Subdata Selection: A Unified Framework for Optimal Selection and Statistical Efficiency Assessment

Min Yang
Department of Mathematics, Statistics, and Computer Science
University of Illinois at Chicago
Chicago, IL 60607
myang2@uic.edu

Wei Zheng
Department of Business Analytics and Statistics
University of Tennessee
Knoxville, Tennessee 37996
wzheng9@utk.edu

John Stufken
Department of Statistics
George Mason University
Fairfax, Virginia 22030
jstufken@gmu.edu

Ming-Chung Chang
Institute of Statistical Science
Academia Sinica
Taipei 115201, Taiwan
mcchang@stat.sinica.edu.tw

Ting Tian
School of Mathematics
Sun Yat-sen University
Guangdong, China 510275
tiant55@mail.sysu.edu.cn

Xueqin Wang
School of Management
University of Science and Technology of China
Anhui, China, 230052
wangxq20@ustc.edu.cn

Abstract

When labeling is expensive or datasets exceed computing capacity, selecting an informative subset of data is a practical necessity. Identifying such subdata is a classical NP-hard problem due to its inherent discreteness. While many subdata selection methods have been proposed for parametric models, two fundamental challenges remain: first, how can we accurately assess the statistical efficiency of selected subdata relative to the theoretical optimum? Second, given N data

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

points and a subdata size n, which n points should be selected to achieve high statistical efficiency? We address both challenges within a unified framework. We develop a new information-based subdata selection methodology grounded in optimal approximate design theory, yielding subdata that approaches the theoretical optimum. Our algorithm is general, accommodates arbitrary N and n, supports multiple optimality criteria, and is accompanied by a convergence proof. Crucially, our framework provides, for the first time, tight lower and upper bounds on the statistical efficiency of subdata selected by any method, enabling accurate and rigorous benchmarking across the literature. This benchmarking capability reveals the true performance landscape of existing methods and shows that many of them have been operating substantially below their theoretical potential. The subdata produced by our methodology is highly efficient and outperforms all existing methods.

1 Introduction

Subdata selection is a rapidly growing area of research [16], driven in part by the increasing size of datasets in terms of the number of observations. It is also relevant for smaller datasets, particularly when they are unlabeled and labeling is expensive, requiring the labeling of only selected data points [23]. For larger datasets, the trade-off between statistical efficiency and computational cost must be considered, while for smaller datasets, the focus is primarily on statistical efficiency [26]. In both scenarios, the challenge lies in selecting n observations from a dataset with N observations, where N is much larger than n, while retaining the maximum amount of information. We will refer to the original dataset of size N as *full data* and to the selected dataset of size n as *subdata*.

Several approaches have been studied for subdata selection. For example, the coreset approach aims to select subdata that produces results close to those obtained from the full data [7]. Another approach, aimed at selecting subdata with a distribution similar to that of the full data, is the support points method [15]. A third approach assumes a parametric model for the full data and selects subdata with the primary objective of efficient parameter estimation [24], which is the approach we focus on in this manuscript.

Identifying such subdata is a classical NP-hard problem due to its inherent discreteness. While many subdata selection methods have been proposed for parametric models, two fundamental challenges remain: first, how can we accurately assess the statistical efficiency of a selected subdata relative to the theoretical optimum? Second, given N data points and a subdata size n, which n points should be selected to achieve high statistical efficiency?

We address both challenges within a unified framework. For the selection challenge, we develop a novel algorithm grounded in optimal approximate design theory that accommodates general parametric models, arbitrary choices of N and n, and multiple optimality criteria, and for which we establish convergence. For the assessment challenge, our framework provides, for the first time, tight lower and upper bounds on the statistical efficiency of subdata selected by any method, enabling accurate and rigorous benchmarking across the literature. These two contributions are deeply connected: the statistical efficiency bounds not only validate our own methodology but reveal that many existing methods have been operating substantially below their theoretical potential. The subdata produced by our methodology is highly efficient and outperforms all existing approaches.

In Section 2 we will provide background information, which is followed in Section 3 with details about our new approach, including theoretical properties and a description of the algorithm that makes it work. In Section 4 we demonstrate the efficiency of our method through simulation studies and make comparisons to alternative methods. We conclude with a brief discussion in Section 5. A brief introduction to optimal design theory for readers who are less familiar with this literature, proofs, more details of the algorithm, and details of the numerical studies are relegated to the appendix.

2 Preliminaries

In this section, we formulate the problem, connect it to concepts from optimal design of experiments, and motivate why we opt for the information-based approach.

For the ith observation, $i = 1, \ldots, N$, write $(y_i, \boldsymbol{x}_i)$, where y_i and $\boldsymbol{x}_i$ denote the label and $p \times 1$ vector of features, respectively. In settings such as active learning, where, for example, labeling

an image requires an expensive expert assessment, the labels y_i may not yet be available, and the goal is to select which data points to label. Since subdata selection relies only on the feature vectors $\boldsymbol{x}_i$, labels are not required at the selection stage. We assume that the relationship between y_i and $\boldsymbol{x}_i$ can be described by a parametric model for which the Fisher information matrix (or a suitable surrogate) can be computed, with parameter vector $\boldsymbol{\theta}$, and that inferences about $\boldsymbol{\theta}$ are based on its Maximum Likelihood Estimator (MLE). While we primarily consider linear and generalized linear models (GLMs), the methodology applies more broadly, as demonstrated by the clusterwise linear regression example in Section 4, which is neither a standard linear model nor a GLM.

When the number of observations N is extraordinarily large, analyzing the entire dataset can be infeasible or too expensive. Alternatively, when y_i's are unavailable, such as in an active learning setup, we may need to select a subset of $\boldsymbol{x}_i$'s for which we collect the y_i's if labeling is expensive. In both cases, we need to select subdata from the full data and use the subdata for inference. Using indicator variables δ_i, with $\delta_i = 1$ if and only if the ith observation belongs to the selected subdata, we want to identify $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)$, subject to $\sum_{i=1}^{N} \delta_i = n$, where n is the subdata size, such that inferences based on the subdata are optimal in a way to be specified.

We will use the Fisher information matrix for $\boldsymbol{\theta}$, $I_{\boldsymbol{\theta}}$, to evaluate subdata performance. Let $I_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ represent the information matrix for $\boldsymbol{\theta}$ for the ith data point. Assuming independence, the information matrix for $\boldsymbol{\theta}$ using the subdata is given by

$$I_{\boldsymbol{\theta}} = \sum_{i=1}^{N} \delta_i I_{\boldsymbol{\theta}}(\boldsymbol{x}_i). \tag{1}$$

Let $\hat{\boldsymbol{\theta}}$ be the MLE of $\boldsymbol{\theta}$, which is assumed without loss of generality to be unique for the selected subdata. Since $\mathrm{Var}(\hat{\boldsymbol{\theta}})$ is proportional to $I_{\boldsymbol{\theta}}^{-1}$, we aim for subdata that, in some sense, minimizes $I_{\boldsymbol{\theta}}^{-1}$. When interest lies in only a subset of the parameters, the objective is correspondingly to minimize the relevant submatrix of $I_{\boldsymbol{\theta}}^{-1}$. We borrow the approach from optimal design of experiments [11] by minimizing $\Phi(I_{\boldsymbol{\theta}})$ for a convex function Φ. Two popular choices are $\Phi(I_{\boldsymbol{\theta}}) = \log(\det(I_{\boldsymbol{\theta}}^{-1}))$ and $\Phi(I_{\boldsymbol{\theta}}) = \mathrm{Tr}(I_{\boldsymbol{\theta}}^{-1})$, corresponding to D- and A-optimality, respectively. Thus, for some Φ, we seek subdata with indicator $\boldsymbol{\delta}^{opt}$ so that

$$\boldsymbol{\delta}^{opt} = \arg\min_{\boldsymbol{\delta}} \Phi(I_{\boldsymbol{\theta}}). \tag{2}$$

For ease of presentation, we will focus on the A- and D-optimality criteria. However, the results also hold for other optimality criteria, such as the Φ_p-optimality criteria defined in [28], provided that $p < \infty$; here, p denotes the index of the optimality criterion and should not be confused with the dimension of the feature vector. The proofs for the Φ_p-optimality criteria are essentially the same with some additional notations.

These criteria have direct interpretations in terms of estimation error. Under standard regularity conditions, $\hat{\boldsymbol{\theta}}$ is approximately normally distributed with a covariance matrix proportional to $I_{\boldsymbol{\theta}}^{-1}$ (exactly so for a linear model with normal errors). Hence $\mathrm{Tr}(I_{\boldsymbol{\theta}}^{-1})$ is proportional to $\sum_j \mathrm{Var}(\hat{\theta}_j) = E\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2$, the expected squared Euclidean distance between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$, while $\det(I_{\boldsymbol{\theta}}^{-1})^{1/2}$ is proportional to the volume of the confidence ellipsoid for $\boldsymbol{\theta}$. Thus, A-optimal subdata minimizes the total variance of the parameter estimates, and D-optimal subdata minimizes the volume of the confidence ellipsoid; when only some parameters are of interest, the same interpretations hold for the relevant submatrix of $I_{\boldsymbol{\theta}}^{-1}$. Since the statistical efficiencies used in this paper are ratios of criterion values (see Section 3), they inherit this interpretation: roughly speaking, subdata of size n with efficiency e is as informative as optimal subdata of size en. Our methodology relies on two further concepts from optimal design theory, approximate designs and the General Equivalence Theorem, which are introduced in Section 3. For readers who are less familiar with the optimal design literature, Appendix A provides a self-contained introduction to the statistical model, the information matrix, optimality criteria and their connection to estimation error, approximate designs, and the General Equivalence Theorem.

Remark 1. *This formulation establishes a general framework. For example, the subdata selection considered in [26] where $I_{\boldsymbol{\theta}}(\boldsymbol{x}_i) = \boldsymbol{x}_i \boldsymbol{x}_i^T$ can be viewed as subdata selection for the linear first-order model with i.i.d. normally distributed errors under A-optimality.*

With $\binom{N}{n}$ different subdata choices, finding an exact solution to this minimization problem is computationally hard, even for modest values of N and n. Using a result from [27], we show that a polynomial time algorithm does not exist for D-optimality, i.e., it is an NP-hard problem.

Theorem 1. *D-optimal exact subdata selection is NP-hard.*

While Theorem 1 is for D-optimality, we expect the result to hold for most other optimality criteria as well. In view of Theorem 1, we propose a new polynomial time algorithm for selecting highly statistically efficient subdata.

3 The proposed method

We consider the following continuous optimization problem, which is a relaxation of (2):

$$\xi^* = \underset{\xi}{\operatorname{argmin}}\, \Phi(\xi), \tag{3}$$

where $\xi = \{(\boldsymbol{x}_i, w_i), i = 1, \ldots, N\}$ with $0 \leq w_i \leq 1/n$ and $\sum_{i=1}^N w_i = 1$, and $\Phi(\xi) = \Phi\left(\sum_{i=1}^N w_i I_{\boldsymbol{\theta}}(\boldsymbol{x}_i)\right)$. Here, ξ can be interpreted as a bounded approximate design. A key tool in this framework is the general equivalence theorem, which provides necessary and sufficient conditions for optimality under criteria such as A- and D-optimality, and enables efficient identification of optimal designs.

For GLMs, $I_{\boldsymbol{\theta}}$ depends on $\boldsymbol{\theta}$. This creates a circular problem in which optimizing Equation (3) over ξ is to find a design that optimizes inference for $\boldsymbol{\theta}$, but the function to be optimized depends on the unknown $\boldsymbol{\theta}$. We will return to this in Subsection C.3, but for simplicity we assume for now that $I_{\boldsymbol{\theta}}$ is entirely known, as is the case for linear models.

To find an optimal bounded approximate design, we will need an equivalence theorem for bounded approximate designs. To find subdata of size n from an optimal bounded approximate design ξ^*, we will take all x_i's with weight $1/n$, and supplement this set with points that have the next largest weights in ξ^*, until we have a total of n points. This works because in an optimal bounded approximate design, there are at least n points with a nonzero weight.

For an optimal bounded approximate design ξ^*, let S^* denote the subdata of size n obtained from ξ^* as described in the previous paragraph. For arbitrary subdata S of size n, we use the same notation S for the bounded approximate design, which assigns weight $1/n$ to points in the subdata S and 0 to all other points. Furthermore, let S_{opt} denote the optimal subdata, which is usually unknown. Then

$$\Phi(\xi^*) \leq \Phi(S_{\text{opt)}} \leq \Phi(S^*). \tag{4}$$

The statistical efficiency of the arbitrary subdata S can then be defined as $\text{Eff}(S) = \frac{\Phi(S_{\text{opt}})}{\Phi(S)}$. Although this is unknown since we don't know S_{opt}, we can find an upper and lower bound for this efficiency by using Equation (4):

$$\frac{\Phi(\xi^*)}{\Phi(S)} \leq \text{Eff}(S) \leq \frac{\Phi(S^*)}{\Phi(S)} = \frac{\Phi(\xi^*)}{\Phi(S)} \Big/ \frac{\Phi(\xi^*)}{\Phi(S^*)}\,. \tag{5}$$

Remark 2. *In the empirical studies of Section 4, we observe that S^* is highly efficient, with a statistical efficiency lower bound $\frac{\Phi(\xi^*)}{\Phi(S^*)}$ of at least 99.99% for large N and n. Consequently, the lower and upper bounds for Eff(S) are nearly identical, yielding a tight and accurate assessment of the efficiency of any subdata selection method.*

We point out that when computing the statistical efficiency for D-optimality, we use the commonly accepted form $\Phi(I_{\boldsymbol{\theta}}) = \det(I_{\boldsymbol{\theta}}^{-1})^{1/p_1}$, where p_1 is the dimension of $I_{\boldsymbol{\theta}}$. The form $\Phi(I_{\boldsymbol{\theta}}) = \log(\det(I_{\boldsymbol{\theta}}^{-1}))$ given in Section 2 is however used for the Equivalence Theorem in Subsection 3.1.

Since all of the above hinges on finding the optimal bounded approximate design ξ^*, we now turn to the theorem and algorithm that will facilitate this.

3.1 The Equivalence Theorem for bounded approximate designs

Before presenting the first theorem, we define the class of optimality criteria considered in this paper. The optimality criterion Φ must be convex and linearly differentiable. The latter means that for

approximate designs ξ and $\eta = \{(\boldsymbol{x}_i, \lambda_i), i = 1, ..., N\}$ it holds that

$$F_\Phi(\xi, \eta) = \sum_{i=1}^{N} \lambda_i F_\Phi(\xi; \boldsymbol{x}_i), \text{ where } F_\Phi(\xi, \eta) = \lim_{\alpha \downarrow 0} \frac{\Phi((1-\alpha)\xi + \alpha\eta) - \Phi(\xi)}{\alpha}$$

is the directional derivative of Φ at ξ in the direction of η. Also, $F_\Phi(\xi; \boldsymbol{x}_i)$ is the directional derivative of Φ at ξ in the direction of the Dirac measure $\delta_{\boldsymbol{x}_i}$. An additional property that we will use is that for designs ξ_1, ξ_2 and ξ_3, for all positive values γ for which $\xi = \xi_1 + \gamma(\xi_2 - \xi_3)$ is again a design, it holds that $\Phi(\xi)$ is infinitely differentiable in γ. The class of Φ_p-optimality criteria for finite p (cf. [28]), which includes D- and A-optimality, meets these conditions. In the statements of the theorems, we will use Λ for the class of optimality criteria that satisfy the above conditions.

The following theorem is adapted from [22]. A major difference is that the design space $\mathcal{X}$ is discrete in our case and continuous in [22]. In terms of weight constraints, the formulation of the theorem, which uses the lower bound ν_i and the upper bound μ_i for the weight w_i is more general than we will need. For our application $\nu_i = 0$ and $\mu_i = 1/n$ for all $i = 1, \ldots, N$.

Theorem 2. *(Equivalence Theorem for bounded approximate designs) Let $\{\nu_1, \ldots, \nu_N\}$ and $\{\mu_1, \ldots, \mu_N\}$ be two sets of numbers so that $0 \leq \nu_i < \mu_i$, $\sum_{i=1}^N \nu_i \leq 1$, and $\sum_{i=1}^N \mu_i \geq 1$. Let Ξ be the set of all bounded approximate designs $\{(\boldsymbol{x}_i, w_i), i = 1, \ldots, N\}$ with $\nu_i \leq w_i \leq \mu_i$ and $\sum_{i=1}^N w_i = 1$, and let $\Phi \in \Lambda$. The following two statements are equivalent:*

(I) $\xi^* = \{(\boldsymbol{x}_i, w_i^*), i = 1, \ldots, N\}$ is Φ-optimal in Ξ.

