Technometrics

A Journal of Statistics for the Physical, Chemical, and Engineering Sciences

Taylor & Francis
Taylor & Francis Group

Technometrics

ISSN: 0040-1706 (Print) 1537-2723 (Online) Journal homepage: www.tandfonline.com/journals/utch20

Multi-Layer Sliced Design and Analysis with Application to AI Assurance

Qing Guo, Xinwei Deng & Peter Chien

To cite this article: Qing Guo, Xinwei Deng & Peter Chien (02 Sep 2025): Multi-Layer Sliced Design and Analysis with Application to AI Assurance, Technometrics, DOI: 10.1080/00401706.2025.2537033

To link to this article: https://doi.org/10.1080/00401706.2025.2537033

View supplementary material

Published online: 02 Sep 2025.

Submit your article to this journal

Article views: 77

View related articles

View Crossmark data

TECHNOMETRICS
2025, VOL. 00, NO. 0, 1–13
https://doi.org/10.1080/00401706.2025.2537033

Taylor & Francis
Taylor & Francis Group

Multi-Layer Sliced Design and Analysis with Application to AI Assurance

Qing Guo[a], Xinwei Deng[a], and Peter Chien[b]

[a]Department of Statistics, Virginia Tech, Blacksburg, VA; [b]Department of Statistics, University of Wisconsin-Madison, Madison, WI

ABSTRACT
Enhancing AI assurance in tuning configurations and hyper-parameters of AI algorithms is an important problem in many applications. This work provides an experimental design method to address this challenging problem. The key idea of the method is to conduct an efficient experimental design to detect and quantify the effects of hyper-parameters on the performance of AI algorithms. Specifically, the method proposes a multi-layer sliced design to enable quantifying the effects of slice factors and design factors to account for hyper-parameters having different effects under different configurations of the AI algorithm. Moreover, this method develops an effective analysis procedure to estimate the effects of these factors and test their significance. The performance of the proposed design and analysis methods is successfully illustrated by simulation studies and real-world AI applications.

ARTICLE HISTORY
Received July 2023
Accepted July 2025

KEYWORDS
Deep learning; Hyper-parameters; Induced Lasso; Sliced design

1. Introduction

In the modern design of experiment applications, including online experiments and hyper-parameter tuning for AI algorithms, several factors of particular interest often exist in comparison with other design factors. For example, (Sadeghi, Chien, and Arora 2020) considered an experiment on how to construct online designs of website layouts across multiple platforms such as laptops, cellphones, and iPads. In their work, the factor of “platform” is identified as a *slice factor* and other factors related to the website layout are considered as *design factors*. We note that the slice factors differ from design factors in the sense that the experimenters are interested in estimating the effects of design factors under different levels of the slice factor. The work of Sadeghi, Chien, and Arora (2020) mainly considers a sliced design with a single slice factor. However, many applications have multiple slice factors of interest. For example, a retail company would like to conduct online experiments of web advertisements across different user devices (e.g., desktops, cellphones) and apps (e.g., Facebook and Instagram). Such experiments will involve two slice factors: user devices and social-media apps. It is important to evaluate the impact of advertisements under every combination of the slice factors. Another example is that AI assurance faces the challenge of exploring the effects of hyper-parameters on model performance. Oftentimes, the effect of hyper-parameters on the model performance can be different across different models and optimization methods in the AI algorithm. From this viewpoint, we consider the modeling choices and optimization methods used in the AI algorithm can be considered as two slice factors, and other hyper-parameters in the neural network of the AI algorithm as design factors.

In the area of AI assurance (Batarseh, Freeman, and Huang 2021; Batarseh and Freeman 2022; Batarseh, Chandrasekaran, and Freeman 2023), it is important to investigate the effects of hyper-parameters on the model performance of AI algorithms under different configurations (e.g., different level combinations of model choice and optimization method). Investigating the effects of hyper-parameters on the AI algorithm has attracted considerable interest (Snoek, Larochelle, and Adams 2012; Bergstra and Bengio 2012; Bardenet et al. 2013; Li et al. 2020a, 2020b). To design an appropriate experiment to evaluate the performance of the AI algorithm, a suitable design is needed to make the effects of hyper-parameters estimable under each configuration, which is closely related to the concept of the sliced design strategy (Sadeghi, Chien, and Arora 2020).

In this work, we propose a *multi-layer sliced design* (MLSD) to deal with multiple slice factors, with application to the investigation of the effect of hyper-parameters on the AI algorithm under different configurations. In this application, the factors involving configurations are considered slice factors. Multiple slice factors can have different importance or a hierarchical structure. Specifically, we consider the MLSD with each factor at two levels and propose the ordered word-length pattern for finding the ordered minimum aberration design for MLSD. Our proposed criterion is flexible in dealing with both equal importance and ordered importance of the slice factors. Moreover, we also develop a novel analysis method to obtain a parsimonious model by leveraging the sparsity principle (Box, Hunter, and Hunter 1978; Wu and Hamada 2021) in the design of experiments (Yuan, Joseph, and Lin 2007; Seeger, Steinke, and Tsuda 2007; Dougherty et al. 2015). We further enable the hypothesis testing of significant effects among a variety of main effects, two-factor interaction

CONTACT Xinwei Deng xdeng@vt.edu Department of Statistics, Virginia Tech, Blacksburg, VA 24061.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

effects. The proposed MLSD has a stronger estimation capability for slice factors, and it can estimate the corresponding factorial effects accurately. Recent work on sliced experimental design (Sadeghi, Chien, and Arora 2020) considered multiple submodel estimations for each platform. In contrast, our proposed analysis method can estimate these effects simultaneously by adopting the induced Lasso technique (Cilluffo et al. 2020). The proposed MLSD framework has practical applications beyond the investigation of hyper-parameters in AI algorithms. It can also be used in other aspects of AI assurance, such as investigating the robustness of AI algorithms.

