.12
	STG-AE-K (ours)	91.42 ± 0.18	0.61 ± 0.02	174.85 ± 0.11	$\mathbf{1.15 \pm 0.08}$	120.92 ± 0.32	5.80 ± 0.21	1.31 ± 0.07
	STS-AE (ours)	88.23 ± 0.13	0.62 ± 0.04	173.93 ± 0.17	1.24 ± 0.10	121.39 ± 0.27	5.78 ± 0.26	1.35 ± 0.10
	STS-AE-K (ours)	$\mathbf{86.48 \pm 0.13}$	$\mathbf{0.59 \pm 0.01}$	$\mathbf{170.74 \pm 0.31}$	1.19 ± 0.09	$\mathbf{119.47 \pm 0.28}$	$\mathbf{5.71 \pm 0.24}$	$\mathbf{1.30 \pm 0.06}$
MLP	RND-AE	95.70 ± 0.65	0.78 ± 0.03	180.80 ± 0.26	1.35 ± 0.12	128.36 ± 0.35	9.74 ± 0.34	3.86 ± 0.62
	RND-AE-K	93.49 ± 0.66	0.75 ± 0.02	177.83 ± 0.29	1.31 ± 0.11	126.65 ± 0.34	9.18 ± 0.32	3.80 ± 0.60
	ADM-AE	93.65 ± 0.76	0.75 ± 0.02	179.99 ± 0.32	1.34 ± 0.11	127.46 ± 0.31	7.95 ± 0.41	3.61 ± 0.55
	ADM-AE-K	92.85 ± 0.73	0.71 ± 0.02	176.74 ± 0.30	1.28 ± 0.13	125.19 ± 0.28	7.91 ± 0.38	3.59 ± 0.53
	GKS-AE	93.88 ± 0.89	0.81 ± 0.08	179.85 ± 0.30	1.34 ± 0.11	128.10 ± 0.35	8.46 ± 0.35	3.59 ± 0.54
	GKS-AE-K	91.74 ± 0.85	0.77 ± 0.08	175.94 ± 0.26	1.29 ± 0.10	126.85 ± 0.31	8.31 ± 0.34	3.51 ± 0.52
	STG-AE (ours)	91.65 ± 0.86	0.69 ± 0.02	177.97 ± 0.32	1.28 ± 0.12	124.29 ± 0.34	8.21 ± 0.33	2.98 ± 0.65
	STG-AE-K (ours)	90.88 ± 0.81	$\mathbf{0.65 \pm 0.01}$	174.48 ± 0.28	$\mathbf{1.24 \pm 0.09}$	122.67 ± 0.32	8.09 ± 0.32	2.94 ± 0.63
	STS-AE (ours)	89.70 ± 0.68	0.72 ± 0.03	174.99 ± 0.33	1.31 ± 0.16	121.45 ± 0.29	7.90 ± 0.29	2.63 ± 0.57
	STS-AE-K (ours)	$\mathbf{88.37 \pm 0.69}$	0.68 ± 0.01	$\mathbf{172.71 \pm 0.28}$	1.27 ± 0.11	$\mathbf{118.92 \pm 0.28}$	$\mathbf{7.84 \pm 0.29}$	$\mathbf{2.56 \pm 0.53}$

Note that, as test labels are available in both synthetic and real data scenarios, we report this metric for all experiments.

Model Error. In synthetic experiments, where the true model $\boldsymbol{\beta}_k$ is available, we measure the model error at each stage k with the Euclidean distance between the true model $\boldsymbol{\beta}_k$ and the estimator $\hat{\boldsymbol{\beta}}_{k|k}$. In particular:

$$\text{Model Error}_k = \left\| \boldsymbol{\beta}_k - \hat{\boldsymbol{\beta}}_{k|k} \right\|_2 . \tag{148}$$

In the non-linear setting, we instead evaluate the discrepancy through the model outputs over the test set:

$$\text{Model Error}_k = \left\| h(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{\beta}_k) - h(\boldsymbol{X}_k^{\text{test}}, \hat{\boldsymbol{\beta}}_{k|k}) \right\|_2 . \tag{149}$$

The mean model error per experiment is:

$$\overline{\text{Model Error}} = \frac{1}{T-\tau} \sum_{k=\tau+1}^{T} \text{Model Error}_k, \tag{150}$$

where T is the number of stages from Table 1. Note that we report this for synthetic experiments only, as we do not have such a true model $\boldsymbol{\beta}_k$ in the real data scenarios.

Objective Value. Lastly, for the linear model utilized in synthetic sub-scenarios in App. L.7, we use the objective value at each stage. From Prob. (64), the objective formula for the stage-wise optimal

Table 9: Average per-stage wall-clock time (seconds) of each design-selection method, reported per dataset and model architecture, averaged over estimation variants, stages, and 100 seeds.