(II) There is a partition of the design space, $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$, and a number s such that

(a) $w_i^* = \nu_i$ for $\boldsymbol{x}_i \in \mathcal{X}_1$ and $w_i^* = \mu_i$ for $\boldsymbol{x}_i \in \mathcal{X}_3$;

(b) If $\mathcal{X}_2$ is empty, then $\max_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i) \leq \min_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)$; and

(c) If $\mathcal{X}_2$ is nonempty, then there is a number s so that for every $\boldsymbol{x}_i \in \mathcal{X}_2$

(i) $\nu_i < w_i^* < \mu_i$;

(ii) $F_\Phi(\xi^*; \boldsymbol{x}_i) = s$; and

(iii) $\max_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i) \leq s \leq \min_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)$.

A proof of Theorem 2, inspired by the proof in [22], is given in the supplementary material. It can be shown that if (I) in Theorem 2 holds, then the value of s in (II)(c) is given by

$$s = \frac{-\sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i - \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i}{1 - (\sum_{\boldsymbol{x}_i \in \mathcal{X}_1} \nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} \mu_i)}. \tag{6}$$

As already noted, for the subdata selection problem, we have $\nu_i = 0$ and $\mu_i = 1/n$, $i = 1, \ldots, N$.

3.2 Identifying an optimal bounded approximate design: The main algorithm

As observed just prior to Subsection 3.1, the proposed methodology depends on identifying a Φ-optimal bounded approximate design ξ^*. Theorem 2, with $\nu_i = 0$ and $\mu_i = 1/n$, $i = 1, \ldots, N$, will guide us in finding such a design via an iterative algorithm.

For an index t, during the algorithm, we will have a partition $\mathcal{P}^{(t)} = (\mathcal{X}_1^{(t)}, \mathcal{X}_2^{(t)}, \mathcal{X}_3^{(t)})$ of the design space $\mathcal{X}$ with weights $w_i^{(t)}$. As in Theorem 2, $\mathcal{X}_1^{(t)}$ and $\mathcal{X}_3^{(t)}$ contain the points in $\mathcal{X}$ with weights 0 and $1/n$, respectively, for a bounded approximate design $\xi_{\mathcal{P}^{(t)}}$ at that stage of the algorithm, while $\mathcal{X}_2^{(t)}$ contains the points, if any, with optimally selected weights that are strictly between these two extremes. The following lemma provides an interesting property for these types of designs.

Lemma 1. *For $\Phi \in \Lambda$, consider a design $\xi_{\mathcal{P}}$ corresponding to a partition $\mathcal{P} = (\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ in which the weights for points in $\mathcal{X}_2$, if any, have been optimized and are strictly between 0 and $1/n$. If $\mathcal{X}_2 \neq \emptyset$, then $F_\Phi(\xi_{\mathcal{P}}; \boldsymbol{x}_i)$ is constant for $\boldsymbol{x}_i \in \mathcal{X}_2$.*

Note that part (c) of Theorem 2 makes this statement already for a Φ-optimal design ξ^*.

We present pseudocode for finding a Φ-optimal design in Algorithm 1. As shown in Algorithm 1, we first search for a reasonable starting design based on subdata, followed by an iteration step to find an

optimal bounded approximate design. During the iteration, optimal weights must be computed for points in $\mathcal{X}_2^{(t)}$ (see line 17). This weight optimization is performed using Newton's method applied to the stationary conditions of Φ with respect to the weights, which yields quadratic convergence; the detailed procedure is presented as Algorithm 2 in Appendix C. Algorithm 2 is invoked at every iteration of Algorithm 1 whenever $\mathcal{X}_2^{(t)} \neq \emptyset$. It is also worth pointing out that the proofs will show that the iteration procedure of Algorithm 1 results in a better design if conditions (II)(b) or (II)(c) of Theorem 2 are not satisfied. That is, in that case $\Phi(\mathcal{P}^{(t+1)}) < \Phi(\mathcal{P}^{(t)})$.

Algorithm 1 Main Algorithm

1: **Input**: $\boldsymbol{x}_i$, $i = 1, \ldots, N$; subdata size n; optimality criterion $\Phi \in \Lambda$; information matrix $I_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$ of dimension p_1
2: **Output**: Optimal bounded approximate design ξ^*
3: **procedure** PREPARATION
4: Using IBOSS [24], select subdata S of size n and let ξ_S be the approximate design as in Eq. (3)
5: Compute $F_\Phi(\xi_S, \boldsymbol{x}_i)$, $i = 1, \ldots, N$
6: Form E_1 consisting of $\lfloor n/p_1 \rfloor$ points $\boldsymbol{x}_i \notin S$ with the smallest values for $F_\Phi(\xi_S, \boldsymbol{x}_i)$
7: Form E_3 consisting of $\lfloor n/p_1 \rfloor$ points $\boldsymbol{x}_i \in S$ with the largest values for $F_\Phi(\xi_S, \boldsymbol{x}_i)$
8: Assign $S = (S \backslash E_3) \cup E_1$
9: Repeat lines 5–8 p_1 times
10: Obtain $\boldsymbol{x}_{1*} = \operatorname{argmin}_{\boldsymbol{x}_i \notin S} F_\Phi(\xi_S, \boldsymbol{x}_i)$ and $\boldsymbol{x}_{3*} = \operatorname{argmax}_{\boldsymbol{x}_i \in S} F_\Phi(\xi_S, \boldsymbol{x}_i)$
11: Assign $S = (S \backslash \{\boldsymbol{x}_{3*}\}) \cup \{\boldsymbol{x}_{1*}\}$
12: Repeat lines 10–11 n times
13: Let $\xi_{\mathcal{P}^{(0)}}$ be the approximate design corresponding to the partition $\mathcal{P}^{(0)} = (\mathcal{X}_1^{(0)}, \mathcal{X}_2^{(0)}, \mathcal{X}_3^{(0)})$, where $\mathcal{X}_1^{(0)} = \mathcal{X} \backslash S$, $\mathcal{X}_2^{(0)} = \emptyset$, and $\mathcal{X}_3^{(0)} = S$
14: **end procedure**
15: **procedure** ITERATION
16: Set $t = 0$
17: If $\mathcal{X}_2^{(t)} \neq \emptyset$, compute the optimal weights between 0 and $1/n$ for points $\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}$ and redefine the partition $\mathcal{P}^{(t)}$ if some weights become 0 (move those points to $\mathcal{X}_1^{(t)}$) or $1/n$ (move those points to $\mathcal{X}_3^{(t)}$)
18: If conditions (II)(b) and (II)(c) in Theorem 2 are satisfied for $\xi_{\mathcal{P}^{(t)}}$, set $\xi^* = \xi_{\mathcal{P}^{(t)}}$ and go to line 23
19: Else, obtain $\boldsymbol{x}_{1*} = \operatorname{argmin}_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}, \boldsymbol{x}_i)$ and $\boldsymbol{x}_{3*} = \operatorname{argmax}_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}, \boldsymbol{x}_i)$
20: Define $\mathcal{X}_1^{(t+1)} = \mathcal{X}_1^{(t)} \setminus \{\boldsymbol{x}_{1*}\}$, $\mathcal{X}_2^{(t+1)} = \mathcal{X}_2^{(t)} \cup \{\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}\}$, $\mathcal{X}_3^{(t+1)} = \mathcal{X}_3^{(t)} \setminus \{\boldsymbol{x}_{3*}\}$
21: Set $t = t + 1$
22: Repeat lines 17–21 until $\Phi(\xi_{\mathcal{P}^{(t)}})$ converges
23: **end procedure**

An important property of any algorithm is that it converges. Our algorithm is designed for a general model, arbitrary choices of N and n, and multiple optimality criteria. To establish convergence in such a general setting, we introduce an assumption. Notice that this assumption is only needed in Theorems 3 and 5.

Let Ξ again denote the set of all bounded approximate designs $\xi = \{(\boldsymbol{x}_i, w_i), i = 1, \ldots, N\}$, but now for $0 \leq w_i \leq 1/n$ and $\sum_{i=1}^N w_i = 1$. Let $\xi \in \Xi$ be a design in the iteration stage of Algorithm 1, $\boldsymbol{x}_+$ and $\boldsymbol{x}_-$ be two points with weights w_+ and w_- in ξ, respectively, so that $w_+ < 1/n$ and $w_- > 0$. For small enough $\gamma > 0$, we can then define a new design $\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma] \in \Xi$ such that

(1) $w_i(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]) = w_i$ for $\boldsymbol{x}_i \in \mathcal{X} \setminus \{\boldsymbol{x}_+, \boldsymbol{x}_-\}$;
(2) $w_i(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]) = w_+ + \gamma$ for $x_i = \boldsymbol{x}_+$; and
(3) $w_i(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]) = w_- - \gamma$ for $x_i = \boldsymbol{x}_-$.

We will make the following assumption.

Assumption 1. For all designs ξ generated during the iteration procedure, it holds for a constant $C > 0$ that $|\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])| < C|\Phi(\xi)|$ for all γ with $\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma] \in \Xi$.

Remark 3. *Although this assumption may seem restrictive, it holds under mild regularity conditions. The construction of $\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]$ simply decreases the weight of one point and increases the weight of another (the resulting weights are still bounded between 0 and $1/n$), while leaving all other weights unchanged. Since the weight for any point is at most $1/n$, this adjustment will not significantly increase the value Φ, except in extreme cases (for example, when the size of the sub-data n is very*

small, such as being equal to the dimension p of the feature space). However, in practice, the size n is typically much larger than p. Assumption 1 can also be easily verified empirically. In our examples, $\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]) < 1.01\Phi(\xi)$ holds for all cases.

The next theorem shows that Algorithm 1 performs as desired.

Theorem 3. *Let $\Phi \in \Lambda$. With $\xi_{\mathcal{P}_0}$ as the bounded approximate design in line 3 of Algorithm 1, if the information matrix for $\xi_{\mathcal{P}^{(0)}}$ is non-singular and Assumption 1 holds, then the Φ-values for the sequence of designs $\xi_{\mathcal{P}^{(t)}}$ converge to the Φ-value of an optimal bounded approximate design as t increases. That is,*

$$\lim_{t\to\infty} \Phi(\xi_{\mathcal{P}^{(t)}}) = \min_{\xi\in\Xi} \Phi(\xi). \tag{7}$$

The result in Theorem 3 is desirable, but provides no information about the speed of convergence. Furthermore, strict verification of conditions (II)(b) and (II)(c) in Theorem 2 is infeasible due to floating-point errors in computing. Instead, we define a design ξ to be an ϵ-approximation of the optimal design ξ^* if $\Phi(\xi) - \Phi(\xi^*) \leq \epsilon$. In practice, we aim for an ϵ-approximation of an optimal design for small values of ϵ. We first give a sufficient condition for a design to be an ϵ-approximation of a Φ-optimal design.

Theorem 4. *Let Ξ be the set of all bounded approximate designs $\{(\boldsymbol{x}_i, w_i), i = 1, \ldots, N\}$ with $0 \leq w_i \leq 1/n$ and $\sum_{i=1}^N w_i = 1$, and let $\Phi \in \Lambda$. For $\epsilon > 0$, a design $\xi_\epsilon = \{(\boldsymbol{x}_i, w_i^\epsilon), i = 1, \ldots, N\}$ is an ϵ-approximation of a Φ-optimal design if there is a partition of the design space, $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \mathcal{X}_3$ and*

(a) $w_i^\epsilon = 0$ for $\boldsymbol{x}_i \in \mathcal{X}_1$ and $w_i^\epsilon = 1/n$ for $\boldsymbol{x}_i \in \mathcal{X}_3$;

(b) if $\mathcal{X}_2 = \emptyset$, then $\max_{\boldsymbol{x}_i\in\mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \leq \min_{\boldsymbol{x}_i\in\mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) + \epsilon$

(c) if $\mathcal{X}_2 \neq \emptyset$, then there is a number s so that for every $\boldsymbol{x}_i \in \mathcal{X}_2$

(i) $0 < w_i^\epsilon < 1/n$;
(ii) $s - \frac{\epsilon}{2} < F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) < s + \frac{\epsilon}{2}$;
(iii) $\max_{\boldsymbol{x}_i\in\mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \leq s + \frac{\epsilon}{2}$ and $s - \frac{\epsilon}{2} \leq \min_{\boldsymbol{x}_i\in\mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i)$.

The following theorem provides a result for the computational complexity of Algorithm 1 for finding a design ξ that is an ϵ-approximation of an optimal design ξ^* under Φ.

Theorem 5. *Let $\Phi \in \Lambda$ and Assumption 1 holds. For $\epsilon > 0$, if conditions (II)(b) and (II)(c) of Theorem 2 in line 18 of Algorithm 1 are replaced by (b) and (c) of Theorem 4, then Algorithm 1 is a polynomial time algorithm for finding an ϵ-approximation $\xi_{\mathcal{P}^{(t)}}$ of ξ^*.*

Remark 4. *Rather than using Algorithm 1 as presented, we make the substitutions suggested by Theorem 5 and stop the algorithm when an ϵ-approximation of ξ^* has been found. The proof of Theorem 5 also guarantees that, for a fixed ϵ, in the worst case, the computational complexity of Algorithm 1 is $O(nN^2p_1^2/\epsilon)$. With $\epsilon = 10^{-5}$, our numerical examples in Section 4 require relatively very few iterations for the identification of an ϵ-approximation of ξ^*.*

This computational complexity is, depending on the size of n and p relative to N, often lower than that for analyzing the full dataset in complex modeling scenarios—such as mixture-of-experts models—where analysis is very costly. More importantly, even if the proposed method is not used for subdata selection for large N, it still serves two key roles. First, it provides a principled benchmark for assessing the statistical efficiency of existing subdata selection methods, which is largely unavailable in the current literature. Second, it is particularly relevant in measurement-constrained settings, where responses are expensive or difficult to obtain. In such cases, the dominant cost lies in data acquisition rather than data analysis, and the primary objective is to maximize the statistical efficiency using a limited number of measured responses.

Remark 5. *We will refer to the subdata obtained at the end of line 9 of Algorithm 1 as IBOSS+ subdata. The computational complexity for IBOSS+ is just $O(Np^3)$, and we will see in Section 4 that the IBOSS+ subdata is usually already highly statistically efficient.*

Remark 6. *Since $\Phi(\xi)$ is not scale-invariant, it may be unclear how to select a meaningful value for ϵ in Theorem 5. We address a solution to this problem in Appendix C.4.*

Remark 7. *Theorems 3 and 5 provide theoretical guarantees for the main algorithm, but several implementation choices affecting computational efficiency remain. In the Appendix, we address these issues, including the selection of the initial subdata S in line 4 and the computation of optimal weights in line 17 of Algorithm 1.*

3.3 Subdata selection based on an optimal bounded approximate design

An optimal bounded approximate design ξ^* must be converted to subdata S of size n, or equivalently to a bounded approximate design ξ_S with weight $1/n$ for n points and 0 for all others. Although we could use an algorithm for converting an approximate design to an exact design, such as the rounding algorithm in [20] or the branch-and-bound algorithm in [1], our problem allows for a simpler efficient solution. In practice, close to n support points of ξ^* attain the upper bound weight of $1/n$. We include all of these points in the subdata, and add the points with the largest weights until we reach n points. Thus, the selected subdata consist of the n points with the largest weights in ξ^*. Observe that ξ^* has at least n points with a positive weight due to the constraint $0 \leq w_i \leq 1/n$. In the next section, we will see that the lower bound of the statistical efficiency of subdata selected by this method is generally more than 99.99%.

4 Simulations and comparisons

The previous sections introduced the methodology and established its theoretical properties. In this section, we evaluate its empirical performance through a series of simulation studies. We consider three scenarios: (i) a linear first-order regression model with $p = 10$ features; (ii) a full second-order logistic regression model with $p = 3$ features; and (iii) a clusterwise linear regression (CLR) model with $p = 10$ features and $G = 10$ clusters. The CLR setting is particularly noteworthy because it represents a class of models, mixture-of-experts models, for which the standard information matrix does not admit a closed-form expression. Instead, following [13], a surrogate matrix of dimension 110 is used instead of the information matrix. This makes Scenario (iii) a compelling test case: it demonstrates that our methodology extends beyond standard linear and generalized linear models to settings where only a surrogate for the information matrix is available. We give detailed description of the three scenarios in Appendix D.3, D.4 and D.5.

Where appropriate, we compare the proposed method with several existing approaches: SRS (simple random sampling); ODB (optimal design–based sampling) [6]; LEV (leverage sampling) [14]; IBOSS (information-based optimal subdata selection) [24]; OSS (orthogonal subsampling) [25]; SEQ (sequential sampling) [18]; and FO (first-order algorithms for optimal bounded approximate designs) [1]. We refer to our method as IBOSS OBD, which enhances IBOSS using optimal bounded designs. We also propose two simpler variants: IBOSS+, obtained by applying lines 5–9 of Algorithm 1 after IBOSS, and IBOSS++, which further applies lines 10–13.

