IMPERIAL

19 August 2026

Bayesian experimental design without posterior calculations

An adversarial approach

Anqi Qu, Christophe Muller, Debolina Paul, Firas Darwish, Harleen Gulati, Sofia Gilardini Benitez

Experimental Design

- Selecting a good design for an experiment can be crucial to extracting useful information and controlling costs.

Experimental Design

- Selecting a good design for an experiment can be crucial to extracting useful information and controlling costs.
- Select a design τ, where τ can be seen as a set of design points

$$\tau_1, \ldots, \tau_d.$$

Experimental Design

- Selecting a good design for an experiment can be crucial to extracting useful information and controlling costs.
- Select a design τ, where τ can be seen as a set of design points

$$\tau_1, \ldots, \tau_d.$$

- The experiment produces data y, a vector of n observations, with likelihood

$$f(y \mid \theta; \tau),$$

where θ is a vector of parameters.

Bayesian Experimental Design

- Working in a Bayesian framework, introduce a prior density $\pi(\theta)$ for θ.

Bayesian Experimental Design

- Working in a Bayesian framework, introduce a prior density $\pi(\theta)$ for θ.
- The posterior density is then

$$\pi(\theta \mid y; \tau) = \frac{\pi(\theta) f(y \mid \theta; \tau)}{\pi(y; \tau)}, \qquad \pi(y; \tau) = \mathbb{E}_{\theta \sim \pi(\theta)} \big[f(y \mid \theta; \tau) \big].$$

Bayesian Experimental Design

- Working in a Bayesian framework, introduce a prior density $\pi(\theta)$ for θ.
- The posterior density is then

$$\pi(\theta \mid y; \tau) = \frac{\pi(\theta) f(y \mid \theta; \tau)}{\pi(y; \tau)}, \qquad \pi(y; \tau) = \mathbb{E}_{\theta \sim \pi(\theta)} \big[f(y \mid \theta; \tau) \big].$$

- The Bayesian approach to experimental design involves selecting a function

$$\mathcal{U} = \mathcal{U}(\tau, \theta, y)$$

giving the utility of the chosen design τ given observations y and parameters θ.

Bayesian Experimental Design

- Working in a Bayesian framework, introduce a prior density $\pi(\theta)$ for θ.
- The posterior density is then

$$\pi(\theta \mid y; \tau) = \frac{\pi(\theta) f(y \mid \theta; \tau)}{\pi(y; \tau)}, \qquad \pi(y; \tau) = \mathbb{E}_{\theta \sim \pi(\theta)} \big[f(y \mid \theta; \tau) \big].$$

- The Bayesian approach to experimental design involves selecting a function

$$\mathcal{U} = \mathcal{U}(\tau, \theta, y)$$

  giving the utility of the chosen design τ given observations y and parameters θ.
- Want to obtain a design maximising the expected utility

$$\mathcal{J}(\tau) = \mathbb{E}_{(\theta, y) \sim \pi(\theta, y; \tau)} \big[\mathcal{U}(\tau, \theta, y) \big].$$

Motivation

- One such example of a utility function is the Shannon information gain:

$$\mathcal{U}_{\mathsf{SIG}}(\tau, \theta, y) = \log \pi(\theta \mid y; \tau) - \log \pi(\theta).$$

Motivation

- One such example of a utility function is the Shannon information gain:

$$\mathcal{U}_{\mathsf{SIG}}(\tau, \theta, y) = \log \pi(\theta \mid y; \tau) - \log \pi(\theta).$$

- SIG measures how much the experiment shrinks the uncertainty from the prior to the posterior.

Motivation

- One such example of a utility function is the Shannon information gain:

$$\mathcal{U}_{\text{SIG}}(\tau, \theta, y) = \log \pi(\theta \mid y; \tau) - \log \pi(\theta).$$

- SIG measures how much the experiment shrinks the uncertainty from the prior to the posterior.
- Optimising the design implies posterior calculations need to be repeated for a large number of potential designs and simulated datasets.

Motivation

- One such example of a utility function is the Shannon information gain:

$$\mathcal{U}_{\text{SIG}}(\tau, \theta, y) = \log \pi(\theta \mid y; \tau) - \log \pi(\theta).$$

- SIG measures how much the experiment shrinks the uncertainty from the prior to the posterior.
- Optimising the design implies posterior calculations need to be repeated for a large number of potential designs and simulated datasets.
- This is expensive and prohibits the scaling up of these methods to models with more parameters or designs with many unknowns to select.