The remainder of the article is organized as follows: In Section 2, we briefly review the factorial design and its use in AI-related applications. Section 3 details the proposed multi-layer sliced design. In Section 4, we focus on the analysis method for estimation of the effects of interest based on MLSD. Section 5 conducts simulations to examine the performance of the proposed design and analysis method. Section 6 presents a practical application of our method in AI assurance. Finally, we conclude the article with some discussions in Section 7.

2. Literature Review

Hyper-parameter tuning for AI algorithms is important(Mantovani et al. 2016; Lee, Park, and Sim 2018; Probst, Wright, and Boulesteix 2019) but often costly in practice (Hutter, Kotthoff, and Vanschoren 2019). Traditionally, this is mainly a manual process heavily relying on the experience of investigators. For simple settings with one or two hyper-parameters involved (e.g., bandwidth parameter selection for kernel learning) in the AI algorithm, straightforward exhaustive approaches such as grid search usually work well (Bergstra et al. 2011). However, this simple approach quickly becomes impractical when there are a large number of hyper-parameters, where Monte Carlo approaches including random search (Bergstra and Bengio 2012) and the one-factor-at-time procedure are often used. Such methods are unfortunately ineffective for high-dimensional hyper-parameters or can miss the interaction between different hyper-parameters. Methods such as genetic algorithm (GA) (Lessmann, Stahlbock, and Crone 2005) and particle swarm optimization (PSO) (Lorenzo et al. 2017) are also used as heuristics to prioritize the settings of hyper-parameters. Recently, Bayesian optimization has gained great attention by its effectiveness, especially in complex models such as deep neural networks (Eggensperger et al. 2013; Feurer, Springenberg, and Hutter 2015; Klein et al. 2017) and knowledge transfer (Yogatama and Mann 2014; Joy et al. 2016). Many existing Bayesian optimization techniques face a common challenge. The acquisition criterion is often non-convex and potentially non-differentiable, making it difficult for standard local numerical optimization methods to find the optimal solution reliably. Recent work has explored Delaunay triangulation to address this challenge (Gramacy, Sauer, and Wycoff 2022).

When emphasizing the main effects and two-factor interaction effects, one can exploit fractional factorial designs (Box and Hunter 1961; Gunst and Mason 2009; Wu and Hamada 2021) to use a small number of experimental trials (i.e., a level combination of factors) to adequately estimate the effects up to the second order. We use ideas from the design of experiments literature, the hyper-parameter tuning can be investigated from a new and different angle. By considering the hyper-parameters with possible discrete values, the factorial design can be applied to estimate the effects of different hyper-parameters (Cheng 2016; Kittitharayada et al. 2021). Due to limitations on resources, the fractional factorial design aims at economically investigating the cause-and-effect relationships (Box and Hunter 1961; Gunst and Mason 2009). It allows for more efficient use of resources by reducing the number of experiments. To find optimal fractional factorial designs, a widely used criterion is the maximum resolution criterion (Box and Hunter 1961) and the minimum aberration criterion (Box and Hunter 1961; Fries and Hunter 1980; Tang and Wu 1996), both of which are based on using the word-length pattern (Fries and Hunter 1980; Cheng, Steinberg, and Sun 1999; Wu and Hamada 2021).

In the direction of using fractional factorial designs for novel applications, Sadeghi, Chien, and Arora (2020) proposed the sliced design for the multi-platform online experiments. Their research focused on identifying the optimal sliced design through the sliced minimum aberration criterion, and they developed linear models to estimate all effects. Chang (2022) provides theoretical support for the sliced minimum aberration design from the view of Bayesian analysis. However, the sliced design is not readily applicable to scenarios with multiple slicing factors. Additionally, their methodology requires separate modeling and estimation processes for different platforms.

The sliced design approach proposed by Sadeghi, Chien, and Arora (2020), which treats factors differently, has connections with other existing methodologies in the literature. For instance, in a split-plot design (Jones and Nachtsheim 2009; Wu and Hamada 2021), the whole plot factors are assigned to main plots, and subplot factors are applied within these subplots (Fisher 1970). Similarly, robust parameter designs explore interactions between control factors and noise factors (i.e., uncontrollable variables) (Taguchi 1987). The branching and nested design (Phadke 1995; Hung, Joseph, and Melkote 2009) includes branching and nested factors and the nested factors differ for the levels of branching factors.

3. Multi-Layer Sliced Design

This section details the proposed multi-layer sliced design (MLSD). In the MLSD, we consider two classes of factors, slice (platform) factors, and design factors, as shown in Table 1. We denote the slice factors as $S_i, i = 1, \ldots, m$, where each S_i has l_i levels. The design factors are $X_j, j = 1, \ldots, k$, where each X_j has h_j levels. Different levels of combinations of design factors are to be conducted to understand the effects of design factors on the response (i.e., experiment outputs). The slice factors are often of great importance to be considered as platform effects, that is, the experimenter expects that the effect of design factors can vary according to the different settings of slice factors. It is

Table 1. Factors in the multi-layer sliced design.

	Slice factors			Design factors		
	S_1	…	S_m	X_1	…	X_k
Number of levels	l_1	…	l_m	h_1	…	h_k

important to distinguish the roles of the slice factors and design factors in both design criterion and data analysis. Additionally, considering multiple slice factors allows for the incorporation of more complex statistical models that can address both the main effects of each individual factor and their interaction effects, thereby enhancing the flexibility and depth of the analysis.

For example, for a two-layer sliced design in a webpage layout application, online platforms can be regarded as slice factors. The S_1 can be electronic devices such as cell phones and laptops for web browsing. And S_2 is the social apps such as Instagram or Facebook used by customers. Different advertisement designs under the ith level of S_1 and jth level of S_2 can be constructed as $\mathbf{D}_{ij}$ for the design factors $X_1, \dots, X_k$. The $\mathbf{D}_{ij}$ can be a full factorial design or a fractional factorial design, while all level combinations of the slice factors will be considered. Using such a multi-layer sliced design, the experimenter can investigate how the slice factors and design factors affect customer shopping behaviors.

For ease of presentation, we will start with our proposed method under the two-layer sliced design with $m = 2$. The presented definition, properties, and analysis methods can be extended to multi-layer sliced design with $m \geq 3$.