The proposed framework is flexible and can be applied when interest lies in either the full parameter vector or specific components. In contrast, most competing methods for parameter estimation are primarily designed for settings in which all model parameters are of interest. Furthermore, IBOSS and OSS are tailored to first-order models and are used only in those settings. Methods such as LEV and FO require specific forms of the information matrix, making their application to Scenario (iii) unclear. The SEQ method generates a random sample size intended to approximate a prespecified target n. For comparability, we use the realized SEQ sample size also for other methods; however, when the realized size deviates substantially from the target, SEQ is excluded from comparison.

Finally, ODB is excluded from Scenarios (ii) and (iii) due to its unfavorable trade-off between statistical efficiency and computational scalability. We also considered the PGD method (projected gradient descent–based sampling) [26], but excluded it from the simulations due to scalability limitations (see Appendix D.2). For each scenario, we consider both D- and A-optimality for all or some parameters. Specifically, for Scenario (i), we consider D-optimality for all parameters and A-optimality for the first five slope parameters; for Scenario (ii), we consider D-optimality for the the main effects parameters and A-optimality for all parameters; for Scenario (iii), we consider D-optimality and A-optimality for all parameters. A summary of the applicability for each method can be found in Table 1.

The feature vectors, $\boldsymbol{x}_i$, $i = 1, \ldots, N$, are generated from the multivariate normal distribution $N_p(\mathbf{1}_p, \Sigma_p)$, where the diagonal and off-diagonal elements of Σ_p are 1 and 0.5, respectively. For each scenario, $N = 100000$ and $n = 1000$, except Scenario (iii) where $n = 10000$ (for the explanation,

Table 1: Applicability of subdata selection methods across scenarios and optimality criteria

Scenario	SRS	ODB	LEV	IBOSS	OSS	SEQ	FO	IBOSS+	IBOSS++	IBOSS OBD
(i) D-opt (all)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(i) A-opt (partial)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
(ii) D-opt (partial)	✓		✓			✓	✓	✓	✓	✓
(ii) A-opt (all)	✓		✓				✓	✓	✓	✓
(iii) D-opt (all)	✓					✓		✓	✓	✓
(iii) A-opt (all)	✓							✓	✓	✓

please see Appendix D.5). Results are based on averaging over 100 simulated data sets, each time generated according to the scenario under consideration.

All coding was done in Julia, and computations were performed on a Dell Laptop (2.3 GHz and 16 GB RAM). The ϵ in Theorem 4 is set to be 10^{-5} for all examples. For each configuration, the reported results are based on 100 replications. While we will report the computing times (in seconds) of the various methods, it should be noted that there could be tricks for some methods that we are unaware of. In particular, the OSS method, which is only a candidate for Scenario (i), shows larger computing times than we expected. It is possible that more efficient coding could result in shorter computing times.

Figure 1 provides a high-level assessment of all subdata selection methods in terms of statistical efficiency and computation time across all scenarios, offering a comprehensive evaluation of their relative performance. Detailed results are presented in Appendices D.3, D.4, and D.5.

SRS and LEV are by far the fastest methods, but this speed comes at a substantial loss in statistical efficiency. As a result, they are not competitive when statistical accuracy is a primary concern. ODB and OSS occupy a middle ground, achieving moderate statistical efficiency but at relatively high computational cost, leading to an unfavorable efficiency–time tradeoff. IBOSS is computationally efficient but consistently underperforms in terms of statistical efficiency, offering limited improvement over simpler baselines.

When all parameters are of interest, SEQ and FO emerge as the most competitive alternatives. However, SEQ performs poorly under A-optimality in Scenario (ii), where the realized sample sizes often fall well below the target, resulting in reduced statistical efficiency. The applicability of FO to the clusterwise linear regression model in Scenario (iii) is unclear. Moreover, even when FO converges, it is slower than IBOSS OBD, and its computational cost increases substantially under A-optimality. In addition, both SEQ and FO tend to lose statistical efficiency when interest is restricted to a subset of parameters, which is expected given that they are primarily developed for full-parameter settings.

In contrast, all three proposed methods consistently achieve near-perfect statistical efficiency across all scenarios, representing a clear improvement over existing approaches. Among them, IBOSS+ offers the most favorable balance between efficiency and computation time, delivering high statistical efficiency while remaining computationally efficient, and thus is particularly attractive in practice. IBOSS++ and IBOSS OBD also achieve near-optimal statistical efficiency, although at higher computational cost, especially in more challenging settings. Note that we can only make the claim about the high statistical efficiency of IBOSS+ because of the development of our IBOSS OBD methodology.

5 Discussion

By enabling rigorous assessment of subdata selection methods and principled identification of the most informative data points, this work supports more reliable and reproducible research in any field where data collection is costly or resource-constrained, including clinical trials, environmental monitoring, and large-scale surveys.

While many algorithms exist for subdata selection, their actual effectiveness is poorly understood. This paper introduces IBOSS OBD, which produces highly efficient subdata and, to our knowledge for the first time, provides a principled way to assess the statistical efficiency of any subdata selection method. Theoretically proven to converge to the true optimal design, IBOSS OBD performs well across linear, generalized linear, and clusterwise linear regression models. In practice, IBOSS+ offers the best tradeoff, achieving near-perfect statistical efficiency with low computation time across all scenarios.

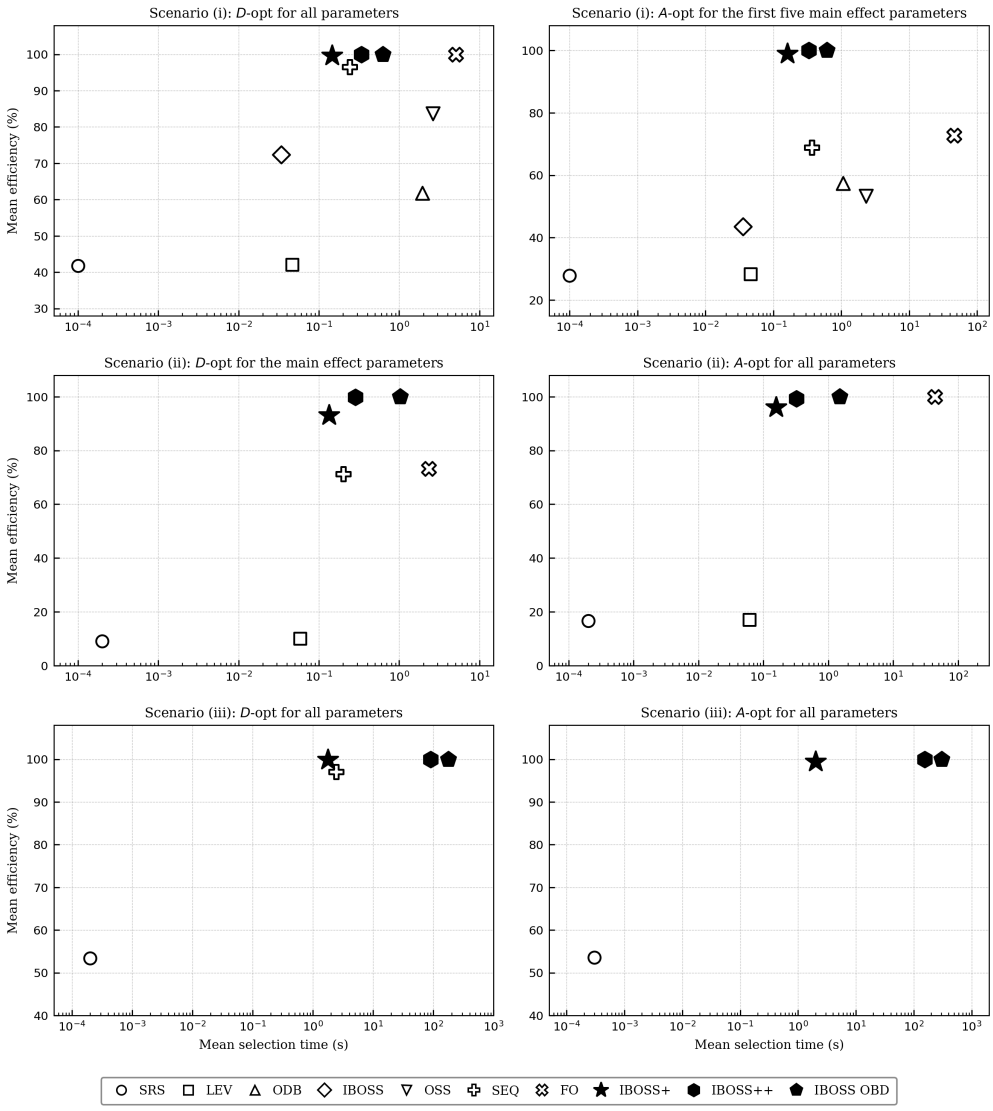

Figure 1: Statistical efficiency versus computation time

While this paper offers a very efficient method for subdata selection, there are various issues that go beyond the scope of this paper and that will require further study. First, for small datasets (which could be encountered in some of the machine learning applications), the lower and upper bounds in Equation (5) for the efficiency of the subdata may be further apart than in the empirical studies conducted here. This is simply because for smaller datasets, the ratio $\Phi(\xi^*)/\Phi(S^*)$ in Equation (5) may not be as close to 1. Hence, in such a case there will be more uncertainty in the efficiency of subdata S. Second, as in the case of Scenario (iii), more complex models may not have an explicit information matrix. Clusterwise linear regression models are a special and relatively simple case of mixture-of-experts models. While [13] developed a surrogate matrix that could be targeted for the case of clusterwise linear regression, this is, at least from a theoretical perspective, more difficult for general mixture-of-experts models. Third, paraphrasing George Box, all models are wrong, though some are useful. However, it may be uncertain which model is useful. Thus, expanding the methodology to selecting subdata that is efficient for a family of models will be very helpful. Finally, the algorithm for IBOSS OBD is fairly complex, and the authors are planning to develop corresponding software packages to simplify the implementation and broaden the usage of this algorithm.

Acknowledgments and Disclosure of Funding

The research of Min Yang was supported in part by the National Science Foundation (NSF) under grant 2210546. The research of John Stufken was supported in part by NSF grant 2304767. The research of Ming-Chung Chang was supported in part by Academia Sinica under grant AS-CDA-111-M05 and by the National Science and Technology Council of Taiwan under grants NSTC 113-2124-M-001-020, NSTC 114-2628-M-001-008-MY3, and NSTC 114-2124-M-001-008. The authors declare no competing interests. The authors thank the anonymous reviewers and the area chair for their constructive comments, which helped to improve the paper.

References

[1] S. D. Ahipaşaoğlu. A branch-and-bound algorithm for the exact optimal experimental design problem. *Statistics and Computing*, 31(5):65, 6 2021.

[2] A. Atkinson, A. Donev, and R. Tobias. *Optimum Experimental Designs, with SAS*. 9 2007.

[3] A. M. Bagirov, A. Mahmood, and A. Barton. Prediction of monthly rainfall in Victoria, Australia: Clusterwise linear regression approach. *Atmospheric Research*, 188, 5 2017.

[4] M. J. Brusco, J. D. Cradit, and A. Tashchian. Multicriterion Clusterwise Regression for Joint Segmentation Settings: An Application to Customer Value. *Journal of Marketing Research*, 40(2), 5 2003.

[5] Q. Cheng, H. Wang, and M. Yang. Information-based optimal subdata selection for big data logistic regression. *Journal of Statistical Planning and Inference*, 209:112–122, 12 2020.

[6] L. Deldossi and C. Tommasi. Optimal design subsampling from big datasets. *Journal of Quality Technology*, 54:93–101, 2022.

[7] D. Feldman. Core-sets: Updated survey. In F. Ros and S. Guillaume, editors, *Sampling Techniques for Supervised or Unsupervised Tasks*, page 23. Springer, Switzerland, 2020.

[8] Y. Huang, L. Tong, and J. Yang. Constrained d-optimal design for paid research study. *Statistica Sinica*, to appear, 2023.

[9] E. Isaacson and H. B. Keller. *Analysis of numerical methods*. John Wiley & Sons Inc., 1966.

[10] M. Khadka and A. Paz. Comprehensive Clusterwise Linear Regression for Pavement Management Systems. *Journal of Transportation Engineering, Part B: Pavements*, 143(4), 12 2017.

[11] J. Kiefer. General Equivalence Theory for Optimum Designs (Approximate Theory). *The Annals of Statistics*, 2(5):849–879, 6 1974.

[12] J. Kiefer and J. Wolfowitz. Optimum Designs in Regression Problems. *The Annals of Mathematical Statistics*, 30(2), 6 1959.

[13] Y. Liu, J. Stufken, and M. Yang. Information-based optimal subdata selection for clusterwise linear regression. *Statistica Sinica*, 36:715–740, 2026.

[14] P. Ma, M. W. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, 16:861–911, 2015.

[15] S. Mak and V. R. Joseph. Support points. *The Annals of Statistics*, 46:2562–2592, 2018.

[16] B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. *ICML'20: Proceedings of the 37th International Conference on Machine Learning*, 645:6950–6960, 2020.

[17] Y. W. Park, Y. Jiang, D. Klabjan, and L. Williams. Algorithms for Generalized Clusterwise Linear Regression. *INFORMS Journal on Computing*, 29(2), 5 2017.

[18] L. Pronzato and H. Wang. Sequential online subsampling for thinning experimental designs. *Journal of Statistical Planning and Inference*, 212:169–193, 2021.

[19] F. Pukelsheim. *Optimal design of experiments*. John Wiley & Sons, Inc., New York, NY, 1993.

[20] F. Pukelsheim and S. Rieder. Efficient rounding of approximate designs. *Biometrika*, 79:763–770, 1992.

[21] N. Rios and J. Stufken. Bootstrap aggregated designs for generalized linear models. *Journal of Data Science, Statistics, and Visualization*, 5, 2025.

[22] M. Sahm and R. Schwabe. A Note on Optimal Bounded Designs. pages 131–140. 2001.

[23] A. Tharwat and W. Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11:850, 2023.

[24] H. Wang, M. Yang, and J. Stufken. Information-Based Optimal Subdata Selection for Big Data Linear Regression. *Journal of the American Statistical Association*, 114(525):393–405, 1 2019.

[25] L. Wang, J. Elmstedt, W. K. Wong, and H. Xu. Orthogonal subsampling for big data linear regression. *The Annals of Applied Statistics*, 15(3):1273–1290, 9 2021.

[26] Y. Wang, A. W. Yu, and A. Singh. On computationally tractable selection of experiments in measurement-constrained regression models. *Journal of Machine Learning Research*, 18:1–41, 2017.

[27] W. Welch. Algorithmic complexity: Three np-hard problems in computational statistics. *Journal of Statistical Computation and Simulation*, 15:17–25, 1982.

[28] M. Yang, S. Biedermann, and E. Tang. On Optimal Designs for Nonlinear Models: A General and Efficient Algorithm. *Journal of the American Statistical Association*, 108(504):1411–1420, 6 2013.

[29] Y. Yu. D-optimal designs via a cocktail algorithm. *Statistics and Computing*, 21:475–481, 2011.

A A brief introduction to optimal design of experiments

This appendix gives a self-contained introduction to the concepts from the theory of optimal design of experiments on which our methodology is built. It is intended for readers who are not familiar with this literature, and it uses the notation of the main text. Comprehensive treatments can be found in [19] and [2]; the approximate design theory used here goes back to [12] and [11].

A.1 Statistical model and information matrix

In a designed experiment, the experimenter chooses the settings $\boldsymbol{x}$ of the explanatory variables at which responses y are observed. The set $\mathcal{X}$ of admissible settings is called the design space. The relationship between y and $\boldsymbol{x}$ is described by a parametric model with density or probability mass function $p(y \mid \boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^{p_1}$ is an unknown parameter vector, and observations are assumed to be independent. The Fisher information matrix for $\boldsymbol{\theta}$ from a single observation at $\boldsymbol{x}$ is the $p_1 \times p_1$ nonnegative definite matrix

$$I_{\boldsymbol{\theta}}(\boldsymbol{x}) = E\left[-\frac{\partial^2 \log p(y \mid \boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}^T}\right] = E\left[\frac{\partial \log p(y \mid \boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\, \frac{\partial \log p(y \mid \boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T}\right]. \tag{8}$$

For independent observations, information is additive: the information matrix for n observations taken at $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ is $\sum_{i=1}^{n} I_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. Equation (1) in the main text is precisely this expression for the observations that belong to the subdata.

Two classes of models are of particular interest in this paper.