Model	Method	Beijing PM2.5	S&P 500	Bike Sharing	Cal. Housing	AT&T Downlink	AT&T Uplink	Synthetic
Linear	STG	0.060 ± 0.006	0.042 ± 0.004	0.064 ± 0.005	0.092 ± 0.005	0.083 ± 0.006	0.069 ± 0.010	0.689 ± 0.021
	STS	0.002 ± 0.001	0.004 ± 0.001	0.002 ± 0.001	0.005 ± 0.001	0.013 ± 0.003	0.007 ± 0.002	0.023 ± 0.005
	GKS	0.003 ± 0.003	0.002 ± 0.001	0.003 ± 0.003	0.004 ± 0.003	0.002 ± 0.003	0.002 ± 0.003	0.009 ± 0.004
	ADM	0.080 ± 0.009	0.069 ± 0.009	0.078 ± 0.009	0.080 ± 0.009	0.059 ± 0.029	0.044 ± 0.037	0.117 ± 0.007
	RND	0.002 ± 0.003	0.001 ± 0.002	0.001 ± 0.002	0.002 ± 0.003	0.001 ± 0.003	0.001 ± 0.003	0.002 ± 0.001
ResNet	STG	1.280 ± 0.063	1.320 ± 0.095	1.290 ± 0.140	1.590 ± 0.143	1.290 ± 0.108	1.280 ± 0.117	4.660 ± 5.500
	STS	2.960 ± 3.160	1.670 ± 2.170	0.516 ± 0.349	8.700 ± 6.590	1.890 ± 1.500	2.290 ± 1.170	8.720 ± 7.800
	GKS	0.130 ± 0.057	0.119 ± 0.068	0.125 ± 0.057	0.154 ± 0.085	0.153 ± 0.092	0.130 ± 0.068	0.306 ± 0.252
	ADM	0.242 ± 0.065	0.130 ± 0.074	0.222 ± 0.074	0.257 ± 0.088	0.232 ± 0.069	0.180 ± 0.081	0.454 ± 0.254
	RND	0.119 ± 0.057	0.115 ± 0.062	0.115 ± 0.055	0.132 ± 0.080	0.120 ± 0.060	0.117 ± 0.059	0.280 ± 0.246
CNN	STG	0.771 ± 0.082	0.758 ± 0.087	0.778 ± 0.115	0.971 ± 0.117	0.772 ± 0.091	0.785 ± 0.119	3.090 ± 6.330
	STS	2.280 ± 2.420	1.820 ± 2.470	0.554 ± 0.730	6.880 ± 4.360	1.750 ± 1.430	2.200 ± 1.170	7.370 ± 4.800
	GKS	0.113 ± 0.069	0.104 ± 0.082	0.128 ± 0.098	0.126 ± 0.082	0.117 ± 0.079	0.112 ± 0.073	0.175 ± 0.132
	ADM	0.205 ± 0.051	0.109 ± 0.077	0.195 ± 0.051	0.222 ± 0.080	0.201 ± 0.059	0.173 ± 0.076	0.315 ± 0.129
	RND	0.093 ± 0.049	0.091 ± 0.068	0.089 ± 0.046	0.105 ± 0.072	0.090 ± 0.052	0.095 ± 0.065	0.162 ± 0.131
MLP	STG	0.563 ± 0.037	0.489 ± 0.050	0.702 ± 0.046	0.551 ± 0.050	1.340 ± 0.100	0.659 ± 0.049	3.620 ± 7.230
	STS	0.320 ± 0.152	0.240 ± 0.040	0.434 ± 0.460	1.060 ± 0.718	0.360 ± 0.110	0.312 ± 0.122	7.500 ± 8.713
	GKS	0.069 ± 0.034	0.067 ± 0.054	0.071 ± 0.034	0.080 ± 0.043	0.069 ± 0.036	0.069 ± 0.040	0.101 ± 0.063
	ADM	0.070 ± 0.034	0.065 ± 0.045	0.127 ± 0.063	0.090 ± 0.048	0.073 ± 0.036	0.073 ± 0.037	0.210 ± 0.070
	RND	0.064 ± 0.035	0.060 ± 0.043	0.062 ± 0.033	0.068 ± 0.039	0.061 ± 0.034	0.064 ± 0.036	0.085 ± 0.062

design is:

$$\text{Objective Value}_k = \log\det\left[\frac{n_k}{\sigma^2}\boldsymbol{B}(\boldsymbol{\pi}_k) + \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}}\right], \tag{151}$$

where $\boldsymbol{\pi}_k \in [0,1]^{p_k}$ is the fractional policy at stage $k \in \mathbb{N}$, $\boldsymbol{B}(\boldsymbol{\pi}) = \sum_{i=1}^{p_k} \pi_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$, and $\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\xi}_{k|k-1}} \in \mathbb{R}^{d\times d}$ is the inverse of the a priori covariance given by Eq. (61b). The mean objective value per experiment is:

$$\overline{\text{Objective Value}} = \frac{1}{T-\tau}\sum_{k=\tau+1}^{T} \text{Objective Value}_k. \tag{152}$$

Note that, for the optimal stationary policy (STS-AE), the DARE implies that, if we start with the steady-state covariance $\overline{\boldsymbol{\Sigma}}_{\boldsymbol{\xi}}$ (Thm. 4), as we do, then the covariance remains at this value, and the objective value in Eq. (151) is constant over time.

L.6 Computing Infrastructure

All computations were performed on a high-performance computing cluster managed by SLURM. The experiments were run on CPU nodes equipped with dual-socket Intel Xeon E5-2680 v2 processors (40 logical cores, 2.80 GHz) and 100 GB of allocated memory per job. Jobs were submitted via *sbatch* using single-core execution.

Table 9 reports the average per-stage wall-clock time (in seconds) of each design-selection method on this infrastructure, averaged over all stages, estimation variants, and seeds, and reported separately for each dataset and model architecture. The reported times are dominated by the design-selection step; the model-estimation overhead (AE, RLS, SE, SGD) is comparable across selection methods. These times are modest, and since each stage corresponds to a batch of newly collected data, they sit comfortably within a setting where retraining happens periodically.

L.7 Additional Synthetic Sub-Scenarios (I&II)

In addition to the synthetic experiment in the experimental setup in Sec. 4.1, we consider two synthetic sub-scenarios for the linear setting (Sub-Scenarios I and II), summarized in Table 6, in which we simulate the same dynamics in Eq. (3). In these sub-scenarios, we simulate the linear model in Eq. (4) to analyze the impacts of different setups on our setting.

In particular, in Sub-Scenario I, we set the parameters the same as the main synthetic dataset in Table 1 ($d = 30$, $p = 40$, $n = 50$), but the set of candidate experiments $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}_i\}_{i=1}^{p} \subset \mathbb{R}^d$ is fixed through the stages. In Sub-Scenario II, we set $d = 30$, $p = 15$, and $n = 20$; note that, in this case, $p < d$.