Motivation

- One such example of a utility function is the Shannon information gain:

$$\mathcal{U}_{\text{SIG}}(\tau, \theta, y) = \log \pi(\theta \mid y; \tau) - \log \pi(\theta).$$

- SIG measures how much the experiment shrinks the uncertainty from the prior to the posterior.
- Optimising the design implies posterior calculations need to be repeated for a large number of potential designs and simulated datasets.
- This is expensive and prohibits the scaling up of these methods to models with more parameters or designs with many unknowns to select.
- *Bayesian Experimental Design without Posterior Calculations: An Adversarial Approach* (Prangle, Harbisher and Gillespie) presents approaches which avoid the need for posterior inference.

Fisher Information

- The *score function* is the gradient of the log-likelihood:

$$u(y, \theta; \tau) = \nabla_\theta \log f(y \mid \theta; \tau).$$

Fisher Information

- The *score function* is the gradient of the log-likelihood:

$$u(y, \theta; \tau) = \nabla_\theta \log f(y \mid \theta; \tau).$$

- The **Fisher information matrix** (FIM) is its variance – the score has mean zero, so there is no mean term to subtract:

$$I(\theta; \tau) = \mathbb{E}_{y \sim f(y|\theta;\tau)} \left[u\, u^\top \right] = -\mathbb{E}_y \left[\nabla_\theta^2 \log f(y \mid \theta; \tau) \right].$$

Fisher Information

- The *score function* is the gradient of the log-likelihood:

$$u(y, \theta; \tau) = \nabla_\theta \log f(y \mid \theta; \tau).$$

- The **Fisher information matrix** (FIM) is its variance – the score has mean zero, so there is no mean term to subtract:

$$I(\theta; \tau) = \mathbb{E}_{y \sim f(y|\theta;\tau)} \left[u\, u^\top \right] = -\,\mathbb{E}_y \left[\nabla_\theta^2 \log f(y \mid \theta; \tau) \right].$$

- It is the expected *curvature* of the log-likelihood: how sharply the data pin down θ. It approximates the posterior precision.

Fisher Information

- The *score function* is the gradient of the log-likelihood:

$$u(y, \theta; \tau) = \nabla_\theta \log f(y \mid \theta; \tau).$$

- The **Fisher information matrix** (FIM) is its variance – the score has mean zero, so there is no mean term to subtract:

$$I(\theta; \tau) = \mathbb{E}_{y \sim f(y|\theta;\tau)} \left[u\, u^\top \right] = -\mathbb{E}_y \left[\nabla^2_\theta \log f(y \mid \theta; \tau) \right].$$

- It is the expected *curvature* of the log-likelihood: how sharply the data pin down θ. It approximates the posterior precision.
- Crucially, it is often available in **closed form**, with no integration:
  - Poisson: $y \sim \mathrm{Poisson}(\phi) \;\Rightarrow\; I(\phi) = 1/\phi$.
  - Normal: $y \sim N(\mu, \Sigma) \;\Rightarrow\; I(\mu) = \Sigma^{-1}$.

Fisher Information Gain (FIG)

- Maximise the expected *trace* of the FIM:

$$J_{\mathrm{FIG}}(\tau) = \mathbb{E}_{\theta\sim\pi}[\mathrm{tr}\, I(\theta;\tau)] = \mathrm{tr}\, \bar{I}(\tau), \qquad \bar{I}(\tau) = \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

Fisher Information Gain (FIG)

- Maximise the expected *trace* of the FIM:

$$J_{\mathrm{FIG}}(\tau) = \mathbb{E}_{\theta\sim\pi}[\mathrm{tr}\, I(\theta;\tau)] = \mathrm{tr}\, \bar{I}(\tau), \qquad \bar{I}(\tau) = \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

- **The key point:** J_{FIG} involves *no posterior* and *no evidence* $\pi(y;\tau)$ – only the FIM, which is closed form.

Fisher Information Gain (FIG)

- Maximise the expected *trace* of the FIM:

$$J_{\mathrm{FIG}}(\tau) = \mathbb{E}_{\theta\sim\pi}[\mathrm{tr}\, I(\theta;\tau)] = \mathrm{tr}\, \bar{I}(\tau), \qquad \bar{I}(\tau) = \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

- **The key point:** J_{FIG} involves *no posterior* and *no evidence* $\pi(y;\tau)$ – only the FIM, which is closed form.
- Recall SIG needed $\log \pi(y;\tau)$: computational bottleneck.

Why FIG is attractive

- The FIM is closed form and differentiable $\Rightarrow$ cheap gradients via automatic differentiation.

Why FIG is attractive

- The FIM is closed form and differentiable $\Rightarrow$ cheap gradients via automatic differentiation.
- No log-evidence to estimate $\Rightarrow$ standard stochastic gradient optimisation applies directly.