Definition 1 (Two-layer sliced Design). Consider two slice factors S_1 with l_1 levels and S_2 with l_2 levels and k design factors $X_1, X_2, \dots, X_k$. The whole design for slice factors and design factors, denoted as $\mathbf{D}$, consists of subdesigns, $\mathbf{D}_{11}, \dots, \mathbf{D}_{1,l_2}, \dots, \mathbf{D}_{l_1,1}, \dots, \mathbf{D}_{l_1,l_2}$ associated with each level combination of slice factors.

Figure 1 presents the design set $\mathbf{D}$ of the experiment in Definition 1. When one considers both the slice factor and design factor with two levels, a full factorial two-layer sliced design $\mathbf{D}$ can be denoted as $2^2 2^k$. To reduce the run size, especially in a situation with a large number of design factors, we would consider the $\mathbf{D}_{ij}$ to be the fractional factorial design. Consequently, the whole MLSD design $\mathbf{D}$ will also be a fractional factorial design. To enable the investigation of how the design factors affect the response under different level combinations of the slice factors, a suitable two-layer sliced design should have the following two characteristics:

(i) The subdesigns $\mathbf{D}_{ij}, i = 1, \dots, l_1; j = 1, \dots, l_2$ should attain a preferable estimation for the effects of design factors. The selected subdesigns should have enough estimation capabilities for the main effects of the design factors.

(ii) The whole design $\mathbf{D}$ for slice factors and design factors can estimate the effects of slice factors and the two-way interaction effects between slice factors and design factors.

To formalize these properties, we categorize the factorial effects into two distinct sets based on their relevance to the slice and design factors. Let E_I be the set of all factorial effects with words that exclude slice factor S_1 and S_2 (e.g., A, AB, ABC), and E_S be the set of all factorial effects with words that include the slice factor S_1 or S_2 (e.g., AS_1, ABS_2, $ABCS_1S_2$). The design properties (i) and (ii) translate to the following goals for E_I and E_S:

Property (i): The subdesigns $\mathbf{D}_{ij}$ should ensure that the main effects in E_I can be estimated.

Property (ii): The whole design $\mathbf{D}$ should allow the estimation of the main effects, the two-factor interactions in E_S.

To construct the MLSD design with the above properties, we need to differentiate the importance of different effects. Without loss of generality, a factorial effect can be expressed as a word consisting of slice factors (e.g., S_1, S_2) and design factors (e.g., $A, B, C, \dots$). Furthermore, we consider the importance of slice factors in two situations: (i) one of the slice factors is more important than the other slice factor (e.g., $S_1 \succ S_2$ or $S_2 \succ S_1$); and (*ii*) two slice factors have equal importance ($S_1 \triangleq S_2$). When $S_1 \succ S_2$, we denote S_1 to be the primary slice factor and S_2 to be the secondary slice factor. Specifically, we propose the following hierarchy principle for the MLSD design.

Principle 1 (Effect hierarchy for MLSD design). The ordering of importance for effects is determined by the following rules:

(i) For the union of E_I and E_S, lower-order effects are more important than higher-order effects.
(ii) For E_I, effects of the same order are equally important.
(iii) For E_S, effects with the primary slice factor are more important than the ones with a secondary slice factor of the same order. If slice factors are the same important, effects of the same order are equally important.
(iv) Any effect in the set E_S is more important than an effect in E_I with the same order.

The hierarchy principle serves as a guideline to construct MLSD designs that balance the importance of estimating effects in E_S and E_I. Next, we will establish some criteria to compare different MLSD designs given the number of slice factors and design factors. To facilitate our discussion, we will consider all slice factors and design factors at two levels. Following Definition 1 with both S_1 and S_2 having two levels, we consider the fractional factorial design for the whole MLSD design $\mathbf{D}$ as $2^2 \cdot 2^{k-p}$. Here p represents that the run size of $\mathbf{D}$ is a 2^{-p}th fractional of the full factorial design of slice factors and design factors. For simplicity, we will demonstrate the MLSD design at two levels for both slice factors and design factors as a $2^{2+(k-p)}$ design. In a fractional factorial design, the defining relation is essential for constructing word-length patterns (Cheng 2016; Wu and Hamada 2021). These patterns allow statisticians to assess

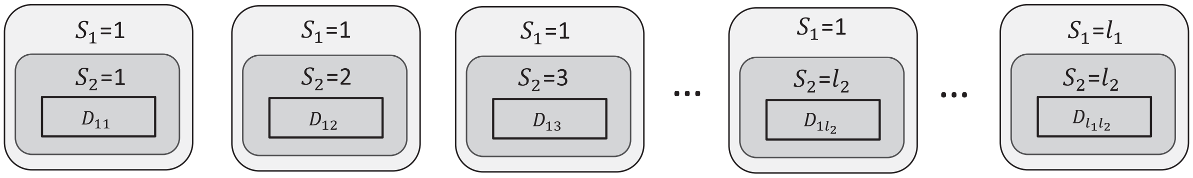

Figure 1. An illustration of a two-layer sliced design in Definition 1.

the capability of a design to distinguish between the effects of various factors and their interactions. In addition, word-length patterns serve as a measure of a design's resolution, effectively indicating the extent of confounding among factors. Similarly, it is essential to establish criteria that guide the selection of designs for MLSD. To compare different $2^{2+(k-p)}$ MLSD designs, we introduce the concept of *sliced defining relations* for the two slice factors. In the MLSD, slice factors are of important interest, thus their defining relation takes priority over other defining relations. For the design with one slice factor, the sliced defining relation of design is composed of the slice factor S and the group of its aliasing effects. It is formed by multiplying the defining relation of design by the slice factor S (i.e., $S = ABCS$ is obtained from $I = ABC$) (Sadeghi, Chien, and Arora 2020). We note that S is aliased with the effects $ABCS$. The word ABC is called the generator of the design and it has length 3. In the two-layer situation, we will have the sliced defining relation for each slice factor. For example, we will have two sliced defining relations $S_1 = ABCS_1$ and $S_2 = ABCS_2$ for a two-layer sliced design. Correspondingly, there are separate word-length patterns for different slice factors. Next, we will define the sliced word-length pattern.