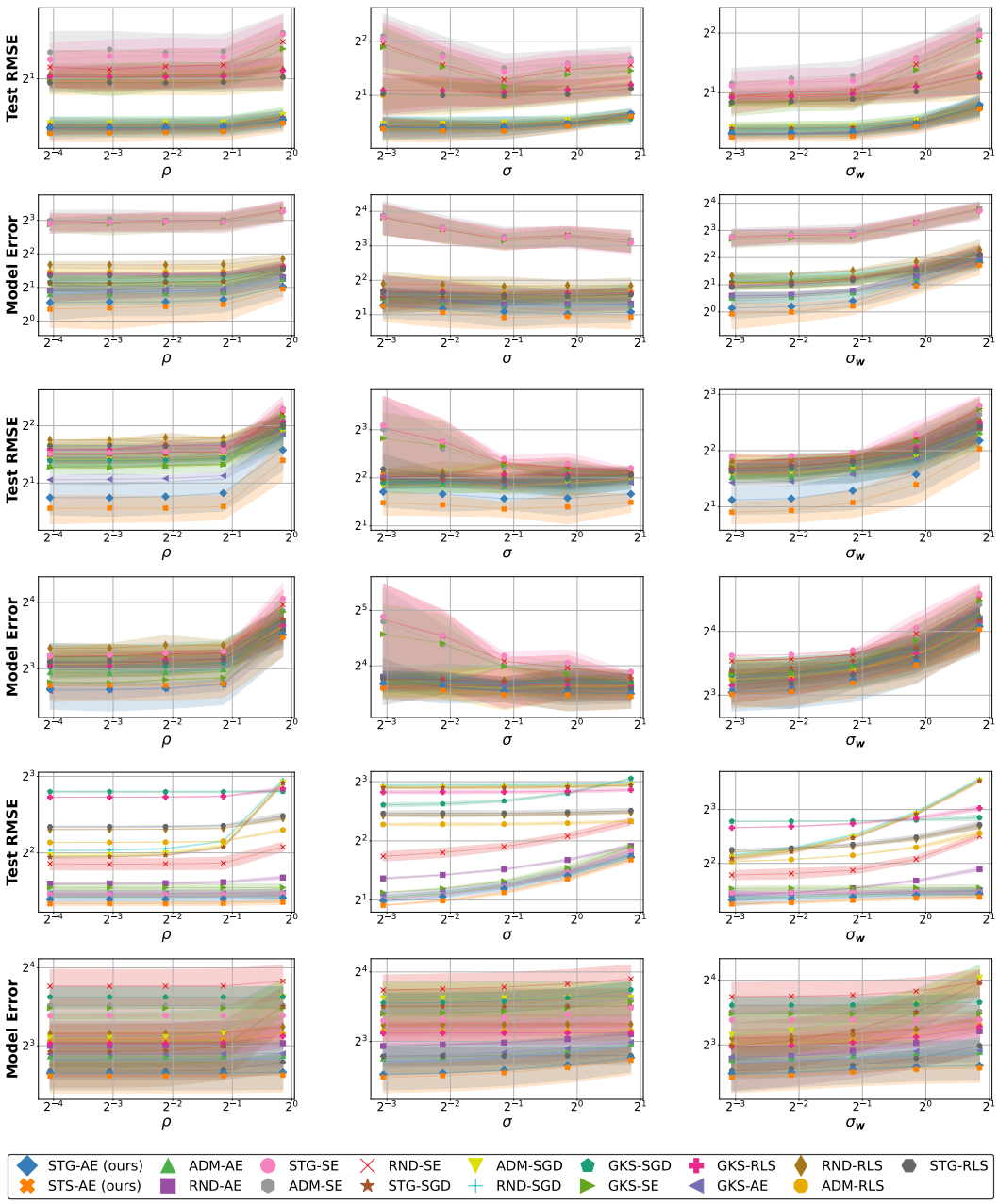
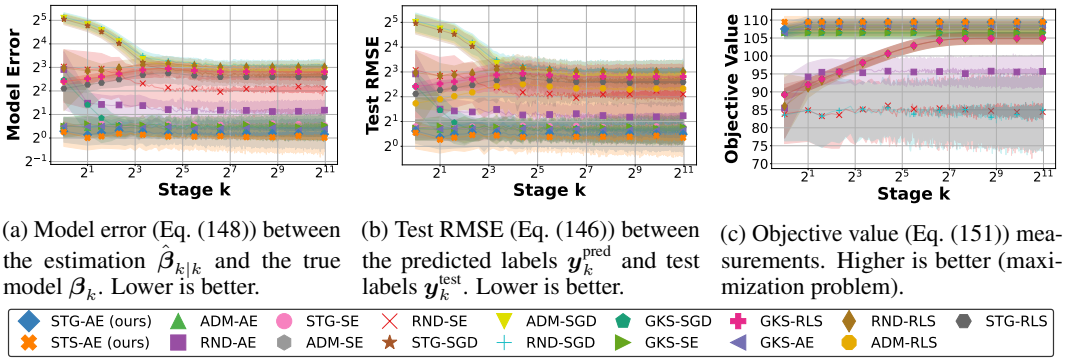


Figure 9: Test RMSE (Eq. (146)) and model error (Eqs. (148)–(149)) comparisons with respect to different values of ρ, σ, and $\sigma_{\boldsymbol{w}}$ on the synthetic dataset, for the CNN (rows 1–2), MLP (rows 3–4), and linear model (rows 5–6). Within each pair of rows, the upper row reports test RMSE and the lower row reports model error. Other parameters as in Table 4. Lower is better. Shaded regions represent ± 1 standard deviation from the mean.

We sample $\boldsymbol{x}_{ik} \in \mathbb{R}^d$ in $\boldsymbol{\mathcal{X}}_k$ from a normal distribution with zero mean and unit variance, i.e., $\boldsymbol{x}_{ik} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, sampled once in Sub-Scenario I and independently at each stage in Sub-Scenario II. Beyond the aforementioned set of candidate feature vectors, we create another synthetic set of feature vectors $\boldsymbol{X}_k^{\text{test}} = \{\boldsymbol{x}_{ik}\}_{i=1}^{n/4} \subset \mathbb{R}^d$, coupled with labels $\boldsymbol{y}^{\text{test}} \in \mathbb{R}^{n/4}$ generated via Eq. (2) with the ground truth model $\boldsymbol{\beta}_k$. We use this test set to measure the predictive performance of our estimates (see also the performance metrics in Sec. 4.1). In both synthetic scenarios, we set the parameters in the model dynamics (Eq. (3)) as $\rho = 0.9$, and the noise parameters to $\sigma = 0.9$ and $\sigma_{\boldsymbol{w}} = 0.9$



(a) Model error (Eq. (148)) between the estimation $\hat{\boldsymbol{\beta}}_{k|k}$ and the true model $\boldsymbol{\beta}_k$. Lower is better.