Why FIG is attractive

- The FIM is closed form and differentiable $\Rightarrow$ cheap gradients via automatic differentiation.
- No log-evidence to estimate $\Rightarrow$ standard stochastic gradient optimisation applies directly.
- Scales easily to high-dimensional designs, where posterior-based methods become infeasible.

Why FIG is attractive

- The FIM is closed form and differentiable $\Rightarrow$ cheap gradients via automatic differentiation.
- No log-evidence to estimate $\Rightarrow$ standard stochastic gradient optimisation applies directly.
- Scales easily to high-dimensional designs, where posterior-based methods become infeasible.
- **But there is a price:**

The two main problems with FIG

1. **Not invariant to reparameterisation**
   - The "optimal" design can change if we rewrite the same model using different parameters.
   - This is not good because parameterisation is arbitrary, so the results should not be impacted by reparameterisation.

2. **Can produce intuitively poor designs**
   - FIG maximises the *trace* of the Fisher information matrix.
   - The trace can be large because one parameter direction is learned extremely well,
   - and the other directions may remain almost as uncertain as under the prior.

Problem 1: Reparameterisation

Consider a probability model with parameters θ, and a function

$$\phi = \phi(\theta)$$

producing an alternative vector of parameters ϕ. The new parameter vector ϕ may be shorter or longer than θ.

Let $J(\phi)$ be the Jacobian matrix, with entries

$$J(\phi)_{ij} = \frac{\partial \phi_i}{\partial \theta_j}.$$

Then the Fisher information transforms as

$$I_\theta(\theta) = J(\phi)^T I_\phi(\phi) J(\phi).$$

So Fisher information itself depends on the parameterisation in a structured way.

Problem 1: FIG is not invariant to parameterisation

FIG optimises the expected trace of the Fisher information:

$$J_{\mathrm{FIG}}(\tau) = \mathrm{tr}\left(\bar{I}_\theta(\tau)\right), \qquad \bar{I}_\theta(\tau) = \mathbb{E}_\theta\left[I_\theta(\theta;\tau)\right].$$

For a linear reparameterisation, like $\phi = B\theta$, the Fisher information transforms as

$$I_\phi(\phi;\tau) = B^{-T} I_\theta(\theta;\tau) B^{-1}.$$

So after reparameterisation, FIG becomes

$$J^{\phi}_{\mathrm{FIG}}(\tau) = \mathrm{tr}\left(B^{-T}\bar{I}_\theta(\tau)B^{-1}\right).$$

In general,

$$\mathrm{tr}\left(B^{-T}\bar{I}_\theta(\tau)B^{-1}\right) \neq c\ \mathrm{tr}\left(\bar{I}_\theta(\tau)\right),$$

This is undesirable: the optimal design can depend on the seemingly irrelevant choice of parameterisation.

Problem 2: FIG can reward bad designs

FIG optimises

$$J_{\mathrm{FIG}}(\tau) = \mathrm{tr}\left(\bar{I}(\tau)\right).$$

The eigenvalues of $\bar{I}(\tau)$ are $\lambda_1, \ldots, \lambda_p$, so $\mathrm{tr}\left(\bar{I}(\tau)\right) = \sum_{i=1}^{p} \lambda_i$

The problem: a sum can be large even if only one term is large:

$$\lambda_1 \gg 0, \qquad \lambda_2, \ldots, \lambda_p \approx 0.$$

The Fisher information approximates posterior precision, so this means that one parameter combination can be learnt extremely well while the other combinations are left as uncertain as before.

This can lead to designs with excessive replication, such as placing all observations at one time point. Such a design may nail one quantity, but learn almost nothing about the rest of the model. (We will see an example of this later...)

The Fix: Enter the Adversarial Critic

- To solve both issues simultaneously, the authors introduce a **game theoretic framework**.

The Fix: Enter the Adversarial Critic

- To solve both issues simultaneously, the authors introduce a **game theoretic framework**.
- Instead of static optimization problem, frame experimental design as a two-player game:
  - **The Experimenter** optimizes the design τ.
  - **The Critic** searches the worst parameterisation of θ.

The Fix: Enter the Adversarial Critic

- To solve both issues simultaneously, the authors introduce a **game theoretic framework**.
- Instead of static optimization problem, frame experimental design as a two-player game:
  - **The Experimenter** optimizes the design τ.
  - **The Critic** searches the worst parameterisation of θ.
- **The Rules of the Game:**
  1. The experimenter chooses design τ.
  2. The critic selects an invertible linear transformation matrix A to redefine the parameters: $\phi = A^{-1}\theta$ (constrained to $\det A = 1$).
  3. The critic's reward is exactly the negative of the experimenter's score.