Definition 2 (Sliced word-length pattern). The sliced defining relations of $\mathbf{D}$ are the aliasing relations involving slice factors $S_i, i = 1, 2$. The sliced word-length pattern is

$$\mathrm{SW} = \{\mathrm{SW}_1, \mathrm{SW}_2\},$$

where SW_i represents the set of word-length counts for the sliced relation of slice factor S_i. Specifically,

$$\mathrm{SW}_i = \left\{(3^{B_{S_i,3}}), \ldots, ((k+2)^{B_{S_i,k+2}})\right\},$$

where $B_{S_i,j}$ denotes the number of effects with length j for the sliced relation of S_i, and k is the number of design factors.

The number of sliced defining relations equals the number of slice factors. The sliced defining relations can be derived from multiplying the defining contrast subgroup of $\mathbf{D}$ by the slice factors. For illustration, consider a 2^{2+3-1} MLSD. Assume that the whole design $\mathbf{D}$ is a 2^{5-1} fractional factorial design that consists of four subdesigns, each of which is a 2^{3-1} fractional factorial design for design factors. We now consider three strategies to construct an MLSD design. $d^{(1)}$: $I = ABC$; $d^{(2)}$: $I = ABCS_1S_2$; $d^{(3)}$: $I = ABCS_2$. For $d^{(1)}$, the sliced defining relations are $S_1 = ABCS_1$ and $S_2 = ABCS_2$. The sliced word-length pattern is $SW = \{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$. For $d^{(2)}$, the sliced defining relations are $S_1 = ABCS_2$ and $S_2 = ABCS_1$. The sliced word-length pattern is $SW = \{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$. For $d^{(3)}$, the sliced defining relations are $S_1 = ABCS_1S_2$ and $S_2 = ABC$. The sliced word-length pattern is $SW = \{(3^0, 4^0, 5^1), (3^1, 4^0, 5^0)\}$.

Since the slice factors S_1 and S_2 can be of different importance, we can order the two sliced relations and define the ordered sliced word-length pattern as follows.

Definition 3. With the ordered slice factors, the ordered sliced word-length pattern can be defined as follows.

(a) When slice factor S_1 and S_2 are equally important (i.e., $S_1 \triangleq S_2$), then the ordered sliced word-length pattern grouping is defined by the combination of the sliced word-length patterns for each factor:

$$SW = \prod_{i=1}^{2} SW_i = SW_1 \times SW_2,$$

where $SW_i = \{(j+2)^{B_{S_i,j+2}} : j = 1, 2, \ldots, k\}$ for each slice factor S_i. Note here that the product operation on the sliced word-length patterns can be extended to any number of layers, not just the two-layer case presented here.

(b) When one slice factor S_i is more important than another $S_{i'}$ (i.e., $S_1 \succ S_2$ or $S_2 \succ S_1$), then the ordered sliced word-length pattern grouping is ordered by importance:

$$SW = \{\underbrace{\mathrm{SW}_i}_{\text{Part 1}}, \underbrace{\mathrm{SW}_{i'}}_{\text{Part 2}}\}.$$

When the slice factors are equally important, the sliced word-length pattern only has one part. For other situations, it will contain several parts, and the number of parts is determined by the number of slice factors. With the defined ordered sliced word-length pattern, we can continue to compare two design strategies $d^{(2)}$ with $I = ABCS_1S_2$ and $d^{(3)}$ with $I = ABCS_2$ in the 2^{2+3-1} MLSD. If $S_1 \succ S_2$, the ordered sliced word-length pattern for $d^{(2)}$ is $\{(3^0, 4^1, 5^0), (3^0, 4^1, 5^0)\}$ and that for $d^{(3)}$ is $\{(3^0, 4^0, 5^1), (3^1, 4^0, 5^0)\}$. When $S_1 \triangleq S_2$, the word-length pattern of $d^{(2)}$ is $SW = \{(3^0, 4^2, 5^0)\}$ and the word-length pattern of $d^{(3)}$ is $SW = \{(3^1, 4^0, 5^1)\}$.

Next, we develop proper criteria to compare the MLSD. The maximum resolution (Box and Hunter 1961) and minimum aberration (Fries and Hunter 1980) are two popular criteria for selecting the optimal design. We now extend resolution and aberration to accommodate ordered slice factors and propose the ordered sliced resolution as follows:

Criterion 1 (Ordered Sliced Resolution). The ordered sliced resolution of a $2^{2+(k-p)}$ complete design $\mathbf{D}$ is defined to be the smallest j such that $B_{S_i,j} \geq 1$ or $B_{S_1,j} + B_{S_2,j} \geq 1$ in Part 1 based on Definition 3.

According to the sliced hierarchy principle, a suitable design is to maximize the ordered resolution. The design with a large resolution can ensure the capability to estimate important slice factors and their interaction with design factors. Here, we use sliced defining relation and sliced word pattern to find a sliced minimum aberration design. The objective is to minimize the aliasing of slice factors with higher-order effects, thereby preserving their estimability.

Criterion 2 (Ordered Sliced Minimum Aberration). Suppose that two $2^{2+(k-p)}$ MLSD $\ddot{d}$ and $\tilde{d}$ are to be compared. Let r be the smallest integer such that $\sum_{S_i} B_{s_i,r}(\ddot{d}) \neq \sum_{S_i} B_{s_i,r}(\tilde{d})$, S_i are all the primary slice factors. Design $\ddot{d}$ is said to have less sliced aberration if $\sum_{S_i} B_{s_i,r}(\ddot{d}) < \sum_{S_i} B_{s_i,r}(\tilde{d})$. If there is no design with less sliced aberration than $\ddot{d}$, then $\ddot{d}$ is called a sliced minimum aberration design.

To construct a sliced minimum aberration design, the length of effects in sliced defining relation plays a key role.