(b) Test RMSE (Eq. (146)) between the predicted labels $\boldsymbol{y}_k^{\text{pred}}$ and test labels $\boldsymbol{y}_k^{\text{test}}$. Lower is better.

(c) Objective value (Eq. (151)) measurements. Higher is better (maximization problem).

Figure 10: Performance w.r.t. model error, test RMSE, and objective value for synthetic sub-scenario I as a function of the stage k. Shaded regions represent ± 1 standard deviation from the mean. We observe that our methods STG-AE and STS-AE achieve lower model error and test RMSE, as well as higher objective value compared to other methods.

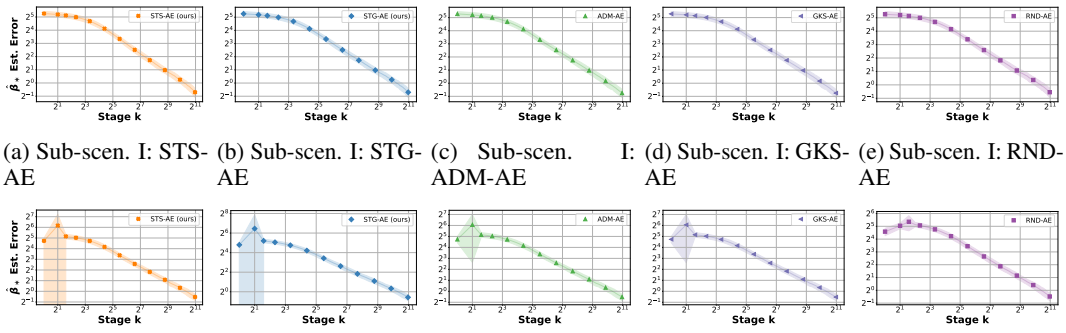
(see Table 6). The prior mean is randomly sampled as $\boldsymbol{\beta}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the prior covariance is constructed as $\boldsymbol{\Sigma}_0 = c\mathbf{I}$, where $c = 50$. The first stage model is sampled as $\boldsymbol{\beta}_1 \sim \mathcal{N}(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. Moreover, the mean fixed-point is randomly sampled as $\boldsymbol{\beta}_* \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We run each scenario for a total of $T = 2000$ stages; we repeat this 100 times with different random seeds to compute confidence intervals.

We note that we *purposefully design the sub-scenario II* to pose a significant challenge for inference, as the number of candidate features is smaller than the dimension (i.e., $p < d$). This makes estimation and tracking of changes due to drift even more challenging. Moreover, the conditions of Thm. 2 are violated; to address this, we regularize the linear estimates in Eq. (8) by adding a small quadratic penalty (i.e., performing ridge regression); this modified estimator still approximates $\boldsymbol{\beta}_*$ quite well in practice, even if the estimator does not converge (see App. M.2.4), learning missing directions as they are revealed. In contrast, in Sub-Scenario I, the conditions of Thm. 2 are always satisfied, as the design matrix is always rank d.

M Additional Experimental Results

M.1 Model Error Across Architectures for the Synthetic Scenarios

Recall that, in the synthetic scenarios, we have access to the ground truth models. Table 7 reports the model error on synthetic data across the four model architectures (Linear, ResNet, CNN, MLP) over the synthetic dataset. Our proposed methods STS-AE and STG-AE achieve the lowest error under every architecture, with STS-AE yielding the best overall result on CNN (1.94) and ResNet (1.82). The gap between adaptive (-AE) and the (-SE), (-SGD), and (-RLS) estimation baselines is substantial regardless of the selection strategy, underscoring the importance of Kalman-based parameter tracking when the model drifts. On the selection side, the benefit of optimal design over greedy, random or Adam-based selection is present across all architectures but is especially pronounced for the linear model, where STS-AE (6.16) halves the error of RND-SGD (12.43). We note that it is the margin of the external greedy and the Adam-based baselines over random selection that narrows on the non-linear architectures, rather than ours. On ResNet and CNN, GKS-AE (2.41, 2.45) and ADM-AE (2.43, 2.39) improve upon RND-AE (2.47, 2.47) by at most a few percent, while on the linear model they improve by 9% and 11%, respectively; STS-AE, by contrast, stays 19–26% below all three on both architectures. This is consistent with the fact that neither baseline solves Prob. (9) to optimality. The greedy rule commits the budget to the few candidates of highest leverage, and projected Adam ascent converges to a point that need not be optimal, and the resulting loss is architecture-dependent, whereas our convex solvers attain the optimum by Thms. 3 and 5. Among non-linear architectures, CNN and ResNet consistently track the true parameters more accurately than MLP, likely due to their structural inductive biases (convolutions and skip connections) producing a more favorable loss landscape for the EKF updates.



(a) Sub-scen. I: STS-AE (b) Sub-scen. I: STG-AE (c) Sub-scen. I: ADM-AE (d) Sub-scen. I: GKS-AE (e) Sub-scen. I: RND-AE

(f) Sub-scen. II: STS-AE (g) Sub-scen. II: STG-AE (h) Sub-scen. II: ADM-AE (i) Sub-scen. II: GKS-AE (j) Sub-scen. II: RND-AE

Figure 11: Estimation error between the estimation $\hat{\boldsymbol{\beta}}_*$ and the true mean fixed-point $\boldsymbol{\beta}_*$ for the AE methods on the synthetic dataset. Rows (top to bottom) correspond to the two synthetic sub-scenarios: Sub-scenario I and Sub-scenario II. Columns (left to right) correspond to the five AE methods: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE. Shaded regions represent ± 1 standard deviation from the mean.