What does the adversarial game optimise?

The experimenter chooses τ; the critic chooses the worst linear reparameterisation

$$\phi = A^{-1}\theta, \qquad \det A = 1.$$

Result 5 shows that the Subgame Perfect Equilibria is equivalent to:

$$\min_{\tau} \max_{A:\det A=1} K(\tau, A), \qquad K(\tau, A) = -\mathbb{E}_{\theta\sim\pi}\left[\operatorname{tr}\left(A^T I(\theta;\tau)A\right)\right].$$

The resulting optimal designs are exactly those maximising

$$J_{\mathrm{ADV}}(\tau) = \det \bar{I}(\tau), \qquad \bar{I}(\tau) = \mathbb{E}_{\theta\sim\pi}\left[I(\theta;\tau)\right].$$

So the adversarial game turns FIG's trace objective into a determinant objective.

Why does the determinant fix FIG?

Let the eigenvalues of $\bar{I}(\tau)$ be

$$\lambda_1, \ldots, \lambda_p.$$

FIG

$$J_{\mathrm{FIG}}(\tau) = \operatorname{tr} \bar{I}(\tau) = \sum_{i=1}^{p} \lambda_i$$

A sum can be large because one eigenvalue is huge:

$$\lambda_1 \gg 0, \qquad \lambda_2, \ldots, \lambda_p \approx 0.$$

So FIG may learn one parameter direction and neglect the rest.

ADV

$$J_{\mathrm{ADV}}(\tau) = \det \bar{I}(\tau) = \prod_{i=1}^{p} \lambda_i$$

A product collapses if any eigenvalue is near zero:

$$\lambda_j \approx 0 \quad \Rightarrow \quad J_{\mathrm{ADV}}(\tau) \approx 0.$$

So ADV forces information in every parameter direction.

Intuition: if the design has a weak direction, the critic finds it.

From determinant to minimax

- Target:

$$\mathcal{J}_{\mathrm{ADV}}(\tau) \;=\; \det \bar{I}(\tau), \qquad \bar{I}(\tau) \;=\; \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

From determinant to minimax

- Target:

$$\mathcal{J}_{\mathrm{ADV}}(\tau) \;=\; \det \bar{I}(\tau), \qquad \bar{I}(\tau) \;=\; \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

- **Difficulty.** The determinant is a *nonlinear* functional of $\bar{I}(\tau)$. Plugging in a Monte Carlo estimate gives a *biased* gradient: for prior samples $\theta^{(1)},\ldots,\theta^{(K)}$,

$$\mathbb{E}\left[\nabla_\tau \det\left(\tfrac{1}{K}\textstyle\sum_{k=1}^{K} I(\theta^{(k)};\tau)\right)\right] \;\neq\; \nabla_\tau \det \bar{I}(\tau).$$

From determinant to minimax

- Target:

$$\mathcal{J}_{\mathrm{ADV}}(\tau) \;=\; \det \bar{I}(\tau), \qquad \bar{I}(\tau) \;=\; \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

- **Difficulty.** The determinant is a *nonlinear* functional of $\bar{I}(\tau)$. Plugging in a Monte Carlo estimate gives a *biased* gradient: for prior samples $\theta^{(1)},\ldots,\theta^{(K)}$,

$$\mathbb{E}\Big[\nabla_\tau \det\Big(\tfrac{1}{K}\textstyle\sum_{k=1}^{K} I(\theta^{(k)};\tau)\Big)\Big] \;\neq\; \nabla_\tau \det \bar{I}(\tau).$$

- **Result 5.** Optimal designs equivalently solve

$$\min_{\tau} \max_{A:\,\det A=1} K(\tau, A), \quad K(\tau, A) = -\,\mathbb{E}_{\theta\sim\pi}\Big[\mathrm{tr}\Big(A^\top I(\theta;\tau)\,A\Big)\Big]\,.$$

From determinant to minimax

- Target:

$$\mathcal{J}_{\mathrm{ADV}}(\tau) \;=\; \det \bar{I}(\tau), \qquad \bar{I}(\tau) \;=\; \mathbb{E}_{\theta\sim\pi}[I(\theta;\tau)]\,.$$

- **Difficulty.** The determinant is a *nonlinear* functional of $\bar{I}(\tau)$. Plugging in a Monte Carlo estimate gives a *biased* gradient: for prior samples $\theta^{(1)},\ldots,\theta^{(K)}$,

$$\mathbb{E}\Big[\nabla_\tau \det\Big(\tfrac{1}{K}\textstyle\sum_{k=1}^{K} I(\theta^{(k)};\tau)\Big)\Big] \;\neq\; \nabla_\tau \det \bar{I}(\tau).$$

- **Result 5.** Optimal designs equivalently solve

$$\min_{\tau} \max_{A:\,\det A=1} K(\tau, A), \quad K(\tau, A) = -\,\mathbb{E}_{\theta\sim\pi}\Big[\mathrm{tr}\Big(A^\top I(\theta;\tau)\,A\Big)\Big]\,.$$

- $K(\tau, A)$ is an *expectation* of a linear functional of $I(\theta;\tau)$ $\Longrightarrow$ unbiased Monte Carlo gradients are immediate.