Next, we will establish a property to determine the number of effects containing the secondary slice factor in the defining relation (i.e., defining contract subgroup) for the case of $S_1 \succ S_2$. The secondary slice factor S_2 can help extend the length of the words in the primary sliced defining relation.

Theorem 1. If $S_1 \succ S_2$, then for a $2^{2+(k-p)}$ MLSD $\mathbf{D}$, the largest number of effects in defining relation that can contain the secondary slice factor (S_2) is 2^{p-1}.

Based on the above theorem, the number of effects containing the secondary slice factor in the defining relation can be determined. It will help identify the minimum aberration design. For a concrete example, consider an MLSD 2^{2+6-2} with slice factors S_1, S_2 and design factors A, B, C, D, E, F. The minimum aberration scheme of fractional factorial design 2^{6-2} for design factors A, B, C, D, E, F is $I = ABCD = CDEF = ABEF$. If $S_1 \succ S_2$, the best defining relation is $I = ABCDS_2 = CDEFS_2 = ABEF$. We can find the defining relation that contains a secondary slice factor is $2^{2-1} = 2$. By Theorem 1, there is no other defining relation that can increase the number of effects containing the secondary slice factor. This means that the primary sliced defining relation can get the minimum aberration design.

Although we present the above results under the two-layer sliced design, the definition, criterion, and properties for the multi-layer sliced design can be extended and generalized. Similarly, multiple slice factors $S_1, \ldots, S_m$ can be ordered, such as $S_1 \prec S_2 \prec \cdots \prec S_m$, $S_1 \triangleq S_2 \triangleq \cdots \triangleq S_m$, $S_1 \succ S_2 \succ \cdots \succ S_m$, and $S_1 \succ \cdots \succ S_i \triangleq \ldots \triangleq S_j \succ \cdots \succ S_m$. Next, we will discuss several properties of the multi-layer sliced design.

Proposition 1. In the multi-layer sliced design $2^m 2^{(k-p)}$ with one or two primary slice factors, the corresponding sliced minimum aberration design can be obtained by not including the primary slice factors in the defining relation.

The statement in Proposition 1 aligns with Theorem 1. In a two-layer sliced design, as described in Theorem 1, the objective of achieving sliced minimum aberration is to incorporate a larger number of secondary slice factors in the defining relation. This approach aids in obtaining a suitable primary sliced defining relation. Moreover, Proposition 1 offers valuable guidance for the exploration of sliced minimum aberration designs in general. The following remarks serve as useful guidance for finding a sliced minimum aberration design.

Remark 1. In the multi-layer sliced fractional factorial design $2^m 2^{k-p}$, there exists a sliced word-length pattern by $(3^{B_{S,3}}, \ldots, (k+2)^{B_{S,k+2}})$ for each slice factor. The design and its properties are determined by the grouping of the ordered sliced word-length pattern set $\{(3^{B_{S_1,3}}, \ldots, (k+2)^{B_{S_1,k+2}}), (3^{B_{S_m,3}}, \ldots, (k+2)^{B_{S_m,k+2}})\}$.

The estimation capability of the design is determined by the grouping of the ordered sliced word-length pattern. In the MLSD with $m > 1$, the order of the importance of the slice factors affects how to determine the sliced word-length pattern and search for the sliced minimum aberration design.

Remark 2. In the multi-layer sliced fractional factorial design $2^m 2^{k-p} (m > 2, m > p)$ with slice factors of the same importance, the sliced minimum aberration design can be obtained from the defining relation of design 2^{m-p} and design 2^{k-p} with minimum aberration.

As an illustration, consider a $2^5 2^{5-2}$ MLSD with slice factors $S_1, \ldots, S_5$ and design factors A, B, C, D, E. A minimum aberration design for slice factors is $I = S_1S_2S_4S_5 = S_1S_2S_3 = S_3S_4S_5$. In this case, the aliasing effects in the defining relation can be arranged in a sequence from long to short. Similarly, a minimum aberration design for design factors is $I = ABC = CDE = ABDE$. However, in this defining relation, the sequence of aliasing effects should be ordered from short to long. Combining these two defining relations, we can obtain the optimal defining relation for the MLSD as follows: $I = S_1S_2S_4S_5ABC = S_1S_2S_3CDE = S_3S_4S_5ABDE$.

Remark 3. In the multi-layer sliced fractional factorial design $2^m 2^{k-p}$, if slice factors are of the same importance, the sliced minimum aberration design corresponds to the design with sliced defining relations where all words contain slice factors.

Consider a $2^2 2^{6-2}$ MLSD. We will examine two design strategies, denoted as $d^{(1)}$ and $d^{(2)}$. For $d^{(1)}$, the defining relation is given by $I = ABCD = CDEF = ABEF$, and the sliced defining relations are $S_1 = ABCDS_1 = CDEFS_1 = ABEFS_1$ and $S_2 = ABCDS_2 = CDEFS_2 = ABEFS_2$. The corresponding sliced word-length pattern is $(4^0, 5^6, 6^0)$. On the other hand, for $d^{(2)}$, the defining relation is $I = ABCDS_2 = CDEFS_2 = ABEF$, and the sliced defining relations are $S_1 = ABCDS_1S_2 = CDEFS_1S_2 = ABEF$ and $S_2 = ABCD = CDEF = ABEFS_2$. The sliced word-length pattern associated with $d^{(2)}$ is $(4^3, 5^1, 6^2)$. Considering the scenario where $S_1 \succ S_2$, we have demonstrated that $d^{(2)}$ represents the sliced minimum aberration design. However, as $S_1 \triangleq S_2$, it becomes evident that $d^{(2)}$ is not the sliced minimum aberration design since there exists a design $d^{(1)}$ with a smaller aberration. In $d^{(2)}$, we observe that there are several words in the sliced defining relation that lack slice factors. Additionally, we can establish that $d^{(1)}$ is a minimum sliced aberration design where all words in its sliced defining relation contain slice factors.