- *Linear models.* Let $y = f(\boldsymbol{x})^T\boldsymbol{\theta} + \varepsilon$ with $\varepsilon \sim N(0, \sigma^2)$, where $f(\boldsymbol{x})$ is a known vector of regression functions. For a first-order model $f(\boldsymbol{x}) = (1, \boldsymbol{x}^T)^T$, while for a second-order model $f(\boldsymbol{x})$ also contains the squares and cross-products of the features. Here $I_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma^{-2} f(\boldsymbol{x}) f(\boldsymbol{x})^T$, which does not depend on $\boldsymbol{\theta}$. Since σ^{-2} is a common factor for all designs, it is irrelevant for comparing designs and is usually dropped, which gives the form $I_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{x}) f(\boldsymbol{x})^T$ used in Appendix C and, for the first-order model, the form $\boldsymbol{x}_i \boldsymbol{x}_i^T$ in Remark 1.
- *Generalized linear models (GLMs).* If the distribution of y belongs to an exponential family and depends on $\boldsymbol{x}$ through the linear predictor $\eta = f(\boldsymbol{x})^T\boldsymbol{\theta}$, then $I_{\boldsymbol{\theta}}(\boldsymbol{x}) = v(f(\boldsymbol{x})^T\boldsymbol{\theta})\, f(\boldsymbol{x}) f(\boldsymbol{x})^T$ for a positive weight function $v(\cdot)$ determined by the link and variance functions. For logistic regression, $v(\eta) = e^\eta/(1+e^\eta)^2$. The information matrix now depends on the unknown $\boldsymbol{\theta}$, so that a design can only be optimal *locally*, that is, for a given value of $\boldsymbol{\theta}$. This is the issue discussed in Appendix C.3.

The information matrix is the natural yardstick for the precision of parameter estimation. By the Cramér–Rao inequality, the covariance matrix of any unbiased estimator $\tilde{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ satisfies $\mathrm{Var}(\tilde{\boldsymbol{\theta}}) \succeq I_{\boldsymbol{\theta}}^{-1}$ in the Loewner ordering, where $I_{\boldsymbol{\theta}}$ is the information matrix of all observations. The MLE $\hat{\boldsymbol{\theta}}$ attains this bound: exactly for the linear model with normal errors, where $\mathrm{Var}(\hat{\boldsymbol{\theta}}) = \sigma^2\big(\sum_i f(\boldsymbol{x}_i) f(\boldsymbol{x}_i)^T\big)^{-1} = I_{\boldsymbol{\theta}}^{-1}$, and asymptotically under standard regularity conditions in general, where $\hat{\boldsymbol{\theta}}$ is approximately $N(\boldsymbol{\theta}, I_{\boldsymbol{\theta}}^{-1})$. Thus $I_{\boldsymbol{\theta}}^{-1}$ is the (asymptotic) covariance matrix of $\hat{\boldsymbol{\theta}}$, and a design that makes $I_{\boldsymbol{\theta}}$ "large" makes the estimation errors "small".

A.2 Optimality criteria and their relation to estimation error

The Loewner ordering of information matrices is only a partial ordering, so two designs can in general not be compared directly. An optimality criterion summarizes an information matrix by a scalar $\Phi(I_{\boldsymbol{\theta}})$ that is small when estimation errors are small, and a design is called Φ-optimal if it minimizes $\Phi(I_{\boldsymbol{\theta}})$. The two most widely used criteria are D- and A-optimality.

D-optimality. The D-criterion is $\Phi(I_{\boldsymbol{\theta}}) = \det(I_{\boldsymbol{\theta}}^{-1})^{1/p_1}$ or, equivalently for the purpose of finding an optimal design, the monotone transformation $\Phi(I_{\boldsymbol{\theta}}) = \log\det(I_{\boldsymbol{\theta}}^{-1})$ used in Section 2. Its connection to estimation error is through the confidence ellipsoid for $\boldsymbol{\theta}$. Using the (asymptotic) normality of $\hat{\boldsymbol{\theta}}$, a $100(1-\alpha)\%$ confidence region for $\boldsymbol{\theta}$ is the ellipsoid $\{\boldsymbol{\theta} : (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T I_{\boldsymbol{\theta}} (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \le \chi^2_{p_1, 1-\alpha}\}$, whose volume equals $c_{p_1,\alpha} \det(I_{\boldsymbol{\theta}}^{-1})^{1/2}$ for a constant $c_{p_1,\alpha}$ that depends only on p_1 and α. A D-optimal design therefore minimizes the volume of the confidence ellipsoid for $\boldsymbol{\theta}$ or, equivalently, the generalized variance $\det(\mathrm{Var}(\hat{\boldsymbol{\theta}}))$ of the estimator. An attractive property of the D-criterion is its invariance under nonsingular linear reparameterizations $\boldsymbol{\theta} \mapsto A\boldsymbol{\theta}$, since $\det(I_{\boldsymbol{\theta}}^{-1})$ is then multiplied by $\det(A)^2$ for every design. In particular, the D-optimal design does not depend on the units in which the features are measured.

A-optimality. The A-criterion is $\Phi(I_{\boldsymbol{\theta}}) = \mathrm{Tr}(I_{\boldsymbol{\theta}}^{-1})$. Since the diagonal elements of $I_{\boldsymbol{\theta}}^{-1}$ are the (asymptotic) variances of the components of $\hat{\boldsymbol{\theta}}$,

$$\mathrm{Tr}(I_{\boldsymbol{\theta}}^{-1}) = \sum_{j=1}^{p_1} \mathrm{Var}(\hat{\theta}_j) = E\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2, \tag{9}$$

where the last equality is exact for the normal linear model and holds asymptotically otherwise. An A-optimal design thus minimizes the total (or average) variance of the parameter estimates, which is the expected squared Euclidean distance between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$. Unlike the D-criterion, the A-criterion is not invariant under rescaling of the parameters, so it is most meaningful when the parameters are measured on comparable scales or when the sum of their variances is of direct interest.

Both criteria are members of Kiefer's Φ_p family, $\Phi_p(I_{\boldsymbol{\theta}}) = \big(p_1^{-1}\mathrm{Tr}(I_{\boldsymbol{\theta}}^{-p})\big)^{1/p}$, $0 < p < \infty$, with the D-criterion obtained as the limit $p \to 0$ and the A-criterion for $p = 1$ [11, 28]. As noted in Section 2, the results of this paper apply to the entire family for finite p.

Interest in a subset of the parameters. Frequently only a subvector $\boldsymbol{\theta}_1$ of $\boldsymbol{\theta}$ is of interest, for example the slope parameters but not the intercept, or the main effects but not the interactions. The

(asymptotic) covariance matrix of $\hat{\boldsymbol{\theta}}_1$ is the submatrix of $I_{\boldsymbol{\theta}}^{-1}$ that corresponds to $\boldsymbol{\theta}_1$, and the D- and A-criteria are then applied to this submatrix, with p_1 equal to the number of parameters of interest. All interpretations given above carry over: the criteria measure the volume of the confidence ellipsoid for $\boldsymbol{\theta}_1$ and the total variance of $\hat{\boldsymbol{\theta}}_1$, respectively. This is the setting of the A-criterion for the first five slope parameters in Scenario (i) and the D-criterion for the main effects in Scenario (ii) of Section 4.

Two useful properties. First, both $-\log\det(\cdot)$ and $\mathrm{Tr}(\cdot^{-1})$ are convex functions on the set of positive definite matrices. Convexity of Φ is what makes the approximate design problem of Appendix A.3 tractable. Second, both $\det(I_{\boldsymbol{\theta}}^{-1})^{1/p_1}$ and $\mathrm{Tr}(I_{\boldsymbol{\theta}}^{-1})$ are homogeneous of degree -1, that is, $\Phi(cI_{\boldsymbol{\theta}}) = c^{-1}\Phi(I_{\boldsymbol{\theta}})$ for every $c > 0$. Since taking c times as many observations at the same settings multiplies the information matrix by c, this property gives the Φ-efficiency a concrete interpretation. The efficiency of a design ξ relative to the Φ-optimal design ξ^* is defined as $\mathrm{Eff}_\Phi(\xi) = \Phi(\xi^*)/\Phi(\xi) \in (0, 1]$. If $\mathrm{Eff}_\Phi(\xi) = e$, then the criterion value of ξ equals that of ξ^* with its information matrix multiplied by e. Loosely speaking, n observations under ξ provide the same estimation precision, as measured by Φ, as en observations under the optimal design; equivalently, n/e observations under ξ are needed to match the precision of n observations under ξ^*. An efficiency of 50% thus means that the sample size would have to be doubled to compensate for the suboptimal design. This is the interpretation of the statistical efficiencies reported in Section 4 and Appendices D.3–D.5. Recall that for the D-criterion the efficiency is computed with $\Phi(I_{\boldsymbol{\theta}}) = \det(I_{\boldsymbol{\theta}}^{-1})^{1/p_1}$, as explained in Section 3, because a ratio of $\log\det$ values has no such interpretation.

A.3 Exact and approximate designs

An *exact design* with n observations is a list of n points of $\mathcal{X}$, where points may be repeated. It can be described by its distinct support points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ and the number of observations n_i taken at each of them, with $\sum_{i=1}^k n_i = n$, or by the proportions $w_i = n_i/n$. Its information matrix is $n\sum_{i=1}^k w_i I_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$. Finding an optimal exact design is a combinatorial optimization problem, and for the D-criterion it is NP-hard [27]; this is the result that underlies Theorem 1.

The key idea of [12] is to relax the requirement that nw_i be an integer. An *approximate design* (also called a continuous design) is a probability measure $\xi = \{(\boldsymbol{x}_i, w_i), i = 1, \ldots, k\}$ on $\mathcal{X}$, with $w_i \geq 0$ and $\sum_{i=1}^k w_i = 1$, with information matrix

$$I_{\boldsymbol{\theta}}(\xi) = \sum_{i=1}^{k} w_i I_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \tag{10}$$

and $\Phi(\xi) = \Phi(I_{\boldsymbol{\theta}}(\xi))$ as in Equation (3) of the main text. The matrix in (10) is the information matrix per observation; an exact design with proportions w_i is the special case in which all nw_i are integers. Because $I_{\boldsymbol{\theta}}(\xi)$ is linear in the weights and Φ is convex, $\Phi(\xi)$ is a convex function of ξ. Consequently, every local minimum is a global minimum, and optimality can be verified through first-order conditions, which is the content of the General Equivalence Theorem in Appendix A.4. In classical applications, an optimal approximate design ξ^* is converted to an exact design by rounding the nw_i^* to integers, for instance with the efficient rounding procedure of [20], and the resulting loss of efficiency is small when n is large relative to the number of support points. Moreover, since every exact design is also an approximate design, $\Phi(\xi^*)$ is a lower bound for the criterion value of any exact design of size n. This lower bound is what makes the efficiency bounds in Equation (5) possible.

A.4 The General Equivalence Theorem

For an approximate design ξ and a point $\boldsymbol{x} \in \mathcal{X}$, let $\delta_{\boldsymbol{x}}$ denote the design that puts all its weight at $\boldsymbol{x}$, and consider the directional derivative

$$F_\Phi(\xi; \boldsymbol{x}) = \lim_{\alpha \downarrow 0} \frac{\Phi((1-\alpha)\xi + \alpha\delta_{\boldsymbol{x}}) - \Phi(\xi)}{\alpha}, \tag{11}$$

which is the rate at which the criterion changes when a small fraction of the observations is moved to the point $\boldsymbol{x}$. A negative value means that ξ can be improved by shifting weight towards $\boldsymbol{x}$. The General Equivalence Theorem [11] states that, for a convex and differentiable criterion Φ, an approximate design ξ^* is Φ-optimal if and only if

$$F_\Phi(\xi^*; \boldsymbol{x}) \geq 0 \quad \text{for all } \boldsymbol{x} \in \mathcal{X}, \tag{12}$$

and in that case $F_\Phi(\xi^*; \boldsymbol{x}) = 0$ at every support point of ξ^*. Condition (12) is simply the first-order optimality condition for the convex problem of minimizing $\Phi(\xi)$ over the set of all approximate designs. For the D- and A-criteria, the directional derivatives have the explicit forms

$$F_D(\xi; \boldsymbol{x}) = p_1 - \mathrm{Tr}\big(I_{\boldsymbol{\theta}}(\xi)^{-1} I_{\boldsymbol{\theta}}(\boldsymbol{x})\big) \quad \text{and} \quad F_A(\xi; \boldsymbol{x}) = \mathrm{Tr}\big(I_{\boldsymbol{\theta}}(\xi)^{-1}\big) - \mathrm{Tr}\big(I_{\boldsymbol{\theta}}(\xi)^{-2} I_{\boldsymbol{\theta}}(\boldsymbol{x})\big), \quad (13)$$

where F_D is computed for $\Phi(I_{\boldsymbol{\theta}}) = \log\det(I_{\boldsymbol{\theta}}^{-1})$. For a linear model with $I_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{x}) f(\boldsymbol{x})^T$, the D-optimality condition (12) becomes $\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})^T I_{\boldsymbol{\theta}}(\xi^*)^{-1} f(\boldsymbol{x}) = p_1$. Since $f(\boldsymbol{x})^T I_{\boldsymbol{\theta}}(\xi)^{-1} f(\boldsymbol{x})$ is proportional to the variance of the predicted response at $\boldsymbol{x}$, this is the celebrated result of [12] that a D-optimal design also minimizes the maximum prediction variance over the design space. Thus a D-optimal design is not only good for estimating $\boldsymbol{\theta}$ but also for prediction.

Besides characterizing optimal designs, the General Equivalence Theorem is the engine behind most algorithms for finding them. For a design ξ that is not optimal, points with the most negative values of $F_\Phi(\xi; \boldsymbol{x})$ are the ones that should receive more weight, and support points with the largest values should receive less. Iteratively exchanging weight between such points, with step sizes chosen to decrease Φ, produces a sequence of designs that converges to ξ^*, and $\min_{\boldsymbol{x}} F_\Phi(\xi; \boldsymbol{x})$ provides a lower bound on $\Phi(\xi^*)$ that can be used to monitor convergence. The exchange steps in Algorithm 1 are of exactly this type.

A.5 From optimal design to subdata selection

In subdata selection, the design space is the finite set $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of observed feature vectors, and "taking an observation at $\boldsymbol{x}_i$" corresponds to including the ith data point in the subdata and, in an active learning setting, obtaining its label. This differs from the classical design problem in two respects. First, $\mathcal{X}$ is finite but typically very large. Second, each data point can be included at most once, so that the replication numbers n_i are restricted to be 0 or 1, that is, $w_i \in \{0, 1/n\}$. This is the formulation in Equation (2). The classical relaxation to arbitrary nonnegative weights is not useful here, because an optimal approximate design typically concentrates its weight on a small number of support points (for the D-criterion often only p_1 of them), which cannot be converted to subdata of size $n \gg p_1$ in which every point is used at most once. The appropriate relaxation retains the restriction $w_i \leq 1/n$ and only relaxes the integrality, leading to the *bounded approximate designs* with $0 \leq w_i \leq 1/n$ in Equation (3) [22]. Since $\sum_{i=1}^N w_i = 1$, such a design necessarily has at least n support points. Theorem 2 is the corresponding version of the General Equivalence Theorem: the condition $F_\Phi(\xi^*; \boldsymbol{x}) \geq 0$ in (12) is replaced by a threshold condition under which points with the smallest directional derivatives receive the maximal weight $1/n$, points with the largest directional derivatives receive weight 0, and points with intermediate weights share a common value s of the directional derivative. Section 3 shows how an optimal bounded approximate design ξ^* is converted to subdata and how the inequality $\Phi(\xi^*) \leq \Phi(S_{\text{opt}})$, the analogue of the classical lower bound of Appendix A.3, yields the efficiency bounds in Equation (5).

B Technical appendices and supplementary material

B.1 Proofs of Theorem 1, Theorem 2, and Lemma 1

Proof of Theorem 1. In the context of optimal design, Theorem 3 in [27] shows that identifying a D-optimal exact design is NP-hard. Using the notation in Equation 1, identifying an optimal exact design requires identification of $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_N)$, subject to $\sum_{i=1}^N \delta_i = n$, such that the determinant of the information matrix is maximized. However, while the δ_i's must be 0 or 1 in the subdata selection problem, in the optimal exact design problem they can be any non-negative integer, as long as they add to n.