Tractable gradients and the constraint $\det A = 1$

Unbiased estimator (Result 6). With $\theta^{(1)}, \dots, \theta^{(K)} \overset{\text{i.i.d.}}{\sim} \pi$,

$$\hat{K}(\tau, A) \;=\; -\operatorname{tr}\left[A^\top \left(\frac{1}{K}\sum_{k=1}^{K} I(\theta^{(k)}; \tau)\right) A\right], \quad \mathbb{E}[\nabla \hat{K}] \;=\; \nabla K.$$

Enforcing $\det A = 1$**.** By the cyclic property of trace, K depends on A only through $AA^\top$. Restrict A to lower-triangular Cholesky matrices with unit determinant, parameterised by unconstrained $\eta \in \mathbb{R}^{p(p+1)/2-1}$:

$$A(\eta) \;=\; \begin{pmatrix} e^{\eta_{11}} & 0 & \cdots & 0 \\ \eta_{21} & e^{\eta_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{p1} & \eta_{p2} & \cdots & \exp\Big(-\sum_{i=1}^{p-1} \eta_{ii}\Big) \end{pmatrix}.$$

Initialise $\eta = 0$ so $A(\eta) = I$: the critic starts at the trivial reparameterisation.

Gradient descent ascent (GDA)

To solve $\min_\tau \max_\eta K(\tau, A(\eta))$ the authors use the general GDA scheme: simultaneous descent on the minimising variable x and ascent on the maximising variable y, with unbiased gradient estimates at each step.

Algorithm 1 Gradient descent ascent (GDA)

1: **Input:** initial values x_1, y_1; update subroutines h_x, h_y.
2: **for** $t = 1, 2, \ldots$ **do**
3: Compute $g_{x,t}, g_{y,t}$, unbiased estimates of $-\nabla_x f(x_t, y_t)$ and $\nabla_y f(x_t, y_t)$.
4: Update: $x_{t+1} = x_t + h_x(g_{x,t}), \quad y_{t+1} = y_t + h_y(g_{y,t})$.
5: **end for**

- Reduces to stochastic gradient descent when f depends on x alone.
- Update rules h_x, h_y: e.g. a default rule $z_{t+1} = z_t + \alpha_{z,t}\, g_{z,t}$ with diminishing $\alpha_{z,t}$, or adaptive schemes such as Adam.
- Minimax convergence is subtle: limit cycles can occur, so two-timescale update rule $\alpha_{y,t}/\alpha_{x,t} \to \infty$ (or a similar condition under Adam) and diagnostics are used.

Main algorithm: GDA for Bayesian experimental design

Specialising Algorithm 1 to $f(\tau, \eta) = K(\tau, A(\eta))$ and running R replications in parallel from different initial designs:

Algorithm 2 GDA for Bayesian experimental design

1: **Input:** number of prior samples K; parallel replications R; initial values τ_1^i, η_1^i for $i = 1, \ldots, R$; update subroutines h_τ, h_η.
2: **for** $t = 1, 2, \ldots$ **do**
3: Sample $\theta^{(k)} \sim \pi(\theta)$ for $k = 1, 2, \ldots, K$.
4: Compute $g_{\tau,t}^i$, $g_{\eta,t}^i$, unbiased estimates of $-\nabla_\tau K(\tau_t^i, A(\eta_t^i))$ and $\nabla_\eta K(\tau_t^i, A(\eta_t^i))$ via automatic differentiation of $\hat{K}$, for all i.
5: Update: $\tau_{t+1}^i = \tau_t^i + h_\tau(g_{\tau,t}^i)$, $\eta_{t+1}^i = \eta_t^i + h_\eta(g_{\eta,t}^i)$, for all i.
6: **end for**

- Implemented in PyTorch with the Adam update rule; the same $\theta^{(k)}$ samples are reused across the R runs, giving efficient parallelisation.
- R random initialisations help escape local optima; *point exchange* post-processing then refines cluster sizes when designs replicate.

Pharmacokinetics Example

The Problem

- A subject receives a dose of a drug, and we can choose 15 observation times to measure the drug concentration.