4. The Estimation Method

In this section, we introduce an estimation procedure aimed at identifying significant effects. To streamline the model estimations and avoid the complexity of multiple sub-models, we propose the use of conditional effects. This approach is supported by several studies that have explored conditional effects with minimum aberration (Mukerjee, Wu, and Chang 2017; Chang 2023). In the multi-layer sliced design, we assume that the slice factors $S_1, \ldots, S_m$, and design factors $X_1, \ldots, X_m$ all have two levels, coded as "−1" and "1" (Wu and Hamada 2021).

Let x_i, for $i = 1, \ldots, k$, and s_j, for $j = 1, \ldots, m$, be the corresponding values of X_i and S_j, respectively. The conditional value of X_i given $S_1 = s_1^*, \ldots, S_m = s_m^*$ can be defined as follows:

$$X_i|_{S_1=s_1^*,\dots,S_m=s_m^*} = \begin{cases} x_i & \text{if } s_i = s_i^* \text{ for } i = 1,\dots,m, \\ 0 & \text{otherwise.} \end{cases}$$

Due to the limited experimental run size, we will focus on the estimable main effects and two-factor interaction effects by considering a linear model as follows:

$$y = \beta_0 + \sum_{i=1}^{k} \sum_{s_1=0}^{1} \cdots \sum_{s_m=0}^{1} \beta_{X_i s_1 \dots s_m} (X_i | S_1 = s_1, \dots, S_m = s_m)$$
$$+ \sum_{i=1}^{m} \beta_{S_i} S_i + \sum_{\substack{i=1 \\ i!=j}}^{m} \sum_{j=1}^{m} \beta_{S_i S_j} S_i S_j + \sum_{i=1}^{k} \sum_{j=1}^{m} \beta_{X_i S_j} X_i S_j + \epsilon,$$

where $\boldsymbol{\beta} = (\beta_0, \dots, \beta_{X_i s_1 \dots, s_l}, \dots, \beta_{S_1}, \dots, \beta_{S_i S_j}, \dots, \beta_{X_i S_j}, \dots)$ are the coefficients of the linear model and ϵ is the error term with $\epsilon \sim N(0, \sigma^2)$. The set of coefficients can be simply written as $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_q)$, where q is total number of coefficients. The total number of conducted experiments is n. The vector of response can be denoted as $\boldsymbol{y}$ and the corresponding regression matrix can be written as $\boldsymbol{X}$. In this situation where $n < q$, one possible method for analysis and inference is Lenth's method (Lenth 1989). However, it may not yield accurate parameter estimates for our setting. Some assumptions underlying Lenth's method—such as equal variance among factorial effects and a regular design structure-may not be valid for our experimental designs. According to the sparsity principle in experimental design (Wu and Hamada 2021), it is assumed that only a small of factorial effects will significantly influence the outcome. Therefore, we consider estimating the parameters using the Lasso method (Tibshirani 1996) by minimizing the penalized least squares as

$$L(\boldsymbol{\beta}) = \frac{1}{2} ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_1, \qquad (1)$$

where $||\boldsymbol{\beta}||_1$ is the l_1 norm of $\boldsymbol{\beta}$ and $\lambda \geq 0$ is a tuning parameter. Here we adopt the AIC for the choice of tuning parameter λ (Shao 1997). Due to the non-smoothness of the l_1 norm, the objective function in (1) is not differentiable at zero with respect to $\boldsymbol{\beta}$, which makes it difficult to obtain the derivative of the parameter $\boldsymbol{\beta}$ at all points. Thus one cannot directly apply the sandwich formula to obtain the variance of $\hat{\boldsymbol{\beta}}$ for inference. The work of Cilluffo et al. (2020) incorporates the idea from the induced smoothing (Brown and Wang 2005) to allow estimation and inference on the model coefficients in the Lasso regression. Specifically, the estimating equation from the first "pseudo" derivative of $L(\boldsymbol{\beta})$ is

$$l(\boldsymbol{\beta}) = -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda\{2\mathbb{1}(\boldsymbol{\beta} > 0) - \mathbf{1}_q\},$$

where $\mathbb{1}$ is the indicator function, that is, $\mathbb{1}(a > b) = 1$ if $a > b$, and 0 otherwise. Then we can use $l(\boldsymbol{\beta})$ to get the estimate $\hat{\boldsymbol{\beta}}$ and characterize the distribution $\boldsymbol{\Sigma}^{-1/2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \sim f(\boldsymbol{z})$, where $\boldsymbol{\Sigma} = (\sigma_{ij})_{q \times q} = \text{cov}(\hat{\boldsymbol{\beta}})$ is the covariance matrix of $\hat{\boldsymbol{\beta}}$. Here, we define $f(\boldsymbol{z}) = \prod_{j=1}^{q} f_j(z_j)$, where each z_j is a standard normal random variable and $f_j(\cdot)$ denotes its probability density function. The Lasso method can result in some estimated coefficients being exactly zero. To model this sparsity, we approximate the marginal density $f_j(z)$ using a two-component mixture $f_j(z) \approx c_j\phi(z) + (1 - c_j)\phi_{\tilde{\epsilon}}(z)$, where $\phi(z)$ is the standard normal density and $\phi_{\tilde{\epsilon}}(z)$ is the density of a normal distribution with zero mean and small variance (e.g., $\tilde{\epsilon} = 10^{-6}$).

Table 2. The design points for different experiments.