The result of Theorem 1 follows from a connection between the optimal exact design problem and the subdata selection problem. If we replicate each of N points n times, then selecting optimal subdata of size n from the resulting $n \times N$ points under the D-optimality criterion is equivalent to selecting a D-optimal exact design for the original population of N points. Thus, if there were a polynomial time algorithm for the subdata selection problem, there would be one for the optimal exact design problem, which would contradict the result in [27]. □

Proof of Theorem 2. We first show that $(I) \Rightarrow (II)$. Starting from the Φ-optimal design ξ^* in (I), let $(\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ be the corresponding partition of the points $\boldsymbol{x}_i$ with weight equal to ν_i, between ν_i and μ_i, and equal to μ_i, respectively. We first consider the case that $\mathcal{X}_2 \neq \emptyset$. If there is more than one point in $\mathcal{X}_2$, let $\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2} \in \mathcal{X}_2$ be two distinct points. For a small positive ϵ, define the new design ξ that has the same weights as ξ^*, except that it increases the weight for $\boldsymbol{x}_{c1}$ by ϵ and decreases that for $\boldsymbol{x}_{c2}$ by the same amount. Thus, writing $w_i(\xi)$ for the new weights,

1. $w_i(\xi) = w_i$ for $x_i \in \mathcal{X} \setminus \{\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2}\}$;
2. $w_{c1}(\xi) = w_{c1} + \epsilon$ for $\boldsymbol{x}_{c1}$; and
3. $w_{c2}(\xi) = w_{c2} - \epsilon$ for $\boldsymbol{x}_{c2}$.

Here, ϵ is taken to be small enough so that $w_{c1}(\xi) \leq \mu_{c1}$ and $w_{c2}(\xi) \geq \nu_{c2}$. Using that Φ is linearly differentiable, we have

$$\begin{aligned} F_\Phi(\xi^*, \xi) = & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2 \setminus \{\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2}\}} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i \\ & + F_\Phi(\xi^*; \boldsymbol{x}_{c1})(w_{c1} + \epsilon) + F_\Phi(\xi^*; \boldsymbol{x}_{c2})(w_{c2} - \epsilon) \\ = & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i \\ & + F_\Phi(\xi^*; \boldsymbol{x}_{c1})\epsilon - F_\Phi(\xi^*; \boldsymbol{x}_{c2})\epsilon \\ = & F_\Phi(\xi^*, \xi^*) + \epsilon(F_\Phi(\xi^*; \boldsymbol{x}_{c1}) - F_\Phi(\xi^*; \boldsymbol{x}_{c2})) \\ = & \epsilon(F_\Phi(\xi^*; \boldsymbol{x}_{c1}) - F_\Phi(\xi^*; \boldsymbol{x}_{c2})) \geq 0, \end{aligned}$$

where we used that $F_\Phi(\xi^*, \xi^*) = 0$ and ξ^* is Φ-optimal. Thus, $F_\Phi(\xi^*; \boldsymbol{x}_{c1}) \geq F_\Phi(\xi^*; \boldsymbol{x}_{c2})$. Similarly, $F_\Phi(\xi^*; \boldsymbol{x}_{c2}) \geq F_\Phi(\xi^*; \boldsymbol{x}_{c1})$. Therefore,

$$F_\Phi(\xi^*; \boldsymbol{x}_{c2}) = F_\Phi(\xi^*; \boldsymbol{x}_{c1}). \tag{14}$$

Since $\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2} \in \mathcal{X}_2$ are arbitrary, $F_\Phi(\xi^*; \boldsymbol{x}_i)$ is the same for all $\boldsymbol{x}_i \in \mathcal{X}_2$, say s. Also by $F_\Phi(\xi^*, \xi^*) = 0$, it follows that

$$F_\Phi(\xi^*; \boldsymbol{x}_i) = s = \frac{-\sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i - \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i}{1 - (\sum_{\boldsymbol{x}_i \in \mathcal{X}_1} \nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} \mu_i)} \tag{15}$$

for $\boldsymbol{x}_i \in \mathcal{X}_2$. Since (15) also holds if $\mathcal{X}_2$ consists of a single point, (II)(c) follows. Next we will show that (II)(b) holds. Select points $\boldsymbol{x}_c \in \mathcal{X}_2$ and $\boldsymbol{x}_u \in \mathcal{X}_3$. For a small positive ϵ, we define another new design $\hat{\xi}$ with weights equal to the weights of ξ^*, except that the weight at $\boldsymbol{x}_c$ is ϵ more and the weight at $\boldsymbol{x}_u$ is ϵ less that that of ξ^*. Thus,

1. $w_i(\hat{\xi}) = w_i$ for $\boldsymbol{x}_i \in \mathcal{X} \setminus \{\boldsymbol{x}_c, \boldsymbol{x}_u\}$;
2. $w_c(\hat{\xi}) = w_c + \epsilon$ for $\boldsymbol{x}_c$; and
3. $w_u(\hat{\xi}) = w_u - \epsilon$ for $\boldsymbol{x}_u$.

Here, ϵ is small enough so that $w_c(\hat{\xi}) \leq \mu_c$ and $w_u(\hat{\xi}) \geq \nu_u$. Then,

$$\begin{aligned} F_\Phi(\xi^*, \hat{\xi}) = & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3 \setminus \{\boldsymbol{x}_u\}} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2 \setminus \{\boldsymbol{x}_c\}} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i \\ & + F_\Phi(\xi^*; \boldsymbol{x}_c)(w_c + \epsilon) + F_\Phi(\xi^*; \boldsymbol{x}_u)(w_u - \epsilon) \\ = & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i \\ & + F_\Phi(\xi^*; \boldsymbol{x}_c)\epsilon - F_\Phi(\xi^*; \boldsymbol{x}_u)\epsilon \\ = & \epsilon(F_\Phi(\xi^*; \boldsymbol{x}_c) - F_\Phi(\xi^*; \boldsymbol{x}_u)) \geq 0, \end{aligned}$$

where we used again that $F_\Phi(\xi^*, \xi^*) = 0$ and ξ^* is Φ-optimal. Thus $F_\Phi(\xi^*; \boldsymbol{x}_u) \leq F_\Phi(\xi^*; \boldsymbol{x}_c) = s$ for arbitrary $\boldsymbol{x}_u \in \mathcal{X}_3$. Similarly, we can show that $F_\Phi(\xi^*; \boldsymbol{x}_\ell) \geq s$ for arbitrary $\boldsymbol{x}_\ell \in \mathcal{X}_1$. Consequently (II)(b) follows.

Next we consider the case $\mathcal{X}_2 = \emptyset$. Select arbitrary points $\boldsymbol{x}_\ell \in \mathcal{X}_1$ and $\boldsymbol{x}_u \in \mathcal{X}_3$. For a small positive ϵ, define a new design $\bar{\xi}$ with the same weights as ξ^*, except that the weight for $\boldsymbol{x}_\ell$ is increased by ϵ and that for $\boldsymbol{x}_u$ is decreased by ϵ. Thus,

1. $w_i(\bar{\xi}) = w_i$ for $\boldsymbol{x}_i \in \mathcal{X} \setminus \{\boldsymbol{x}_\ell, \boldsymbol{x}_u\}$;
2. $w_\ell(\bar{\xi}) = w_\ell + \epsilon$; and
3. $w_u(\bar{\xi}) = w_u - \epsilon$.

Here, ϵ is small enough so that $w_\ell(\bar{\xi}) \le \mu_\ell$ and $w_u(\bar{\xi}) \ge \nu_u$. Then,

$$\begin{aligned} F_\Phi(\xi^*, \bar{\xi}) = & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)\nu_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)\mu_i + \epsilon F_\Phi(\xi^*; \boldsymbol{x}_\ell) - \epsilon F_\Phi(\xi^*; \boldsymbol{x}_u) \\ = & \quad \epsilon(F_\Phi(\xi^*; x_\ell) - F_\Phi(\xi^*; x_u)) \ge 0. \end{aligned}$$

Since $\boldsymbol{x}_\ell \in \mathcal{X}_1$ and $\boldsymbol{x}_u \in \mathcal{X}_3$ are arbitrary, it follows that $\max_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i) \le \min_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)$.

Now we show that $(II) \Rightarrow (I)$.
By the convexity of Φ, it suffices to show $F_\Phi(\xi^*, \xi) \ge 0$ for every competing design ξ. For any given $\xi \in \Xi$, we have $w_i(\xi) \ge w_i$ on $\mathcal{X}_1$ and $w_i(\xi) \le w_i$ on $\mathcal{X}_3$. Thus we have

$$\begin{aligned} F_\Phi(\xi^*, \xi) = & \sum_{x_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i(\xi) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i(\xi) + \sum_{x_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i(\xi) \\ \ge & \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i + \min_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi^*; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i \in \mathcal{X}_1}(w_i(\xi) - w_i) \\ & + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i + \max_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi^*; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3}(w_i(\xi) - w_i) \\ & + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2} F_\Phi(\xi^*; \boldsymbol{x}_i)w_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2} F_\Phi(\xi^*; \boldsymbol{x}_i)(w_i(\xi) - w_i) \\ \ge & \quad s(\sum_{\boldsymbol{x}_i \in \mathcal{X}_1}(w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3}(w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2}(w_i(\xi) - w_i)) = 0. \end{aligned}$$

□

Proof of Lemma 1. Let n_i be the number of points in $\mathcal{X}^i$, $i = 1, 2, 3$. Let $n_2 \ge 2$, which is the only case for which a proof is needed. The information matrix of $\xi_\mathcal{P}$ can be viewed as a function of $\boldsymbol{w}'$=$(w_1, \ldots, w_{n_2-1})$, where w_i denotes the weight for $\boldsymbol{x}_i \in \mathcal{X}_2$, $i = 1, \ldots, n_2 - 1$. The weight of $\boldsymbol{x}_{n_2} \in \mathcal{X}_2$ is $1 - n_3/n - \sum_{i=1}^{n_2-1} w_i$. By Theorem 4 of [28], the optimal weights are critical points, that is, solutions of $\frac{\partial \Phi(\xi_\mathcal{P})}{\partial w_i} = 0$, or boundary points, that is, $w_i = 0$ or $1/n$. But in the latter case, the procedure would have moved the points to $\mathcal{X}_1$ or $\mathcal{X}_3$. Hence,

$$\frac{\partial \Phi(\xi_\mathcal{P})}{\partial w_i} = F_\Phi(\xi_\mathcal{P}; \boldsymbol{x}_i) - F_\Phi(\xi_\mathcal{P}; \boldsymbol{x}_{n_2}) = 0, \ i = 1, \ldots, n_2 - 1. \tag{16}$$

□

B.2 Proof of Theorem 3

Before proving Theorem 3, we need the following lemma, which uses the notation and assumptions as in Theorem 3 and Algorithm 1.

Lemma 2. *For $\Phi \in \Lambda$, suppose that there are a bounded approximate design $\xi_0 \in \Xi$ and $\delta > 0$, so that, for an integer $t > 0$, $\Phi(\xi_{\mathcal{P}^{(t)}}) - \Phi(\xi_0) > \delta$. Then one of the following statements must hold.*

(a) If $\mathcal{X}_2^{(t)} = \emptyset$, then

$$F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) \le -\delta. \tag{17}$$

(b) If $\mathcal{X}_2^{(t)} \ne \emptyset$ and $\mathcal{X}_1^{(t)} = \emptyset$, then

$$F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s \ge \delta. \tag{18}$$

(c) If $\mathcal{X}_2^{(t)} \ne \emptyset$ and $\mathcal{X}_3^{(t)} = \emptyset$, then

$$F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s \le -\delta. \tag{19}$$

(d) If $\mathcal{X}_2^{(t)} \ne \emptyset$, $\mathcal{X}_1^{(t)} \ne \emptyset$ and $\mathcal{X}_3^{(t)} \ne \emptyset$, then one of the two following inequalities must hold:

$$\begin{aligned} (i) F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s \le -\frac{\delta}{2}, \\ (ii) F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s \ge \frac{\delta}{2}. \end{aligned} \tag{20}$$

Here $s = F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i)$, $x_i \in \mathcal{X}_2^{(t)}$.

Proof. For $0 \leq \alpha \leq 1$, consider the design $(1-\alpha)\xi_{\mathcal{P}^{(t)}} + \alpha\xi_0$. Using the convexity of Φ, we have that $\Phi((1-\alpha)\xi_{\mathcal{P}^{(t)}} + \alpha\xi_0) \leq (1-\alpha)\Phi(\xi_{\mathcal{P}^{(t)}}) + \alpha\Phi(\xi_0)$. Hence,

$$\begin{aligned} F_\Phi(\xi_{\mathcal{P}^{(t)}}, \xi_0) &= \lim_{\alpha \downarrow 0} \frac{1}{\alpha} \left(\Phi((1-\alpha)\xi_{\mathcal{P}^{(t)}} + \alpha\xi_0) - \Phi(\xi_{\mathcal{P}^{(t)}})\right) \\ &\leq \Phi(\xi_0) - \Phi(\xi_{\mathcal{P}^{(t)}}) \\ &< -\delta. \end{aligned} \tag{21}$$

For each $\boldsymbol{x}_i$, let w_i and $w_i(\xi_0)$ be the weights in $\xi_{\mathcal{P}^{(t)}}$ and ξ_0, respectively. By the definitions of $\mathcal{X}_1^{(t)}$ and $\mathcal{X}_3^{(t)}$, since Φ is linearly differentiable, we have

$$\begin{aligned} F_\Phi(\xi_{\mathcal{P}^{(t)}}, \xi_0) = &\sum_{x_i \in \mathcal{X}_1^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i(\xi_0) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i(\xi_0) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i(\xi_0) \\ \geq &\sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i + F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) \\ &+ \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i + F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) \\ &+ \sum_{\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i) w_i + \sum_{\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}} F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i)(w_i(\xi_0) - w_i) \\ = &F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) \sum_{x_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) + F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) + \\ &+ \sum_{\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}} s(w_i(\xi_0) - w_i), \end{aligned} \tag{22}$$

where we used that $F_\Phi(\xi_{\mathcal{P}^{(t)}}, \xi_{\mathcal{P}^{(t)}}) = 0$. Notice that when $\mathcal{X}_2^{(t)} \neq \emptyset$, by Lemma 1, $s = F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_i)$, $\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}$, is well-defined. By Equations (21) and (22), for the right hand side (RHS) of (22) we have

$$\text{RHS of (22)} < -\delta. \tag{23}$$

Next, we consider each case in the statement of the lemma separately, and will arrive at a contradiction with Equation (23) if a case does not hold.

For Case (a), suppose that Equation (17) does not hold, so that $F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - F_\Phi(\xi_{\mathcal{P}^{(t)}}, \boldsymbol{x}_{3*}) > -\delta$. This implies that

$$\begin{aligned} \text{RHS of (22)} &\geq F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) \sum_{\boldsymbol{x}_i \in \mathcal{X}} (w_i(\xi_0) - w_i) - \delta \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) \\ &\geq -\delta. \end{aligned}$$

This contradicts Equation (23), so that Case (a) must hold.

For Case (b), suppose that Equation (18) does not hold, so that $F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s < \delta$. This implies that

$$\begin{aligned} \text{RHS of (22)} &= s \sum_{\boldsymbol{x}_i \in \mathcal{X}} (w_i(\xi_0) - w_i) + (F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) \\ &\geq \delta \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) \geq -\delta. \end{aligned}$$

This again contradicts Equation (23), so that Case (b) must hold.

Next, for Case (c), suppose that Equation (19) does not hold, so that $F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s > -\delta$. This implies that

$$\begin{aligned}\text{RHS of (22)} &= s \sum_{\boldsymbol{x}_i \in \mathcal{X}} (w_i(\xi_0) - w_i) + (F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s) \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) \\ &\geq -\delta \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) \geq -\delta.\end{aligned}$$

This is once more a contradiction to Equation (23), this time implying that Case (c) must hold.

Finally, for Case (d), suppose that neither of the two inequalities in (20) holds, so that

$$\begin{aligned}(i) F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s > -\frac{\delta}{2}, \text{ and} \\ (ii) F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s < \frac{\delta}{2}.\end{aligned} \tag{24}$$

This implies that

$$\begin{aligned}\text{RHS of (22)} &= s \sum_{\boldsymbol{x}_i \in \mathcal{X}} (w_i(\xi_0) - w_i) + (F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s) \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) \\ &\quad + (F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) \\ &\geq -\frac{\delta}{2} \sum_{\boldsymbol{x}_i \in \mathcal{X}_1^{(t)}} (w_i(\xi_0) - w_i) + \frac{\delta}{2} \sum_{\boldsymbol{x}_i \in \mathcal{X}_3^{(t)}} (w_i(\xi_0) - w_i) \\ &\geq -\delta.\end{aligned} \tag{25}$$

This contradiction to Equation (23) implies that Case (d) must hold. □

Lemma 3. *Under Φ optimality, i.e.,*

$$\Phi(\xi) = \begin{cases} \log det(I_{\boldsymbol{\theta}}^{-1}(\xi)), & D - optimality \\ Tr(I_{\boldsymbol{\theta}}^{-1}(\xi)), & A - optimality \end{cases}, \tag{26}$$

for designs ξ_1, ξ_2 and ξ_3 and positive value γ for which $\xi = \xi_1 + \gamma(\xi_2 - \xi_3)$ is a design and $\Phi(\xi)$ is bounded, then $\Phi(\xi)$ has a bounded second derivative respective to γ.