M.2 Additional Synthetic Scenarios

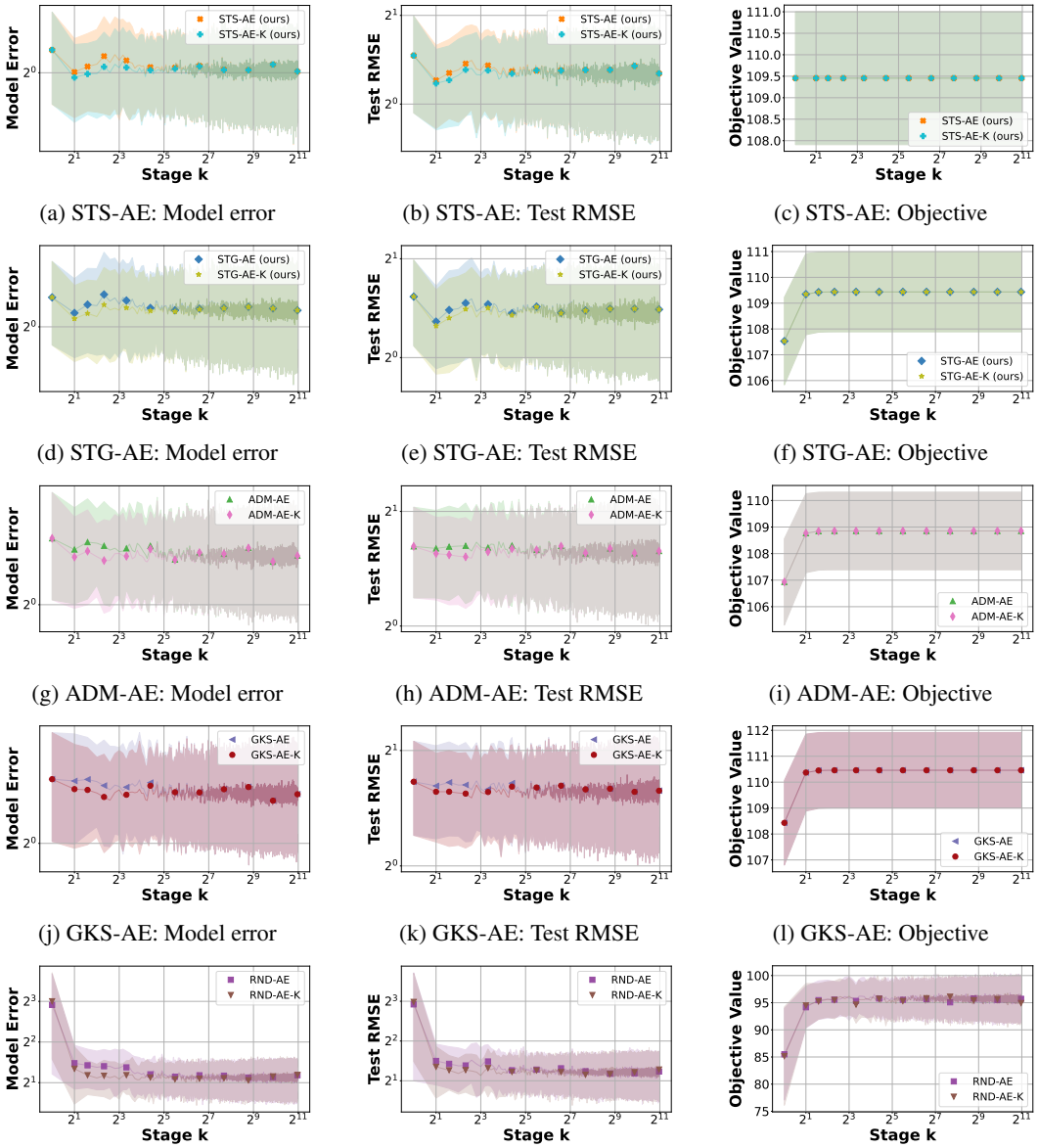
M.2.1 Convergence of $\hat{\beta}_*$ Across Architectures and Impact of Estimation

We plot the convergence of the mean fixed-point estimator evaluated in (8) and (127) for linear and non-linear model functions, respectively, in Fig. 6 for the synthetic scenario. We report the estimation difference $\|\hat{\boldsymbol{\beta}}_* - \boldsymbol{\beta}_*\|_2$ between the estimation $\hat{\boldsymbol{\beta}}_*$ and the true mean fixed-point $\boldsymbol{\beta}_*$ for STG-AE, STS-AE, ADM-AE, GKS-AE, and RND-AE described in Table 5 in Fig. 6: note that estimates are somewhat affected by the selection process, as the latter determines $\boldsymbol{X}_k$; this is why we report these across different methods, even though (8) and (127) stay the same. Similarly, for the non-linear setting, we report $\left\| h(\boldsymbol{X}_k^{\text{test}}, \boldsymbol{\beta}_*) - h(\boldsymbol{X}_k^{\text{test}}, \hat{\boldsymbol{\beta}}_*) \right\|_2$ for the MLP, CNN, and ResNet models in Fig. 6. For all methods, we observe that as k increases, the estimation difference decreases; in particular, our estimator $\hat{\boldsymbol{\beta}}_*$ converges to the true mean fixed-point $\boldsymbol{\beta}_*$.

To further assess the estimation quality of $\boldsymbol{\beta}_*$, in Table 8 and Figs. 7–8, we compare the AE methods with their K variants, in which the mean fixed-point $\boldsymbol{\beta}_*$ is known and used during the model estimation process. For all methods, we observe that the AE methods exhibit very similar performance to their K variants, for all metrics. Across the four model architectures and seven datasets, the gap between each AE method and its K variant is uniformly small, typically within a few percent of the test RMSE, indicating that our estimator $\hat{\boldsymbol{\beta}}_*$ is sufficiently close to the true $\boldsymbol{\beta}_*$ that the downstream design and estimation policies behave nearly identically in both cases. In most cells, the best result per model is a K variant (typically STS-AE-K, with STG-AE-K winning the remaining cases), and the corresponding AE counterpart is the second-best, indicating that our methods sit immediately below their K variants without any baseline able to bridge the gap. The relative ordering across methods is also preserved between the two settings: STS-based methods outperform STG-based ones, which in turn outperform GKS, ADM and RND, irrespective of whether $\boldsymbol{\beta}_*$ is known or estimated. Since the K variants act as an oracle upper bound, the closeness of the AE methods to this bound corroborates the empirical convergence of $\hat{\boldsymbol{\beta}}_*$ shown in Fig. 6, and indicates that our estimation procedure does not meaningfully limit performance in practice.