$d^{(1)}$					$d^{(2)}$					BLHD				
S_1	S_2	A	B	C	S_1	S_2	A	B	C	S_1	S_2	A	B	C
0	0	0	0	0	0	0	0	0	0	1	1	0	0	1
0	0	1	1	0	0	0	0	1	1	0	0	0	1	1
0	0	1	0	1	0	0	1	0	1	0	0	0	0	1
0	0	0	1	1	0	0	1	1	0	1	1	1	1	0
1	0	0	0	0	1	0	0	0	1	0	1	1	0	0
1	0	1	1	0	1	0	0	1	0	0	1	1	1	1
1	0	1	0	1	1	0	1	0	0	1	0	1	0	0
1	0	0	1	1	1	0	1	1	1	1	0	1	1	1
0	1	0	0	0	0	1	0	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	1	0	1	0	0	1	0
0	1	1	0	1	0	1	1	0	0	1	1	0	1	1
0	1	0	1	1	0	1	1	1	1	1	0	0	0	0
1	1	0	0	0	1	1	0	0	0	0	1	0	1	0
1	1	1	1	0	1	1	0	1	1	0	0	1	1	0
1	1	1	0	1	1	1	1	0	1	0	0	1	0	1
1	1	0	1	1	1	1	1	1	0	1	1	1	0	1

The key of the induced smoothing is adding a scaled perturbation of parameters to form the new estimating equation as

$$\begin{aligned} \tilde{l}(\boldsymbol{\beta}) &= \mathbb{E}_{\boldsymbol{z}}[l(\boldsymbol{\beta} + \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{z})] \\ &= \int l(\boldsymbol{\beta} + \boldsymbol{\Sigma}^{\frac{1}{2}}\boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} \\ &= -\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\eta}(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{c}), \end{aligned}$$

where the q-dimensional penalty vector $\boldsymbol{\eta}(\boldsymbol{\beta}, \boldsymbol{z}; \boldsymbol{c})$ has the elements $\eta_j = c_j\{2\Phi(\beta_j/\sqrt{\sigma_{jj}}) - 1\} + (1 - c_j)\{2\Phi_{\tilde{\epsilon}}(\beta_j/\sqrt{\sigma_{jj}}) - 1\}$. Here Φ and $\Phi_{\tilde{\epsilon}}$ are the corresponding cumulative density function of standard normal distribution. Since the $\tilde{l}(\boldsymbol{\beta})$ is a smooth function, we can use the sandwich formula to compute the estimation covariance as

$$\hat{\boldsymbol{\Sigma}} = \tilde{l}'(\hat{\boldsymbol{\beta}})^{-1} \boldsymbol{V} \tilde{l}'(\hat{\boldsymbol{\beta}})^{-1}.$$

Here $\boldsymbol{V} = \text{cov}(\tilde{l}(\boldsymbol{\beta})) \propto \boldsymbol{X}^T\boldsymbol{X}$. Consequently, we can obtain the approximate distribution of $\hat{\boldsymbol{\beta}}$ and perform the statistical hypothesis testing on $\boldsymbol{\beta}$. For example of the hypothesis testing $H_0 : \beta_j = 0$, the Wald statistic under H_0 is

$$W_j = \frac{\hat{\beta}_j}{\sqrt{\text{var}(\hat{\beta}_j)}} \xrightarrow{d} N(0, 1).$$

In the following simulation and application studies, we will use the Wald statistic to test the significance of the coefficients for models. For implementation, we adopt the *islasso* R package for the induced smoothing approach to estimate the effect size and determine statistical significance (Cilluffo et al. 2020). Insignificant effects will be shrunk to zero.

5. Simulation

This section presents the results of several simulation studies that demonstrate the accuracy and robustness of the MLSD model. We extensively tested our model under two-layer and three-layer sliced designs under different noise levels.

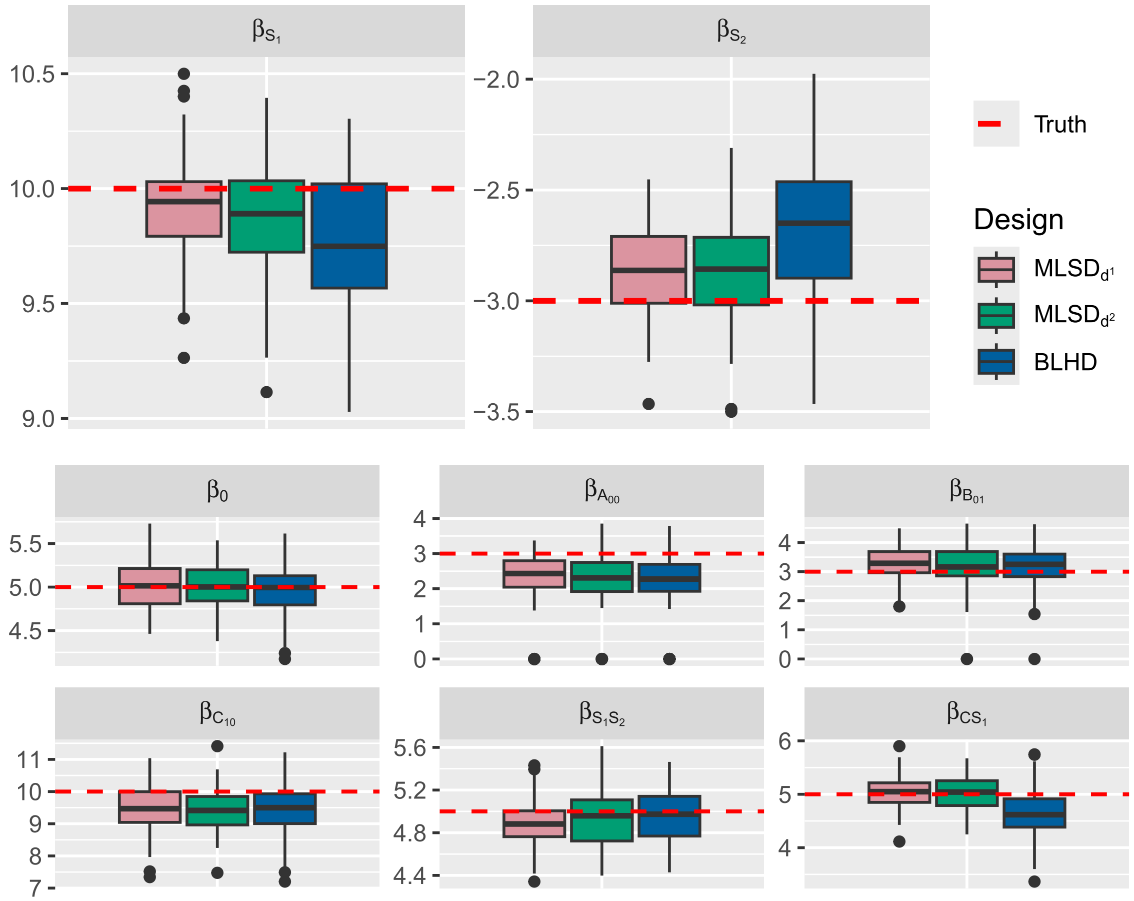

Figure 2. Comparison of simulation results across 100 replications: proposed designs ($d^{(1)}$ and $d^{(2)}$) versus BLHD for accurate and precise estimation of key coefficients.