Pharmacokinetics Example

The Problem

- A subject receives a dose of a drug, and we can choose 15 observation times to measure the drug concentration.
- **Question:** When should we take 15 blood samples over 24 hours?

Pharmacokinetics Example

The Problem

- A subject receives a dose of a drug, and we can choose 15 observation times to measure the drug concentration.
- **Question:** When should we take 15 blood samples over 24 hours?
- **Goal:** Choose sampling times that help us learn the drug concentration curve's unknown parameters as accurately as possible.

Pharmacokinetics Example

The Model

- We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2)$$

Pharmacokinetics Example

The Model

- We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2)$$

- $x(\theta, \tau_i) = \frac{D\theta_2(\exp[-\theta_1\tau_i] - \exp[-\theta_2\tau_i])}{\theta_3(\theta_2 - \theta_1)}$

Pharmacokinetics Example

The Model

- We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2)$$

- $x(\theta, \tau_i) = \frac{D\theta_2(\exp[-\theta_1\tau_i] - \exp[-\theta_2\tau_i])}{\theta_3(\theta_2 - \theta_1)}$
- σ^2=0.1 is known observation noise and $D = 400$

Pharmacokinetics Example

The Model

- We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2)$$

- $x(\theta, \tau_i) = \frac{D\theta_2(\exp[-\theta_1\tau_i] - \exp[-\theta_2\tau_i])}{\theta_3(\theta_2 - \theta_1)}$
- σ^2=0.1 is known observation noise and $D = 400$
- We assume independent log normal priors

$$\theta_1 \sim LN(\log 0.1, 0.05)$$

$$\theta_2 \sim LN(\log 1, 0.05)$$

$$\theta_3 \sim LN(\log 20, 0.05)$$

Pharmacokinetics Example

The Model

- We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2)$$

- $x(\theta, \tau_i) = \frac{D\theta_2(\exp[-\theta_1\tau_i] - \exp[-\theta_2\tau_i])}{\theta_3(\theta_2 - \theta_1)}$
- σ^2=0.1 is known observation noise and $D = 400$
- We assume independent log normal priors

$$\theta_1 \sim LN(\log 0.1, 0.05)$$

$$\theta_2 \sim LN(\log 1, 0.05)$$

$$\theta_3 \sim LN(\log 20, 0.05)$$

- We aim to find $(\tau_1, \ldots, \tau_{15})$ in $[0, 24]$

Pharmacokinetics Example

What are we trying to learn?

- The unknown parameters

$$\theta = (\theta_1, \theta_2, \theta_3)$$

Pharmacokinetics Example

What are we trying to learn?

- The unknown parameters

$$\theta = (\theta_1, \theta_2, \theta_3)$$

- They control the shape of the concentration-time curve: rise, peak, decay, and scale.

Pharmacokinetics Example

Applying the algorithm

- Perform 100 replications in parallel based on 100 initial designs sampled from a uniform distribution over $[0, 24]^{15}$.

Algorithm 3 ADV Design

1: **repeat**
2: Sample parameter $\theta \sim p(\theta)$ from the prior
3: Compute the Fisher information $I(\theta; \tau)$
4: Update design variables τ_i to improve the design
5: Update A
6: **until** convergence or stopping criterion is met

Pharmacokinetics Example

Applying the algorithm

- Perform 100 replications in parallel based on 100 initial designs sampled from a uniform distribution over $[0, 24]^{15}$.

Algorithm 4 ADV Design

1: **repeat**
2: Sample parameter $\theta \sim p(\theta)$ from the prior
3: Compute the Fisher information $I(\theta; \tau)$
4: Update design variables τ_i to improve the design
5: Update A
6: **until** convergence or stopping criterion is met

- Parameter θ sampled once, although sampling more and averaging leads to more stable gradients during optimisation.