Proof. We first consider the case $p = 0$; the case $p > 0$ is analogous. When $p = 0$,

$$\frac{\partial^2 \Phi(\xi)}{\partial \gamma^2} = \mathrm{Tr}\big(I_{\boldsymbol{\theta}}^{-1}(\xi) \Delta I_{\boldsymbol{\theta}}^{-1}(\xi) \Delta\big), \quad \Delta := I_{\boldsymbol{\theta}}(\xi_2) - I_{\boldsymbol{\theta}}(\xi_3).$$

Since $\Phi(\xi)$ is bounded, the eigenvalues (hence all entries) of $I_{\boldsymbol{\theta}}^{-1}(\xi)$ are bounded. Each $I_{\boldsymbol{\theta}}(\xi_j)$, $j = 2, 3$, is a convex combination of finitely many well-defined information matrices, so their entries are bounded; therefore Δ has bounded entries. This establishes the claim.

□

Proof of Theorem 3. Let $\xi^* = argmin_{\xi \in \Xi} \Phi(\xi)$ and consider $\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])$, where Ξ and $\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]$ defined in Section 3.2. Then $\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])$ is infinitely differentiable with respect to γ. In the iteration stage, the Φ-values of designs decrease. Thus, they are uniformly bounded by the Φ-value of the design at the beginning of the iteration stage, say $\Phi(\xi_0)$. By Assumption 1, we have that $|\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])| < C|\Phi(\xi)|$ for some constant $C > 0$. Consequently, $|\Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])| < C|\Phi(\xi_0)|$. By Lemma 3, there exists $K < \infty$, such that

$$\frac{\partial^2 \Phi(\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma])}{\partial \gamma^2} \leq K. \tag{27}$$

Notice that $\Phi(\xi_{\mathcal{P}^{(t)}})$ is a decreasing function of t and bounded by $\Phi(\xi^*)$. Thus $\lim_{t \to \infty} \Phi(\xi_{\mathcal{P}^{(t)}})$ exists. We shall show that

$$\lim_{t \to \infty} \Phi(\xi_{\mathcal{P}^{(t)}}) = \Phi(\xi^*). \tag{28}$$

If Equation (28) does not hold, then, by the monotonicity of $\Phi(\xi_{\mathcal{P}^{(t)}})$, there exists a $\delta > 0$ such that $\Phi(\xi_{\mathcal{P}^{(t)}}) - \Phi(\xi^*) > \delta$ for all t. We will show that this leads to a contradiction by considering the following four exhaustive scenarios for the sequence of partitions $\mathcal{P}^{(t)}$:

(i) $\mathcal{X}_2^{(t)} = \emptyset$ for infinitely many values of t;

(ii) $\mathcal{X}_2^{(t)} \neq \emptyset$ and $\mathcal{X}_1^{(t)} = \emptyset$ for infinitely many values of t;

(iii) $\mathcal{X}_2^{(t)} \neq \emptyset$ and $\mathcal{X}_3^{(t)} = \emptyset$ for infinitely many values of t;

(iv) $\mathcal{X}_2^{(t)} \neq \emptyset$, $\mathcal{X}_1^{(t)} \neq \emptyset$, and $\mathcal{X}_3^{(t)} \neq \emptyset$ for infinitely many values of t.

If Scenario (i) holds, then there must exist an infinite subsequence of $\mathcal{P}^{(t)}$ such that $\mathcal{X}_2^{(t)} = \emptyset$. For simplicity of notation, we index the subsequence again by t, and have a sequence $\mathcal{P}^{(t)}$ so that $\mathcal{X}_2^{(t)} = \emptyset$ for all t. Define the new design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]$, which belongs to Ξ for $\gamma < 1/n$. Note that $\boldsymbol{x}_{1*}$ and $\boldsymbol{x}_{3*}$ depend on t, but we surpress this in the notation for simplicity. By the definition of $\mathcal{P}^{(t+1)}$, we have that

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) \leq \Phi(\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]). \tag{29}$$

Using a Taylor series expansion near $\gamma = 0$, and applying Equations (17) and (27), it can be shown that

$$\begin{aligned}\Phi(\xi_{\mathcal{P}^{(t)}}[x_{1*}, x_{3*}, \gamma]) =& \Phi(\xi_{\mathcal{P}^{(t)}}) + \left(F_\Phi(\xi_{\mathcal{P}^{(t)}}, \boldsymbol{x}_{1*}) - F_\Phi(\xi_{\mathcal{P}^{(t)}}, \boldsymbol{x}_{3*})\right)\gamma \\ &+ \frac{1}{2}\gamma^2 \left.\frac{\partial^2 \Phi(\xi[x_{1*}, x_{3*}, \gamma])}{\partial \gamma^2}\right|_{\gamma=\gamma'} \\ \leq& \Phi(\xi_{\mathcal{P}^{(t)}}) - \delta\gamma + \frac{1}{2}K\gamma^2,\end{aligned} \tag{30}$$

where $\gamma' \in [0, \gamma]$. If $K > n\delta$, let $\gamma = \frac{\delta}{K}$. By Equation (30), we have that

$$\Phi(\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \frac{\delta}{K}]) - \Phi(\xi_{\mathcal{P}^{(t)}}) \leq -\frac{\delta^2}{2K}. \tag{31}$$

By (29) and (31), we have, for all $t \geq 0$,

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) - \Phi(\xi_{\mathcal{P}^{(t)}}) \leq -\frac{\delta^2}{2K}. \tag{32}$$

Since Equation (32) holds for all t, it implies that $\lim_{t\to\infty} \Phi(\xi_{\mathcal{P}^{(t)}}) = -\infty$, which contradicts the fact that $\Phi(\xi_{\mathcal{P}^{(t)}})$ is bounded by $\Phi(\xi^*)$. Similar arguments can be applied to the case when $K \leq n\delta$, in which case we let $\gamma = 1/n$.

If Scenario (ii) holds, then, again by focusing on a subsequence of $\mathcal{P}^{(t)}$, we can assume that $\mathcal{X}_2^{(t)} \neq \emptyset$ and $\mathcal{X}_1^{(t)} = \emptyset$ for all t. Define $\boldsymbol{x}_{2*}$ as a point in $\mathcal{X}_2^{(t)}$ with the smallest weight, say $w_{min}^{(t)}$. Note that $\boldsymbol{x}_{2*}$ depends on t, which we again suppress in the notation for simplicity. Note also that $0 < w_{min}^{(t)} < 1/n$, so that $\liminf_{t\to\infty} w_{min}^{(t)}$ exists. Furthermore, $\liminf_{t\to\infty} w_{min}^{(t)} < 1/n$, because otherwise, for any given $\epsilon > 0$, for large enough t we would have that $w_i > 1/n - \epsilon$ for all $\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}$. Since points that are not in $\mathcal{X}_2^{(t)}$ must be in $\mathcal{X}_3^{(t)}$ for this scenario, it would contradict $\sum_{i=1}^N w_i = 1$. Define $\gamma_0 = \liminf_{t\to\infty} w_{min}^{(t)}$. Consider the design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{2*}, \boldsymbol{x}_{3*}, \gamma]$, where $\gamma \in [0, (1/n - \gamma_0)/2]$. From a Taylor series expansion as in Equation (30), and using Equation (18), the remainder of the argument is similar to that for design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]$ in Scenario (i).

We skip the proof for Scenario (iii) since this case is similar to Scenario (ii).

If Scenario (iv) holds, then (i) or (ii) of Equation (20) must hold for infinitely many values of t. Without loss of generality we assume that (i) since the proof for (ii) is similar. By taking a subsequence, we can again assume that it holds for all t. For this scenario, we define $\boldsymbol{x}_{2*}$ (with its dependence on t again suppressed in the notation) as a point in $\mathcal{X}_2^{(t)}$ with the largest weight, say $w_{max}^{(t)}$. Then, $0 < w_{max}^{(t)} < 1/n$ and $\limsup_{t\to\infty} w_{max}^{(t)}$ exists.

Define $\gamma_1 = \limsup_{t\to\infty} w_{max}^{(t)}$, and assume that it is positive. Consider the design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{2*}, \gamma]$ for $\gamma \in [0, \gamma_1/2]$. From a Taylor series expansion as in Equation (30), and using (i) of Equation (20), the remainder of the argument is similar to that for design $\xi_{\mathcal{P}^{(t)}}[x_{1*}, x_{3*}, \gamma]$ in the proof for Scenario (i).

It remains to consider the case $\gamma_1 = 0$. Then, for any $\epsilon > 0$, for large enough t, $w_i < \epsilon$ for all $\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}$. With n_i denoting the number of points in $\mathcal{X}_i^{(t)}$, $i = 1, 2, 3$, we have, for large enough t, that $n_3/n \leq 1 < n_3/n + n_2\epsilon$. Since this holds for any $\epsilon > 0$, it implies that $n_3/n = 1$ for large enough t. But that implies that $n_2 = 0$, which contradicts that $\mathcal{X}_2^{(t)} \neq \emptyset$ for all t. □

B.3 Proofs of Theorem 4 and Theorem 5

Proof of Theorem 4. From the convexity of Φ, for any design $\xi = \{(\boldsymbol{x}_i, w_i(\xi)), i = 1, \ldots, N\}$, we have that

$$\begin{aligned}\Phi\left((1-\alpha)\xi_\epsilon + \alpha\xi\right) - \Phi(\xi_\epsilon) &\leq (1-\alpha)\Phi(\xi_\epsilon) + \alpha\Phi(\xi) - \Phi(\xi_\epsilon) \\ &= \alpha\left(\Phi(\xi) - \Phi(\xi_\epsilon)\right).\end{aligned}$$

This implies that

$$F_\Phi(\xi_\epsilon, \xi) \leq \Phi(\xi) - \Phi(\xi_\epsilon). \tag{33}$$

Next we shall show that $F_\Phi(\xi_\epsilon, \xi) > -\epsilon$. Using w_i to denote the weight of $\boldsymbol{x}_i$ in ξ_ϵ, note that $w_i(\xi) \geq w_i$ on $\mathcal{X}_1$ and $w_i(\xi) \leq w_i$ on $\mathcal{X}_3$. There are two cases we need to consider (i) $\mathcal{X}_2 \neq \emptyset$ and (ii) $\mathcal{X}_2 = \emptyset$.

Case (i) $\mathcal{X}_2 \neq \emptyset$, then we have

$$\begin{aligned}F_\Phi(\xi_\epsilon, \xi) = &\sum_{x_i\in\mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i(\xi) + \sum_{\boldsymbol{x}_i\in\mathcal{X}_2} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i(\xi) + \sum_{x_i\in\mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i(\xi) \\ \geq &\sum_{x_i\in\mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i + \min_{\boldsymbol{x}_i\in\mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i\in\mathcal{X}_1} (w_i(\xi) - w_i) \\ &+ \sum_{\boldsymbol{x}_i\in\mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i - \max_{\boldsymbol{x}_i\in\mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i\in\mathcal{X}_3} (w_i - w_i(\xi)) \\ &+ \sum_{\boldsymbol{x}_i\in\mathcal{X}_2} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i + (s - \frac{\epsilon}{2}) \sum_{\boldsymbol{x}_i\in\mathcal{X}_2 \text{ and } w_i(\xi) > w_i} (w_i(\xi) - w_i) \\ &+ (s + \frac{\epsilon}{2}) \sum_{\boldsymbol{x}_i\in\mathcal{X}_2 \text{ and } w_i(\xi) \leq w_i} (w_i(\xi) - w_i) \\ \geq &s\left(\sum_{\boldsymbol{x}_i\in\mathcal{X}_1} (w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i\in\mathcal{X}_3} (w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i\in\mathcal{X}_2} (w_i(\xi) - w_i)\right) \\ &- \frac{\epsilon}{2}\left(\sum_{\boldsymbol{x}_i\in\mathcal{X}_1} (w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i\in\mathcal{X}_3} (w_i - w_i(\xi))\right) \\ &- \frac{\epsilon}{2}\left(\sum_{\boldsymbol{x}_i\in\mathcal{X}_2 \text{ and } w_i(\xi) > w_i} (w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i\in\mathcal{X}_2 \text{ and } w_i(\xi) \leq w_i} (w_i - w_i(\xi))\right) \\ \geq &-\epsilon.\end{aligned}$$

Case (ii) $\mathcal{X}_2 = \emptyset$, then we have

$$\begin{aligned}
F_\Phi(\xi_\epsilon, \xi) &= \sum_{x_i \in \mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i(\xi) + \sum_{x_i \in \mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i(\xi) \\
&\geq \sum_{x_i \in \mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i + \min_{\boldsymbol{x}_i \in \mathcal{X}_1} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} (w_i(\xi) - w_i) \\
&\quad + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) w_i - \max_{\boldsymbol{x}_i \in \mathcal{X}_3} F_\Phi(\xi_\epsilon; \boldsymbol{x}_i) \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} (w_i - w_i(\xi)) \\
&\geq \max_{\boldsymbol{x}_i \in \mathcal{X}_3} \left(\sum_{\boldsymbol{x}_i \in \mathcal{X}_1} (w_i(\xi) - w_i) + \sum_{\boldsymbol{x}_i \in \mathcal{X}_3} (w_i(\xi) - w_i) \right) - \epsilon \sum_{\boldsymbol{x}_i \in \mathcal{X}_1} (w_i(\xi) - w_i) \\
&\geq -\epsilon.
\end{aligned}$$

For both cases, we have $F_\Phi(\xi_\epsilon, \xi) > -\epsilon$. By (33), ξ_ϵ is an ϵ-approximation of the Φ-optimal design ξ^*. □

Proof of Theorem 5. We shall study $\Phi(\xi_{\mathcal{P}^{(t)}}) - \Phi(\xi_{\mathcal{P}^{(t+1)}})$ when, for some t, Condition (b) or Condition (c) in Theorem 4 is not satisfied. We use notation defined in the proof of Theorem 3, such as ξ^*, $\xi[\boldsymbol{x}_+, \boldsymbol{x}_-, \gamma]$, and K. First consider the case $\mathcal{X}_2^{(t)} = \emptyset$. Then

$$F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) \leq -\epsilon. \tag{34}$$

For design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]$, from the definition of $\mathcal{P}^{(t+1)}$, it follows that

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) \leq \Phi(\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]). \tag{35}$$

Using again a Taylor series expansion near $\gamma = 0$, and applying Equations (34) and (27), we can show that

$$\begin{aligned}
\Phi(\xi_{\mathcal{P}^{(t)}}[x_{1*}, x_{3*}, \gamma]) =& \Phi(\xi_{\mathcal{P}^{(t)}}) + (F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}))\, \gamma \\
&+ \frac{1}{2}\, \gamma^2 \frac{\partial^2 \Phi(\xi[x_{1*}, x_{3*}, \gamma])}{\partial \gamma^2}\bigg|_{\gamma = \gamma'} \\
\leq& \Phi(\xi_{\mathcal{P}^{(t)}}) - \epsilon\gamma + \frac{1}{2} K \gamma^2,
\end{aligned} \tag{36}$$

where $\gamma' \in [0, \gamma]$. If $K > n\epsilon$, let $\gamma = \frac{\epsilon}{K}$. By Equation (36), we have that

$$\Phi(\xi_{\mathcal{P}^{(t)}}[x_{1*}, x_{3*}, \gamma]) - \Phi(\xi_{\mathcal{P}^{(t)}}) \leq -\frac{\epsilon^2}{2K}. \tag{37}$$

If $K \leq n\epsilon$, let $\gamma = 1/n$. Then we have

$$\Phi(\xi_{\mathcal{P}^{(t)}}[x_{1*}, x_{3*}, \gamma]) - \Phi(\xi_{\mathcal{P}^{(t)}}) \leq -\frac{\epsilon}{2n}. \tag{38}$$

By Equations (35), (37), and (38), we see that

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) - \Phi(\xi_{\mathcal{P}^{(t)}}) < \begin{cases} -\frac{\epsilon^2}{2K} & \text{when } K > n\epsilon \\ -\frac{\epsilon}{2n} & \text{when } K \leq n\epsilon. \end{cases} \tag{39}$$

Next, consider the case $\mathcal{X}_2^{(t)} \neq \emptyset$. Notice that Algorithm 2 guarantees that the weights of points $\boldsymbol{x}_i \in \mathcal{X}_2^{(t)}$ are optimized strictly between 0 and $1/n$. By Lemma 1 and the setting of Algorithm 2, (i) and (ii) of Condition (c) in Theorem 4 are automatically satisfied. There are two possibilities when (iii) of Condition (c) in Theorem 4 is not satisfied:

(i) $F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{1*}) - s < -\frac{\epsilon}{2}$ when $\mathcal{X}_1^{(t)} \neq \emptyset$

(ii) $F_\Phi(\xi_{\mathcal{P}^{(t)}}; \boldsymbol{x}_{3*}) - s > \frac{\epsilon}{2}$ when $\mathcal{X}_3^{(t)} \neq \emptyset$.