M.2.2 Impact of System Parameters

In Fig. 9, for CNN, MLP, and linear models, we report the test RMSE (Eq. (146)) and the model error (Eqs. (148)–(149)) under the impact of varying system parameters ρ, σ, and $\sigma_{\boldsymbol{w}}$ for each of them while keeping the others fixed at their default values (see Table 4). Similar to the results for the ResNet model in Fig. 3, our proposed methods achieve lower model error compared to the baselines.



(a) STS-AE: Model error (b) STS-AE: Test RMSE (c) STS-AE: Objective

(d) STG-AE: Model error (e) STG-AE: Test RMSE (f) STG-AE: Objective

(g) ADM-AE: Model error (h) ADM-AE: Test RMSE (i) ADM-AE: Objective

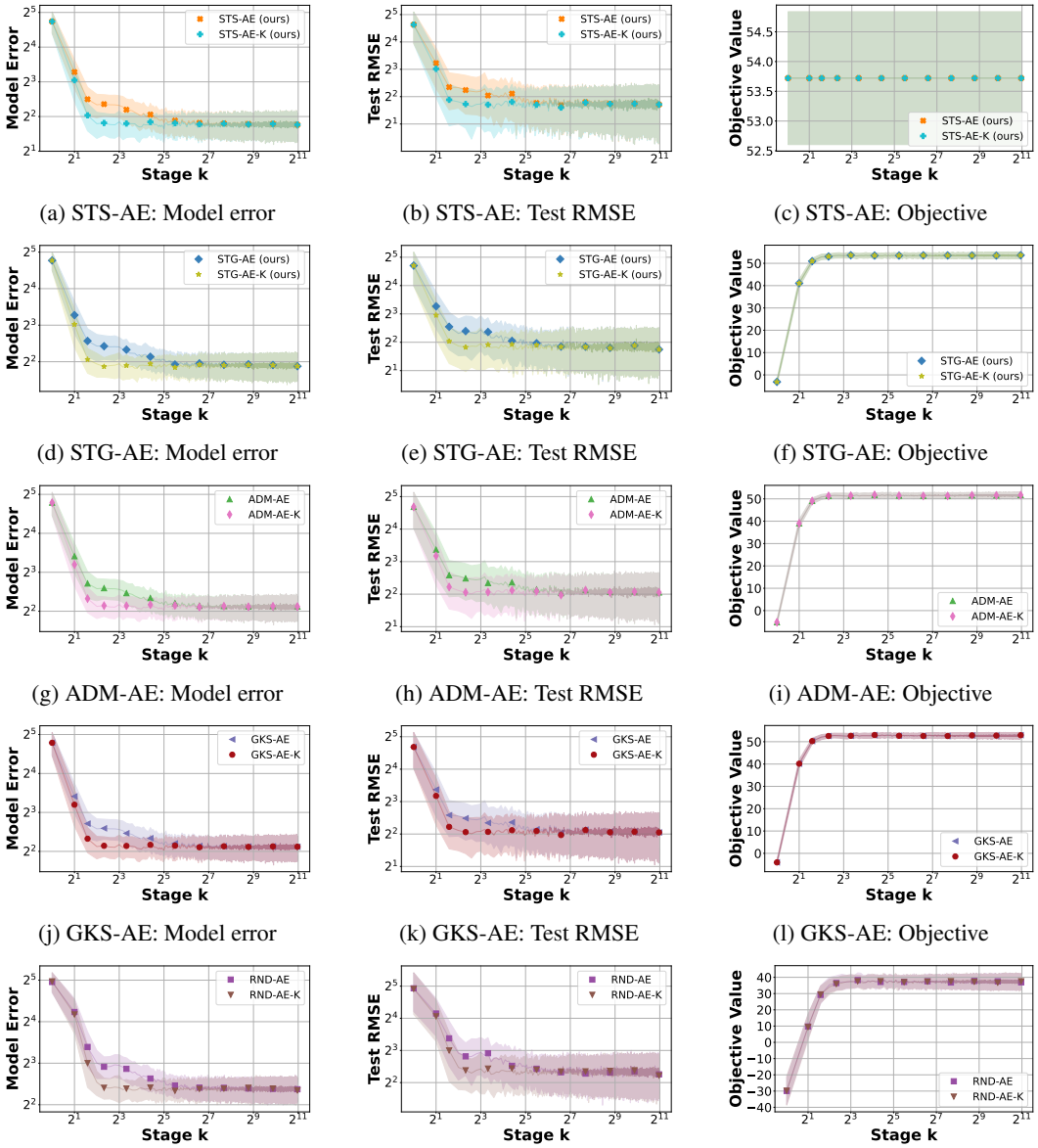
(j) GKS-AE: Model error (k) GKS-AE: Test RMSE (l) GKS-AE: Objective

(m) RND-AE: Model error (n) RND-AE: Test RMSE (o) RND-AE: Objective

Figure 12: Unknown vs. known $\boldsymbol{\beta}_*$ comparisons across the five AE methods in synthetic sub-scenario I. Rows (top to bottom) correspond to the five AE methods: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE. Columns (left to right) correspond to: model error (Eq. (148)) between the estimation $\hat{\boldsymbol{\beta}}_{k|k}$ and the true model $\boldsymbol{\beta}_k$, lower is better; test RMSE (Eq. (146)) between predicted labels $\boldsymbol{y}_k^{\text{pred}}$ and test labels $\boldsymbol{y}_k^{\text{test}}$, lower is better; objective value (Eq. (151)), higher is better (maximization problem). Shaded regions represent ± 1 standard deviation from the mean.