Pharmacokinetics Example

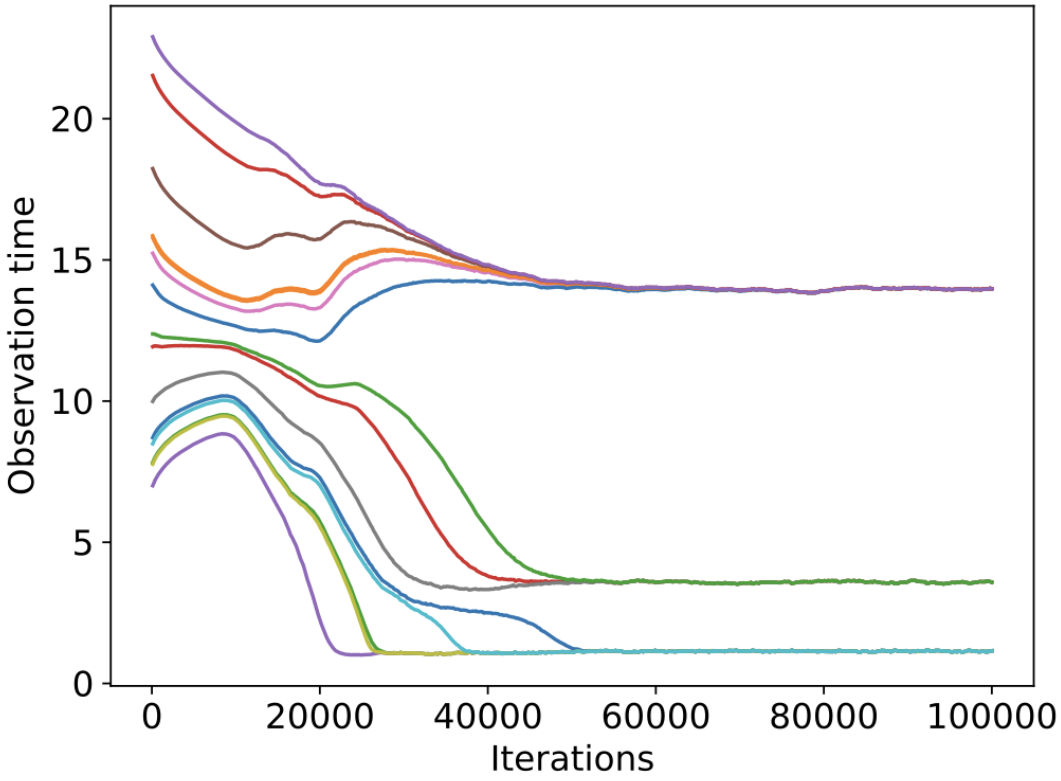

Figure 1: Observation times converging for GDA on the pharmacokinetic example.

Reduce measurement noise rather than sample everywhere.

Pharmacokinetics Example

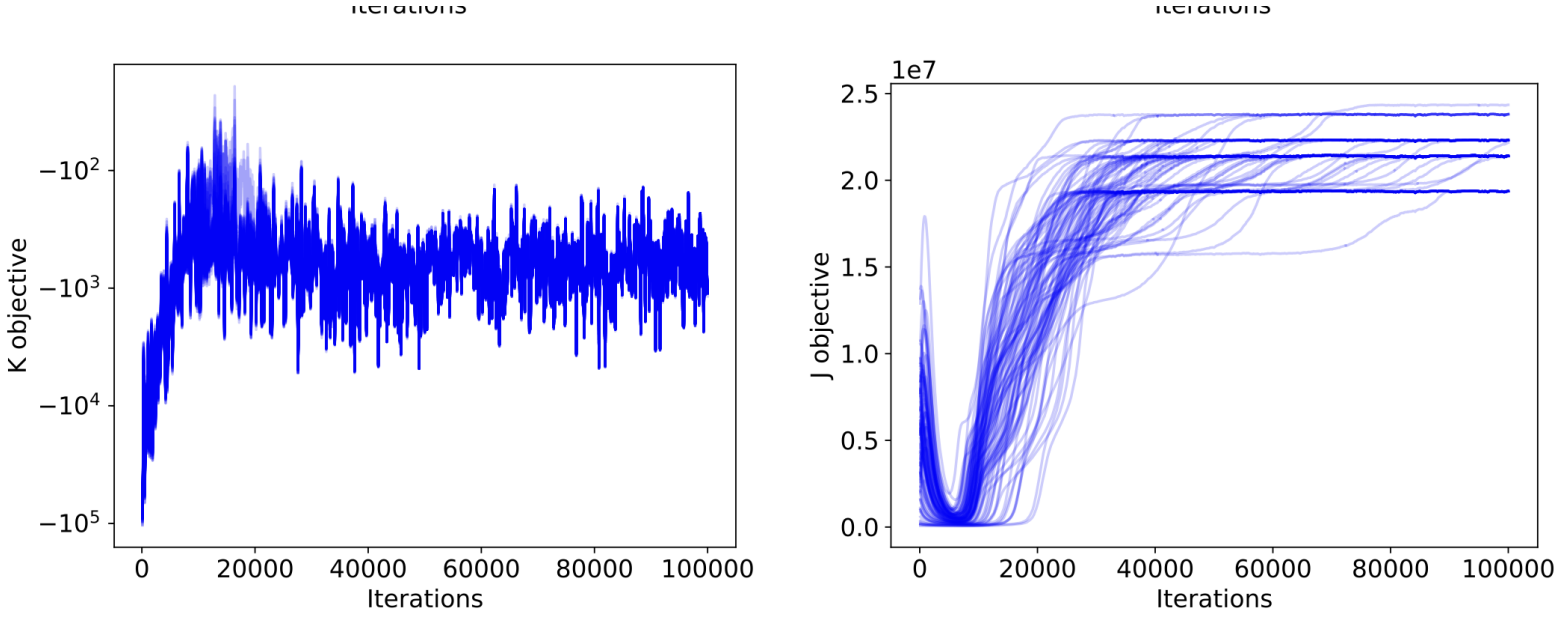

Figure 2: Objective convergence for GDA on the pharmacokinetic example.