If (i) holds, let $\boldsymbol{x}_{2*} \in \mathcal{X}_2^{(t)}$ be a point with the largest weight of the points in $\mathcal{X}_2^{(t)}$. Since the weights must sum to 1, the weight of $\boldsymbol{x}_{2*}$ must be at least $\frac{1}{nN}$. Consider the design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{2*}, \gamma]$ for a small enough γ. Similarly as for $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]$, we have that

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) - \Phi(\xi_{\mathcal{P}^{(t)}}) < \begin{cases} -\frac{\epsilon^2}{8K} & \text{when } K > nN\epsilon/2 \\ -\frac{\epsilon}{4nN} & \text{when } K \leq nN\epsilon/2. \end{cases} \tag{40}$$

If (ii) holds, let $\boldsymbol{x}_{2*} \in \mathcal{X}_2^{(t)}$ be a point with the smallest weight of the points in $\mathcal{X}_2^{(t)}$. The weight of $\boldsymbol{x}_{2*}$ is at most $(n-1)/n^2$. For small enough γ, consider the new design $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{2*}, \boldsymbol{x}_{3*}, \gamma]$. Similarly as for $\xi_{\mathcal{P}^{(t)}}[\boldsymbol{x}_{1*}, \boldsymbol{x}_{3*}, \gamma]$, we have

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) - \Phi(\xi_{\mathcal{P}^{(t)}}) < \begin{cases} -\frac{\epsilon^2}{8K} & \text{when } K > n^2\epsilon/2 \\ -\frac{\epsilon}{4n^2} & \text{when } K \leq n^2\epsilon/2. \end{cases} \tag{41}$$

Thus, when the conditions in Theorem 5 are not satisfied for some t, then we have by Equations (39), (40), and (41)

$$\Phi(\xi_{\mathcal{P}^{(t+1)}}) - \Phi(\xi_{\mathcal{P}^{(t)}}) < max(-\frac{\epsilon^2}{8K}, -\frac{\epsilon}{4nN}). \tag{42}$$

With

$$\xi^* = argmin_{\xi \in \Xi} \Phi(\xi),$$

for a given $\epsilon > 0$, after at most

$$(\Phi(\xi_{\mathcal{P}^{(0)}}) - \Phi(\xi^*)) \times \max(\frac{8K}{\epsilon^2}, \frac{4nN}{\epsilon}) \tag{43}$$

iterations, $\mathcal{P}^{(t)}$ converges to a ϵ-approximation of ξ^*. Notice that $\Phi(\xi_{\mathcal{P}^{(0)}}) - \Phi(\xi^*)$ is finite and K is constant; this implies that the algorithm's run time is bounded by a polynomial function in n and N. Consequently, the desired conclusion follows. □

C Further considerations for the main algorithm

Theorems 3 and 5 provide assurances for the performance of the main algorithm. But choices that can impact speed still need to be made. This includes selecting initial subdata S of size n in line 4 of Algorithm 1 and an efficient algorithm to derive optimal weights in line 17 of Algorithm 1.

C.1 Selecting initial subdata

Starting Algorithm 1 in line 4 with efficient subdata will generally result in a shorter running time to reach convergence. We recommend that an adaptation of the subdata selection method in [24] and [5] be used to select the initial subdata. The models considered by these authors use a predictor of the form $\beta_0 + \boldsymbol{\beta}^T \boldsymbol{x}$, and their recommendation is roughly to select for each j, $1 \leq j \leq p$, the $n/(2p)$ points $\boldsymbol{x}_i$ with the largest values for the jth feature and the $n/(2p)$ points $\boldsymbol{x}_i$ with the smallest values for the jth feature. They demonstrate desirable theoretical results for this strategy, called the IBOSS strategy, and we will adapt it here for more general statistical models.

More generally, the information at a point $\boldsymbol{x}$ will be proportional to $I_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{x}) \times f^T(\boldsymbol{x})$. Here, $f(\boldsymbol{x})$ is a vector for a univariate response variable. If the response variable is multivariate, $f(\boldsymbol{x})$ is a matrix. In that case, we vectorize this matrix, and call it again $f(\boldsymbol{x})$.

Form the matrix $(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))$, remove its constant rows (if any), and let F be the resulting matrix with p_2 rows. We propose the following algorithm to generate initial subdata S.

(1) For the first row of F, select $\lfloor n/(2p_2) \rfloor$ points with the largest values and $\lfloor n/(2p_2) \rfloor$ points with the smallest values in this row.

(2) For the next row, exclude points that have already been selected, and select $\lfloor n/(2p_2) \rfloor$ points with the largest values and $\lfloor n/(2p_2) \rfloor$ points with the smallest values in this row.

(3) Repeat step (2) until all p_2 rows have been used.

(4) If the total number of points selected is less than n, randomly select $n - 2p_2\lfloor n/(2p_2) \rfloor$ points from the remaining points.

(5) Use the selected n points as the initial subdata for line 4 in Algorithm 1.

The computational complexity of this algorithm is $O(Np_2)$. Although the subdata generated by this fast algorithm may not be highly efficient for all models, it provides a starting point for Algorithm 1 that tends to be far better than using randomly selected subdata.

C.2 Deriving optimal weights

Common algorithms for deriving optimal weights, such as the multiplicative algorithm [29], first-order algorithm [1], and lift-one algorithm [8], create a sequence of weights that converges to the optimal weights. These algorithms improve at every iteration and have a closed-form iterative formula for updating the weights. A disadvantage is that they tend to be rather slow, especially for large N.

A different idea for finding optimal weights was proposed in [28]. With constraints $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$, the authors proved that optimal weights correspond to a boundary point, i.e., $w_i = 0$ for at least one i, or to a critical point for the objective function, i.e., $\frac{\partial \Phi(\xi_{\mathcal{P}^{(t)}})}{\partial w_i} = 0$ for all i (see Theorem 4 in [28]), where $\mathcal{P}^{(t)}$ is as in the iteration procedure of Algorithm 1. With constraints $0 \leq w_i \leq 1/n$ and $\sum_{i=1}^{N} w_i = 1$, the same result holds, except that boundary points are now those points with $w_i = 0$ or $w_i = 1/n$ for at least one i. Thus, optimal weights can be obtained by solving nonlinear equations. Normally, there is no closed-form solution, so we use a numerical approach based on Newton's method. In general, this has the desirable quadratic convergence rate, meaning that the number of correct digits roughly doubles after each iteration [9]. This alone already makes the algorithm fast, and, using the notation of the iteration step in Algorithm 1, it is applied only to the points in $\mathcal{X}_2^{(t)}$. With n_2 and n_3 as the number of points in $\mathcal{X}_2$ and $\mathcal{X}_3$, respectively, the procedure for deriving optimal weights using Newton's method is presented as Algorithm 2. The vector $\boldsymbol{w}_2^{(j)}$ in the algorithm will contain weights for the points in $\mathcal{X}_2$. During the algorithm entries can temporarily become negative (line 9 of Algorithm 2), but in the end all elements will be between 0 and $1/n$. The length of the vector can also change during the algorithm (line 13 of Algorithm 2) because the size of $\mathcal{X}_2$ changes when a weight becomes equal to 0 or $1/n$.

Algorithm 2 Optimal Weights Algorithm

1: **Input**: The input for Algorithm 1; a bounded approximate design $\xi_{\mathcal{P}}$ with $0 \leq w_i \leq 1/n$ and the corresponding partition $\mathcal{P} = (\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ with $\mathcal{X}_2 \neq \emptyset$; a positive scalar ϵ_1; the maximum number of iterations MI_1.
2: **Output**: Updated partition $\mathcal{P} = (\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3)$ and optimal weights for the points in $\mathcal{X}_2$
3: **procedure** OBTAIN OPTIMAL WEIGHTS
4: Set n_2 and n_3 as the size of $\mathcal{X}_2$ and $\mathcal{X}_3$, respectively
5: If $n_2 = 1$, set $\boldsymbol{w}_2 = (1 - n_3/n)$ and go to line 16
6: Else
7: Set $\boldsymbol{\nu}_2^{(0)} = \frac{1-n_3/n}{n_2}\mathbf{1}_{n_2-1}$, and set $\boldsymbol{w}_2^{(0)} = ({\boldsymbol{\nu}_2^{(0)}}^T, 1 - n_3/n - {\boldsymbol{\nu}_2^{(0)}}^T\mathbf{1}_{n_2-1})^T$ as the weight vector for points in $\mathcal{X}_2$.
8: Set $\alpha = 1$ and $j = 1$
9: Compute $\boldsymbol{\nu}_2^{(j)} = \boldsymbol{\nu}_2^{(j-1)} - \alpha \left(\frac{\partial^2 \Phi(\xi_{\mathcal{P}})}{\partial \boldsymbol{\nu}_2 \partial \boldsymbol{\nu}_2^T}|_{\boldsymbol{\nu}_2 = \boldsymbol{\nu}_2^{(j-1)}}\right)^{-} \left(\frac{\partial \Phi(\xi_{\mathcal{P}})}{\partial \boldsymbol{\nu}_2}|_{\boldsymbol{\nu}_2 = \boldsymbol{\nu}_2^{(j-1)}}\right)$ and set $\boldsymbol{w}_2^{(j)} = ({\boldsymbol{\nu}_2^{(j)}}^T, 1 - n_3/n - {\boldsymbol{\nu}_2^{(j)}}^T\mathbf{1}_{n_2-1})^T$
10: If, component-wise, $0 \leq \boldsymbol{w}_2^{(j)} \leq 1/n$, go to line 14
11: Else, set $\alpha = \alpha/2$
12: If $\alpha > 10^{-10}$, go to line 9
13: Else, for $\boldsymbol{x} \in \mathcal{X}_2$ with the largest weight, if its weight exceeds $1/n$, set $\mathcal{X}_2 = \mathcal{X}_2 \backslash \{\boldsymbol{x}\}$, $\mathcal{X}_3 = \mathcal{X}_3 \cup \{\boldsymbol{x}\}$, and go to line 4; otherwise, for $\boldsymbol{x} \in \mathcal{X}_2$ with the smallest weight, set $\mathcal{X}_2 = \mathcal{X}_2 \backslash \{\boldsymbol{x}\}$, $\mathcal{X}_1 = \mathcal{X}_1 \cup \{\boldsymbol{x}\}$, and go to line 4
14: If $\| \frac{\partial \Phi(\xi_{\mathcal{P}})}{\partial \boldsymbol{\nu}_2}|_{\boldsymbol{\nu}_2 = \boldsymbol{\nu}_2^{(j)}} \| < \epsilon_1$, set $\boldsymbol{w}_2 = \boldsymbol{w}_2^{(j)}$ and go to line 16
15: Else, if $j \leq MI_1$, set $j = j + 1$ and go to line 9; else set $\boldsymbol{w}_2 = \boldsymbol{w}_2^{(j)}$ and go to line 16
16: Output $\boldsymbol{w}_2$
17: **end procedure**

Remark 8. *In Line 9,* $\left(\frac{\partial^2 \Phi(\xi_{\mathcal{P}})}{\partial \boldsymbol{\nu}_2 \partial \boldsymbol{\nu}_2^T}|_{\boldsymbol{\nu}_2=\boldsymbol{\nu}_2^{(j-1)}}\right)^-$ *refers to a Moore-Penrose pseudo-inverse.*

Remark 9. *In the backtracking line search (Lines 9–12), the step size* α *is halved whenever the updated weights violate the boundary constraints. Once a valid step is found and the algorithm proceeds to the next Newton iteration,* α *is not reset to 1. Although it may be tempting to reset* $\alpha = 1$*, our experience suggests that this is often inefficient, as boundary violations are likely to occur again. Resetting* α *to 1 can therefore delay the detection of such violations and slow down the overall convergence.*

Remark 10. *In Line 13, if a weight hits a boundary constraint, the point is moved to* $\mathcal{X}_1$ *or* $\mathcal{X}_3$*, and the algorithm returns to Line 4. Instead of resetting all remaining weights in* $\mathcal{X}_2$ *to a uniform distribution, one may retain their current values to improve computational efficiency.*

For the examples in Section 4, we set MI_1=40 and $\epsilon_1 = 10^{-6}$.

C.3 An information matrix that depends on $\boldsymbol{\theta}$

As noted previously, the information matrix $I_{\boldsymbol{\theta}}$ depends on $\boldsymbol{\theta}$ for GLMs. This issue has been extensively addressed in the optimal design literature. In the simplest case, one assumes that a guess for $\boldsymbol{\theta}$ is available from prior experience. This value for $\boldsymbol{\theta}$ can then be plugged into $I_{\boldsymbol{\theta}}$, and the resulting matrix can be used in our approach to find the subdata. Alternatively, one could try to find a bounded approximate design that is not optimal for any $\boldsymbol{\theta}$, but that has high efficiency for multiple plausible values of $\boldsymbol{\theta}$. An approach along these lines can be found in [21], although it would have to be extended to bounded designs. Another approach would be to assume a prior distribution for $\boldsymbol{\theta}$ that expresses the uncertainty of our knowledge of $\boldsymbol{\theta}$, and use a semi-Bayesian approach to find a bounded approximate design.

Unlike a typical design of experiments problem, for the subdata selection problem, we have a large amount of data. We can use a randomly selected subset of these data to get an idea of the value of $\boldsymbol{\theta}$. Labels would have to be collected for the randomly selected subset if not yet available. We will not pursue this further here because our objective is to demonstrate the efficiency of our subdata selection method. If $\boldsymbol{\theta}$ is to be estimated, that comparison will be partially obscured by the loss of efficiency resulting from estimating $\boldsymbol{\theta}$.

C.4 Some practical considerations

In line 19 of Algorithm 1, it is theoretically possible for $\mathcal{X}_1^{(t)}$ and/or $\mathcal{X}_3^{(t)}$ to be empty. In such cases, no corresponding point exchange is performed. In lines 11 and 20 of Algorithm 1 and line 13 of Algorithm 2, each time only one point is exchanged. It is possible to increase the number of points that are exchanged, and this could be more efficient for some problems. However, for the problems we considered and because we have good initial subdata from IBOSS+, exchanging a single point each time is sufficient. While applying the exchanges to the IBOSS+ subdata results in small efficiency gains, it also results in a significant increase in computation time (cf. Section 4).

As discussed prior to Theorem 5, an exact verification of the optimality of a partition $\mathcal{P}$ with optimal weights is not possible by checking conditions (II)(b) and (II)(c) of Theorem 2. Therefore, we will actually use the result of Theorem 5 to find an ϵ-approximation of ξ^* for a small positive tolerance value ϵ. For the empirical studies in Section 4, ϵ is set to be 10^{-5}. As explained in Remark 6, a scale-invariant approach with a varying value for ϵ could be considered as an alternative.

Line 22 of Algorithm 1 needs in our experience relatively few iterations to reach convergence. However, to safeguard against the rare situation that more iterations are needed, the maximum number of iterations can be capped by a predetermined number MI_2. As a practical recommendation, we suggest that MI_2 be set to n. Algorithm 1 typically converges before it reaches MI_2, but even when it doesn't, little change occurs after increasing the number of iterations.

In Remark 6 we observe that $\Phi(\xi)$ is not scale-invariant. As an alternative to selecting a value of ϵ to obtain an ϵ-approximation of an optimal design ξ^*, we could consider the relative distance in Φ-values for a design ξ and the Φ-optimal design ξ^*, namely $\left(\Phi(\xi) - \Phi(\xi^*)\right)/\Phi(\xi^*)$. This is easier to interpret since we may think of this ratio as the reciprocal of the relative Φ-efficiency of ξ minus 1. To demonstrate how Theorem 5 can help to find a design that achieves a desired relative distance to a Φ-optimal design, suppose that we wish to find a design ξ with $\left(\Phi(\xi) - \Phi(\xi^*)\right)/\Phi(\xi^*) \leq \alpha$ for some

Table 2: Mean and standard deviation (in parentheses) of $\Phi(\xi_{\text{IBOSS OBD}})/\Phi(\xi_{\text{FO}})$ under the A-optimality criterion

MI_3	n	$10n$	$20n$	$50n$
$\Phi(\xi_{\text{IBOSS OBD}})/\Phi(\xi_{\text{FO}})$	94.72% (1.65%)	99.95% (0.03%)	99.99% (0.01%)	100.00% (0.00%)

specified $\alpha > 0$. At any stage of the algorithm, if ξ_0 is a candidate design, use $\epsilon = \alpha\Phi(\xi_0)/(\alpha+1)$ in Theorem 5. With this ϵ, we see that if $\Phi(\xi_0) - \Phi(\xi^*) \leq \epsilon$, then

$$\frac{\Phi(\xi_0) - \Phi(\xi^*)}{\Phi(\xi^*)} \leq \frac{\epsilon}{\Phi(\xi_0) - \epsilon} = \alpha.$$

Thus, if ξ_0 satisfies conditions (b) and (c) of Theorem 4 for this value of ϵ, then it satisfies $\left(\Phi(\xi_0) - \Phi(\xi^*)\right)/\Phi(\xi^*) \leq \alpha$. If it does not satisfy the conditions, we improve ξ_0 by continuing to run the algorithm, get a design ξ_1, recalculate ϵ, and check the conditions of Theorem 5 again, but now with this new value of ϵ. We continue until we find a design and the corresponding value of ϵ for which the conditions in Theorem 5 hold.