M.2.3 Results for Sub-Scenario I

In Fig. 10, we plot the three metrics (model error, test RMSE, and objective value) for the STS, STG, GKS, ADM, and RND design selection methods coupled with the different model estimation policies, as described in Table 5. To avoid clutter, we skip the K variants of these algorithms and report their performance below (see Fig. 12). We observe that our methods STS-AE and STG-AE outperform other methods both in model error and test RMSE. Overall, as the total number of possible candidate features p is higher than the feature dimension d in this setting (see Table 6), selection methods play a more important role. In particular, we observe that STG-SE and ADM-SE outperform RND-AE



(a) STS-AE: Model error (b) STS-AE: Test RMSE (c) STS-AE: Objective

(d) STG-AE: Model error (e) STG-AE: Test RMSE (f) STG-AE: Objective

(g) ADM-AE: Model error (h) ADM-AE: Test RMSE (i) ADM-AE: Objective

(j) GKS-AE: Model error (k) GKS-AE: Test RMSE (l) GKS-AE: Objective

(m) RND-AE: Model error (n) RND-AE: Test RMSE (o) RND-AE: Objective

Figure 13: Unknown vs. known $\boldsymbol{\beta}_*$ comparisons across the five AE methods in synthetic sub-scenario II. Rows (top to bottom) correspond to the five AE methods: STS-AE, STG-AE, ADM-AE, GKS-AE, RND-AE. Columns (left to right) correspond to: model error (Eq. (148)) between the estimation $\hat{\boldsymbol{\beta}}_{k|k}$ and the true model $\boldsymbol{\beta}_k$, lower is better; test RMSE (Eq. (146)) between predicted labels $\boldsymbol{y}_k^{\text{pred}}$ and test labels $\boldsymbol{y}_k^{\text{test}}$, lower is better; objective value (Eq. (151)), higher is better (maximization problem). Shaded regions represent ± 1 standard deviation from the mean.

in all three metrics. Overall, adaptive methods (AE) have better results compared to static (SE), SGD, and forgetting-factor RLS methods, highlighting the importance of estimating the full posterior via Kalman filter equations in Eqs. (5)–(6) for both the model $\hat{\boldsymbol{\beta}}_{k|k}$ and the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ at each stage. Comparing the design selection methods within the same model estimation policies, conducting an optimal design policy (STS and STG) yields a lower error compared to GKS, ADM and RND selections. Moreover, in Fig. 10c, stage-wise and steady state optimal design selection methods tend to have a higher objective value in comparison due to the same reason above, as they find the maximum value of the function (151) optimally. Additionally, the adaptive methods yield a

higher objective value at each stage compared to the static and SGD estimation methods. Figs. 10b and 10c also let us verify empirically the link between our design objective and the metric we report, discussed in Sec. 4.1. At every stage, the ranking of the methods by objective value in Fig. 10c is the reverse of their ranking by test RMSE in Fig. 10b, and we observe no stage at which one method attains both a higher objective value and a higher test RMSE than another. We stress that the objective value of a method is computed from its own posterior covariance estimate, so this agreement is expected only among methods that estimate the covariance consistently. A method whose covariance is badly mis-estimated can report a high objective while predicting poorly, which is why we report the test RMSE rather than the objective as our headline metric in Table 2.

We provide additional results comparing the -K variants of the above methods; we also provide numerical evidence of the convergence of estimator $\hat{\boldsymbol{\beta}}_*$. In Fig. 11, we report the estimation difference $\|\hat{\boldsymbol{\beta}}_* - \boldsymbol{\beta}_*\|_2$ between the estimation $\hat{\boldsymbol{\beta}}_*$ and the true mean fixed-point $\boldsymbol{\beta}_*$, when designs are selected via STS-AE, STG-AE, ADM-AE, GKS-AE, and RND-AE for Sub-Scenario I. For all methods, we observe that as k increases, the estimation difference decreases rapidly; in particular, our estimator $\hat{\boldsymbol{\beta}}_*$ converges to the true mean fixed-point $\boldsymbol{\beta}_*$.

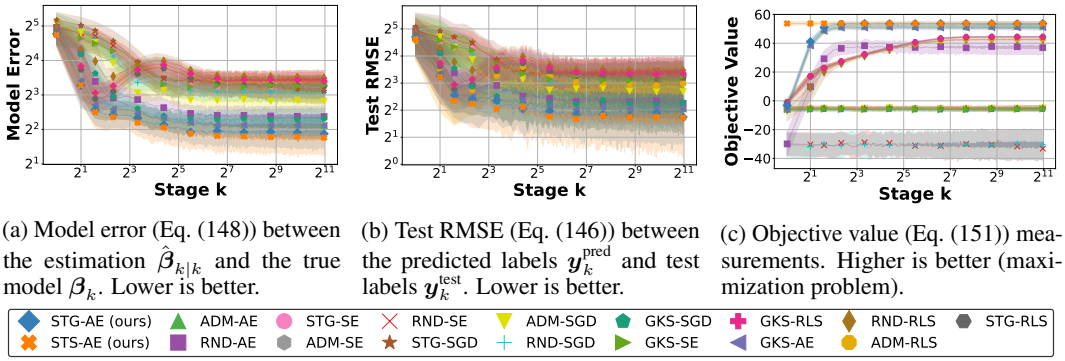
In Fig. 12, we compare AE methods with their K variants, where the real value of the mean fixed-point $\boldsymbol{\beta}_*$ in Eq. (3) is assumed to be known. We report the model error (Eq. (148)), the test RMSE (Eq. (146)), and the objective value (Eq. (151)) in Figs. 12a, 12b, and 12c for the STS-AE method, 12d, 12e, and 12f for the STG-AE method, 12g, 12h, and 12i for the ADM-AE method, 12j, 12k, and 12l for the GKS-AE method, and 12m, 12n, and 12o for the RND-AE method respectively. For all methods, we observe that the AE methods exhibit very similar results to their K variants for all metrics, and that the estimator indeed approximates $\boldsymbol{\beta}_*$ quite well; thus the Euclidean difference between the estimation and the parameter itself converges to 0. Therefore, we justify that for the adaptive methods, we do not have the restriction of the assumption of knowing the true mean fixed-point $\boldsymbol{\beta}_*$.