K objective typically rises and falls before converging as the critic is also changing it. $J_{ADV}(\tau)$ tracks the design quality.

Pharmacokinetics Example

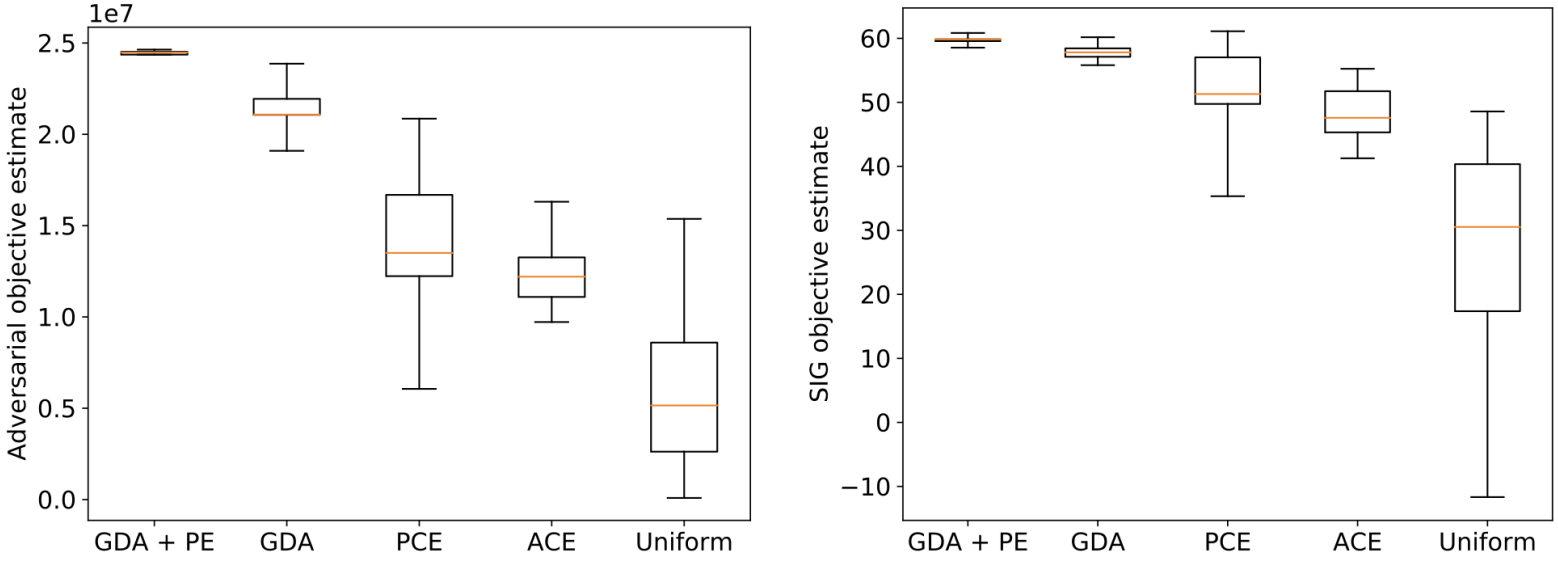

Figure 3: Performance of pharmacokinetic example designs output by various methods. Performance is judged by estimate objective value achieved for (left) the adversarial approach and (right) the Shannon information gain approach.

Point exchange (PE) is a post-processing step that tries swapping individual design points between clusters to improve the objective.

Pharmacokinetics Example

	GDA	GDA+PE	SGD	SGD+PE	ACE	PCE
Mean time (seconds)	1.4	2.2	1.4	1.5	8012	36
Number of repetitions	100	100	100	100	30	100

Figure 4: Mean times to run optimisation methods on the pharmacokinetic example.

- This optimisation method to find Adversial and Fisher Information Gain designs faster than Shannon Information Gain optimisation methods (by a factor of around 10).

References

Dennis Prangle, Sophie Harbisher, and Colin S Gillespie. Bayesian experimental design without posterior calculations: an adversarial approach, 2021. URL `https://arxiv.org/abs/1904.05703`.