D More simulation studies

In this section, we present additional simulation results and comparisons, including a discussion of tuning parameter selection for the FO method, justification for excluding the PGD method, and further empirical evaluations.

D.1 About FO method

For the FO algorithm, the description in [1] provides no details to select an initial design or choose the maximum number of iterations, MI_3. For a fair comparison to IBOSS OBD, we start the algorithm with the design based on the IBOSS subdata and use the same convergence criterion. For MI_3, if it is large, there is a risk that convergence takes a long time. On the other hand, if MI_3 is small, the resulting design could be far from optimal. In contrast, IBOSS OBD tends to converge relatively fast. For example, for $N = 1000$, $n = 100$, and 100 repetitions, Table 2 compares the ratio of the A-optimality criterion values for $\xi_{IBOSS\ OBD}$ and ξ_{FO} for $p = 2$ and the full second-order model. Multiple values of MI_3 are used, keeping $MI_1 = 40$ (recall that MI_1 denotes the number of iterations in Algorithm 2).

Table 2 shows that, in order to obtain results with the FO method that are comparable to those with the IBOSS OBD method, $MI_3 = n$ is insufficient, while the improvements for values of $20n$ and $50n$ are minimal compared to the results for $MI_3 = 10n$. We observed similar behavior for other models and criteria, so that we will set $MI_3 = 10n$. This makes FO more competitive because a larger value of MI_3 comes with a significant increase in computing time, especially for A-optimality. The reason for this is that, even when we set $MI_3 = 50n$, the algorithm improves only slightly and does not reach convergence for A-optimality in the scenarios that we considered. This is in contrast to the performance of IBOSS OBD, which converges in all scenarios.

D.2 About PGD method

The PGD method is a subset selection approach for measurement-constrained regression that targets statistical efficiency via an A-optimality criterion. It is primarily developed for linear models but can be extended to generalized linear models. Table 3 compares the PGD method with IBOSS OBD in terms of the statistical efficiency (A-optimality) of the selected subsets and the corresponding computation time. The set up is as the same as that of Scenario (i), i.e., a linear first-order model for $p = 10$. As shown in Table 3, the PGD method achieves very high statistical efficiency. However, its computational cost is relatively high, particularly for large values of N. Due to these scalability limitations, we excluded PGD from further consideration in subsequent analyses.

D.3 Scenario (i): A linear first-order model for $p = 10$.

Under the D-optimality criterion, the efficiency ratio $\Phi(\xi^*)/\Phi(S^*_{IBOSS\ OBD})$ is 99.999%. Since this represents the lower bound for the efficiency of $S^*_{IBOSS\ OBD}$ (as given in Equation 5), we treat

Table 3: A-optimality criterion for a first-order linear model: mean (std)

(N, n)	(1000, 100)		(10000, 100)	
	PGD	IBOSS OBD	PGD	IBOSS OBD
Eff (%)	98.90 (0.23)	99.74 (0.07)	97.61 (0.50)	99.66 (0.08)
Time (s)	2.07 (0.06)	0.18 (0.11)	69.96 (1.96)	0.71 (0.32)

Table 4: Scenario (i): D-optimality for all parameters ($N = 100000$ and $n = 1000$)

Algorithm	SRS	ODB	LEV	IBOSS	OSS	SEQ	FO	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	41.82	61.84	42.06	72.33	83.69	96.49	100.00	99.67	100.00	100.00
Std Eff (%)	0.60	0.69	0.57	0.54	2.42	0.15	0.00	0.04	0.00	0.00
Mean Time (s)	0.0001	1.9365	0.0465	0.0340	2.6261	0.2428	5.0611	0.1455	0.3385	0.6239
Std Time (s)	0.0000	0.6099	0.0087	0.0096	0.3287	0.0518	0.6286	0.01833	0.0405	0.1604

Mean(size) = 1021.59, Std(size) = 9.94; convergence rates of IBOSS OBD and FO are both 1.00.

it as the true statistical efficiency. Consequently, we report this ratio as the statistical efficiency metric for the subdata obtained by other methods as well.

We first consider D-optimality when all parameters are of interest. From Table 4, LEV shows a modest improvement over SRS, although both methods remain considerably less efficient than ODB. IBOSS outperforms ODB, and OSS further improves upon IBOSS. SEQ and IBOSS+ also demonstrate high efficiency, whereas IBOSS++, IBOSS OBD, and FO achieve near-optimal performance. In terms of computation time, SRS is the fastest, followed by IBOSS and LEV. IBOSS+ requires slightly more time than LEV, followed by SEQ, while IBOSS++ is slower. Among the most efficient methods, IBOSS OBD is substantially faster than FO, although both achieve a 100% convergence rate. In contrast, ODB exhibits a less favorable performance profile, with lower efficiency than several competing methods and considerably higher computational cost. The average subdata size is close to the target of $n = 1000$.

We next consider A-optimality when only the first five parameters are of interest. The results in Table 5 closely resemble those in Table 4, but with two notable differences. First, the proposed methods continue to achieve high efficiency, whereas the competing methods are substantially less efficient than in the D-optimality setting, particularly SEQ and FO, which are no longer effective. This is expected, as most competing methods are designed for settings in which all model parameters are of interest. In contrast, the proposed approach remains effective when attention is restricted to a subset of parameters. Second, the average computation time for FO increases substantially to about 46 seconds, compared with approximately 5 seconds under D-optimality. In contrast, the computation time for IBOSS OBD remains essentially unchanged, and IBOSS OBD converges in all repetitions.

Since ODB does not provide a competitive balance between statistical efficiency and scalability, it is excluded from further consideration in Scenarios (ii) and (iii).

D.4 Scenario (ii): A full second-order logistic regression model with $p = 3$.

A complication is that the information matrix for a logistic regression model depends on the unknown parameters. A common approach to address this is to use a locally optimal design based on a guess for the parameters, either from past experience, a pilot study, or a random sample from the full data. This challenge is the same for all methods, and we will therefore only focus on selecting efficient subdata when the true values of the parameters can be plugged in the information matrix, as is the case for simulations. We set all parameters equal to 1. Under this configuration, the Fisher information matrices are near zero for most observations, creating an ill-conditioned optimization landscape that

Table 5: Scenario (i): A-optimality for the first five slope parameters ($N = 100000$ and $n = 1000$)

Algorithm	SRS	ODB	LEV	IBOSS	OSS	SEQ	FO	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	27.82	57.42	28.40	43.57	53.34	68.83	72.77	98.90	99.98	100.00
Std Eff (%)	0.67	1.11	0.67	0.71	2.19	0.67	0.43	0.15	0.01	0.00
Mean Time (s)	0.0001	1.0670	0.0463	0.0359	2.3201	0.3707	45.9827	0.1618	0.3317	0.6185
Std Time (s)	0.0000	0.2100	0.0116	0.0173	0.3350	0.0481	4.8112	0.0245	0.0589	0.1143

Mean(size) = 928.36, Std(size) = 66.92; convergence rates: IBOSS OBD = 1.00.

Table 6: Scenario (ii): D-optimality criterion for the parameters of the main effects only ($N = 100000$ and $n = 1000$)

Algorithm	SRS	LEV	SEQ	FO	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	9.16	10.04	71.33	73.18	93.05	99.82	100.00
Std Eff (%)	0.67	0.60	0.76	0.56	1.63	0.26	0.00
Mean Time (s)	0.0002	0.0582	0.2009	2.3319	0.1341	0.2837	1.0313
Std Time (s)	0.0000	0.0198	0.0348	0.2831	0.0176	0.0385	0.2577
Mean(size) = 938.54, Std(size) = 41.76; convergence rates of IBOSS OBD is 0.99.							

Table 7: Scenario (ii): A-optimality criterion for all parameters ($N = 100000$ and $n = 1000$)

Algorithm	SRS	LEV	FO	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	16.60	17.11	99.99	96.08	99.30	100.00
Std Eff (%)	0.88	0.51	0.00	1.19	0.02	0.00
Mean Time (s)	0.0002	0.0606	43.9571	0.1563	0.3221	1.5092
Std Time (s)	0.0002	0.0100	1.6908	0.0135	0.0264	0.3608
Convergence rates: FO = 0.00 and IBOSS OBD = 1.00.						

serves as a stress test for the proposed method. Since IBOSS and OSS are designed for first-order linear models, they are not considered for this scenario.

We consider D-optimality when only the parameters of the main effects are of interest. The results in Table 6 show that SRS and LEV, while being the fastest methods, achieve statistical efficiencies of only around 10%. SEQ and FO perform substantially better than SRS and LEV, but remain less competitive than the proposed methods. This is expected, as they are primarily designed for settings in which all model parameters are of interest. IBOSS+ continues to achieve high statistical efficiency while remaining more computationally efficient than SEQ and FO. Both IBOSS++ and IBOSS OBD attain near-perfect statistical efficiencies, with IBOSS OBD converging in 99 out of 100 repetitions.

We consider A-optimality when all parameters are of interest. The SEQ method yields an average subdata size of less than 250, which is well below the target of 1000. Its statistical efficiency is also much lower compared to that for the D-optimality criterion. To obtain a better comparison for the other methods, we excluded SEQ from further consideration so that we could set the subdata size at $n = 1000$ for all other methods.

SRS and LEV remain the fastest methods but have low statistical efficiencies (less than 20%). All remaining methods are highly efficient. IBOSS+ has the lowest statistical efficiency among them (approximately 96%), but is also the fastest with an average computing time of about 0.16 seconds. Both IBOSS++ and FO exhibit statistical efficiencies close to 1, while IBOSS OBD achieves nearly perfect efficiency. IBOSS OBD converges in all repetitions, whereas FO never converges. Despite using $MI_3 = 10n$, this lack of convergence results in significantly longer computing times for FO than for IBOSS OBD. We verified that the same result would have been true for $MI_3 = 50n$, so that our choice of $MI_3 = 10n$ presents FO in the best possible light.

D.5 Scenario (iii): Clusterwise linear model with $p = 10$ and 10 clusters.

Clusterwise linear regression (CLR) models aim to uncover latent clusters within the data, such that within each cluster, the relationship between input and output variables can be adequately modeled by a linear first-order regression model. The model involves G gating functions and G corresponding linear regression models (also known as experts). While each observation is generated from one of these models, it is unknown from which one. CLR models have been widely applied in fields such as social sciences, environmental sciences, and engineering [4, 3, 10]. Research on CLR models remains active, particularly in the development of efficient algorithms to alleviate the high computational demands [17].

Unlike standard linear regression, CLR models do not admit a closed-form expression of the maximum likelihood estimator (MLE). Due to the presence of latent cluster memberships, the Expectation-Maximization (EM) algorithm is the primary tool for estimating model parameters. As a result, the computational cost of fitting a CLR model can be substantial, especially for large datasets. A subdata

Table 8: Scenario (iii): D-optimality criterion for all parameters ($N = 100000$ and $n = 10000$)

Algorithm	SRS	SEQ	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	53.39	97.03	99.87	100.00	100.00
Std Eff (%)	0.26	0.13	0.02	0.00	0.00
Mean Time (s)	0.0002	2.4255	1.7737	89.8514	175.9158
Std Time (s)	0.0000	0.1496	0.2002	5.3997	9.7151
Mean(size) = 7881.3, Std(size) = 104.39; convergence rate of IBOSS OBD = 1.00.					

approach can help reduce this burden. However, the absence of a closed-form expression for the information matrix for a CLR model complicates the application of optimal design techniques for subdata selection.

[13] showed that, under certain regularity conditions, the following matrix for the ith data point can be used for subdata selection:

$$\begin{pmatrix} \pi_1 \frac{\boldsymbol{x}_i \boldsymbol{x}_i^T}{\sigma_1^2} & & \mathbf{0} & \\ & \pi_2 \frac{\boldsymbol{x}_i \boldsymbol{x}_i^T}{\sigma_2^2} & & \\ & & \ddots & \\ \mathbf{0} & & & \pi_G \frac{\boldsymbol{x}_i \boldsymbol{x}_i^T}{\sigma_G^2} \end{pmatrix}, \tag{44}$$

where $\boldsymbol{x}_i$ is the feature vector for the ith data point, π_g is the probability that the ith data point belongs to cluster g, and σ_g^2 is the variance of the error term for the gth linear model. The target matrix, which replaces the information matrix, is the matrix in Equation (44) summed over the indices i that belong to the subdata.

Note that the matrix in (44) is block diagonal, which makes the resulting optimization problem algebraically equivalent to that for the first-order linear model in Scenario (i). Nevertheless, to illustrate the capability of the proposed method in more complex settings, we work directly with the target matrix in (44).

In our setup, we assume $G = 10$ clusters and $p = 10$ features, making the dimension of the matrix in Equation (44) equal to 110. Here, π_g and σ_g^2, $g = 1, \ldots, G$, are unknown parameters, just as the intercepts and slopes of the 10 linear models. As in scenario (ii), this approach requires some information about the parameters that appear in this matrix, that is, π_g and σ_g^2 values. Since this is the same for all methods, we will again use known values in our simulations, and focus on selecting efficient subdata. Specifically, we set $\sigma_g^2 = 1$ and generate π_g values independently from a uniform distribution for $g = 1, \ldots, G$. The generated π_g values are then normalized to ensure that $\sum_{g=1}^{G} \pi_g = 1$.

For this relatively more complex setting, we exclude the LEV and FO methods because they depend on a specific matrix structure that is not directly applicable here. The SEQ approach starts by selecting the first $5 \times 110 = 550$ data points. With a target subdata size of 1000, it would only allow judicious selection of 450 data points. Therefore, to give SEQ a better chance to shine, we consider a subdata size of 10000 in this scenario.

Under the D-optimality criterion in Table 8, SRS achieves a statistical efficiency of 53% and is the fastest method. SEQ performs well, with a statistical efficiency of 97%. IBOSS+ outperforms SEQ, achieving near-perfect statistical efficiency while requiring less computing time. Both IBOSS++ and IBOSS OBD exhibit nearly perfect statistical efficiency, but at the cost of significantly longer computing times. In particular, IBOSS OBD converges in all repetitions, which ensures high average statistical efficiency. Additionally, we observe that the average size of the selected subdata is less than 8000, which is considerably below the target size of 10000.

For the A-optimality criterion, just as in Scenario (ii), the average subdata size selected by the SEQ approach falls to less than 1000, far below the target of 10000. As a result, SEQ is also very inefficient for the same reason as explained in the previous paragraph. Consequently, SEQ is excluded from further consideration, and the subdata size is fixed at 10000 for the other methods.

Table 9 shows that the performance of all methods is similar to that for the D-optimality criterion. In particular, IBOSS+ maintains high statistical efficiency while requiring significantly less computing

Table 9: Scenario (iii): A-optimality criterion for all parameters ($N = 100000$ and $n = 10000$)

Algorithm	SRS	IBOSS+	IBOSS++	IBOSS OBD
Mean Eff (%)	53.58	99.41	100.00	100.00
Std Eff (%)	0.24	0.03	0.00	0.00
Mean Time (s)	0.0003	1.9989	154.8430	305.8149
Std Time (s)	0.0001	0.1740	9.7072	13.3035
Convergence rate of IBOSS OBD = 1.00				

time than IBOSS++ and IBOSS OBD, which achieve nearly perfect statistical efficiencies. IBOSS OBD again converges in all repetitions, ensuring the reliability of its results.

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: We develop a general, convergence-guaranteed algorithm grounded in optimal approximate design theory that yields near-optimal subdata across arbitrary N, n and optimality criteria. Our framework provides, for the first time, tight lower and upper bounds on the statistical efficiency of subdata selected by any method. This benchmarking capability reveals that many existing methods operate substantially below their theoretical potential, while our methodology outperforms them all.

   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: Please see Section 5.

   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: The full set of assumptions are provided in the lemmas and theorems. The complete proofs of each lemma and theorem are provided in the Appendix.

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 give the details how to reproduce the experimental results in Sections 3 and 4.

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 submit the Julia codes and instruction of reproducing all experimental results in a zip file.

   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 give the details how the experiment is set up in Section 4.

   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: Our tables in Appendix include standard deviations across 100 runs.

   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: Described in Section 4.

   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: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: Described in Section 5.

    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: The paper poses no such risks.

    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 compare with several existing algorithms and cite the original papers.

    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: We submit the code for reproducing our empirical results.

    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: The 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: The 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: This research 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.