M.2.4 Results for Sub-Scenario II

In Fig. 14 we plot the three metrics (model error, test RMSE, and objective value) for the STS, STG, GKS, ADM, and RND design selection methods coupled with the different model estimation policies, as described in Table 5. To avoid clutter, we again skip the K variants of these algorithms and report their performance below (see Fig. 13). We observe that our methods STS-AE and STG-AE outperform other methods both in model error and test RMSE. Similar behavior to that in Sub-Scenario I in terms of the superiority of adaptive (-AE) methods is observed in this scenario as well. Our selection methods STS and STG yield a lower error compared to GKS, ADM and RND. Moreover, in Fig. 14c, adaptive methods tend to have a higher objective value in comparison due to the Kalman filter adaptations of the covariance $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{k|k}}$ at each stage, where the STS and STG design selections find the maximum value of the function (151) optimally, yielding a higher objective value at each stage compared to GKS, ADM and RND design selections. As observed in Sub-Scenario I, the objective value and the test RMSE remain consistent with each other in this scenario as well. The ranking of the methods by objective value in Fig. 14c is again the reverse of their ranking by test RMSE in Fig. 14b at every stage, so a design that attains a higher objective also yields a lower test error.

We again provide additional results comparing the -K variants, as well as illustrating the convergence of $\hat{\boldsymbol{\beta}}_*$. Similar to Sub-Scenario I, we plot the convergence of the mean fixed-point estimator evaluated in (8) for AE methods in Fig. 11 for Sub-Scenario II. We report the estimation difference $\|\hat{\boldsymbol{\beta}}_* - \boldsymbol{\beta}_*\|_2$ between the estimation $\hat{\boldsymbol{\beta}}_*$ and the true mean fixed-point $\boldsymbol{\beta}_*$ for STS-AE, STG-AE, ADM-AE, GKS-AE, and RND-AE described in Table 5 in Figs. 11f, 11g, 11h, 11i, and 11j respectively. For all methods, we observe that as k increases, the estimation difference decreases; in particular, our estimator $\hat{\boldsymbol{\beta}}_*$ converges to the true mean fixed-point $\boldsymbol{\beta}_*$.

In Fig. 13, we compare AE methods with their K variants. We again observe that for all metrics, AE methods closely track their K variants, and that the estimator approximates $\boldsymbol{\beta}_*$.



(a) Model error (Eq. (148)) between the estimation $\hat{\boldsymbol{\beta}}_{k|k}$ and the true model $\boldsymbol{\beta}_k$. Lower is better.

(b) Test RMSE (Eq. (146)) between the predicted labels $\boldsymbol{y}_k^{\text{pred}}$ and test labels $\boldsymbol{y}_k^{\text{test}}$. Lower is better.

(c) Objective value (Eq. (151)) measurements. Higher is better (maximization problem).

Figure 14: Performance w.r.t. model error, test RMSE, and objective value for synthetic sub-scenario II as a function of the stage k. Shaded regions represent ± 1 standard deviation from the mean.

N Limitations

Our framework rests on a specific dynamics model in which the parameter $\boldsymbol{\beta}_k$ undergoes Gaussian oscillations around a mean fixed point $\boldsymbol{\beta}_*$ (Eq. (3)). Real-world parameter drift can exhibit richer behavior (level shifts, regime changes, or non-Gaussian innovations) that this model does not capture. The choice is motivated by two considerations: Gaussian oscillations are a standard default when the underlying dynamics are unknown; moreover, deployed model parameters in practice neither diverge nor collapse but rather fluctuate around an operating point, which this model captures compactly. Our modeled dynamics also come with important advantages. To begin with, the resulting DARE characterizing the steady-state covariance (Eq. (10)) admits a closed-form solution, which is what makes Thm. 4 and the steady-state policies analytically tractable. Furthermore, our ability to extend our analysis to the EKF also exploits Eq. (3), as we can linearize the model (i.e., compute the Jacobian) around the mean fixed-point. Extending the framework to richer dynamics, such as regime-switching or heavy-tailed innovations, is left for future work.

Another limitation concerns the non-linear setting. The EKF linearization in Eq. (16) renders the underlying least-squares estimation non-convex, and consequently the mean fixed point $\boldsymbol{\beta}_*$ is not guaranteed to be identifiable in closed form. We approximate it via a "status-quo" estimate that minimizes the squared-error loss while ignoring dynamics; this is an approximation rather than an exact recovery of $\boldsymbol{\beta}_*$. Empirically, the approximation is sufficient: the convergence plots in Fig. 6 show that $\hat{\boldsymbol{\beta}}_*$ tracks the true mean fixed point closely across all four architectures, Table 8 and Figs. 7–8 show that the methods with unknown mean fixed point $\boldsymbol{\beta}_*$ exhibit very similar performance to their variants in which $\boldsymbol{\beta}_*$ is assumed to be known, and the test RMSE results in Table 2 indicate that this is enough to obtain consistently strong downstream performance. Nevertheless, extending guarantees, e.g., when the mean fixed-point is identifiable, is an interesting open direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction are in Section 3, verified with theoretical results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide our limitations in Appendix N.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide proofs for all of our theorems 1, 2, 3, 4, 5, 7, and 8 in Appendices C, D, E, F, H, E.1.1, and K.1 respectively.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the required information for reproduction in Appendix L.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide our source code as mentioned in Section 4: `https://github.com/neu-spiral/MarkovExp`

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide all experimental settings and details in Appendix L.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In all of our numerical evaluations in Section 4 and Appendix M, we report the mean and ± 1 standard deviation over 100 random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide our computing infrastructure in Appendix L.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We read the code of ethics and made every effort to follow it in the paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work proposes a framework to enable data selection through experimental design in the presence of concept drift. There is no societal impact of the work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: We do not release any new datasets or models. We use publicly available datasets as described in Section 4, which do not pose any risk of misuse beyond their existing public availability.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets as described in Section 4 and properly cite them in the paper. We comply with the licenses and usage terms provided by the original creators.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our source code is publicly available at `https://github.com/neu-spiral/MarkovExp`, with detailed instructions and documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: Our paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.