5.1. Two-Layer Sliced Design

We first consider a two-layer sliced design 2^{2+3-1}. Assume that the slice factor S_1 and S_2 are of the same importance. Based on the sliced minimum aberration, we can obtain two best design schemes $d^{(1)}$: $I = ABC$ and $d^{(2)}$: $I = ABCS_1S_2$. The ordered sliced word-length pattern grouping for $d^{(1)}$ and $d^{(2)}$ are $SW_{d^{(1)}} = \{(3^0, 4^2, 5^0)\}$ and $SW_{d^{(2)}} = \{(3^0, 4^2, 5^0)\}$. It means that these two designs both have the ordered sliced resolution of IV. Based on the factorial strategies, the above two designs are obtained with 16 observations. (Hung, Joseph, and Melkote 2009) introduces the concepts of branching and nested factors to describe situations where certain factors only exist within the levels of other factors. These are referred to as nested factors, with the factor containing other factors being called a branching factor. In MLSD, the effects of design factors vary across the levels of the slice factors, exhibiting a dynamic relationship similar to that of nested variables. The design factors in MLSD allow their effects to vary across all levels of the slice factors. Therefore, one can consider the slice factors as branching factors and the design factors as nested factors. The unique aspect here is that all levels of the nested factor can exist under the branching factor. An optimal branching Latin hypercube design (BLHD) is generated by maximizing the minimum inter-site distance. This optimal BLHD is used as the benchmark for comparison. The design points for the three designs are presented in Table 2. Here, 0 and 1 represent two levels of factors.

Given a design matrix, we consider the underlying model for the response as follows.

$$
\begin{aligned}
y = \beta_0 &+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{A_{ij}}(A|S_1 = i, S_2 = j) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{B_{ij}}(B|S_1 = i, S_2 = j) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1} \beta_{C_{ij}}(C|S_1 = i, S_2 = j) \\
&+ \sum_{i=1}^{2} \beta_{S_i} S_i + \beta_{S_1S_2} S_1 S_2 + \sum_{i=1}^{2} \beta_{AS_i} AS_i \\
&+ \sum_{i=1}^{2} \beta_{BS_i} BS_i + \sum_{i=1}^{2} \beta_{CS_i} CS_i + \epsilon, \qquad (2)
\end{aligned}
$$

where ϵ is the error term following $N(0, \sigma^2)$. Here, take $\beta_0 = 5, \beta_{A_{00}} = 3, \beta_{B_{01}} = 4, \beta_{C_{10}} = 10, \beta_{S_1} = 10, \beta_{S_2} = -3, \beta_{CS_1} = 5, \beta_{S_1S_2} = 5$, the other coefficients as 0. Two scenarios of the error terms are considered: $\epsilon \sim N(0, 1)$ and $\epsilon \sim N(0, 0.5)$. When the responses are generated, the proposed analysis method in Section 4 is used to analyze the data.

Figure 2 presents the results of analyses conducted across different experiments with 100 replications. The performance of $d^{(1)}$ and $d^{(2)}$ surpasses that of BLHD in terms of accuracy and precision for the coefficients of S_1 and S_2. The means of the estimated coefficients for the proposed designs are closer to the true values, and the corresponding boxplots are noticeably narrower. For other coefficients, the three designs exhibit comparable performance, with the exception of β_{CS_1}. It is observed that the competitor's estimates for β_{CS_1} deviate more from the truth compared to those from the proposed MLSD method. Overall, significant coefficients can be accurately estimated, as demonstrated by the coverage of the true values within the boxplots. It is interesting to note that the estimation capabilities of significant slice factor effects are more accurate than design factors, as indicated by their small variance in supplementary Table S1. In addition, the proposed method also works better when the noise level is small. One can also obtain more accurate estimations for slice factors under low noise levels. In supplementary Table S1, the insignificant coefficients can be accurately evaluated under small noise conditions. However, under large noise conditions, there is a risk of incorrectly identifying insignificant factors. For example, the true value of $\beta_{A_{01}}$ is 0, but the mean estimated by the BLHD method is small, albeit nonzero. This issue is also present in the other two methods. Nonetheless, although the estimated values are not exactly zero, they are very small. Moreover, we calculate the signal-to-noise ratios (SNR) using formula $\mathrm{var}(\boldsymbol{X}\hat{\boldsymbol{\beta}})/\mathrm{var}(\boldsymbol{Y})$ to access the meaningful information captured in the data. In both levels of noise, the SNR is approximately 95%, indicating 95% of the variability in the data can be attributed to the true signal, while the remaining 5% is due to noise. It suggests that the results of the analysis are likely to be reliable with findings supported by the data.

5.2. Three-Layer Sliced Design

This section examines the performance of the proposed methods under the three-layer situations. Consider a three-layer sliced design 2^{3+3-1} with two design schemes $d^{(1)}$: $I = ABC$ and $d^{(2)}$: $I = ABCS_1S_2S_3$. The ordered sliced word-length pattern grouping for $d^{(1)}$ and $d^{(2)}$ are $SW_{d^{(1)}} = \{(3^0, 4^3, 5^0)\}$ and $SW_{d^{(2)}} = \{(3^0, 4^0, 5^3)\}$, respectively. Note that $d^{(1)}$ is ordered sliced resolution IV and $d^{(2)}$ is ordered sliced resolution V. Thus, according to the sliced minimum aberration criterion, the design $d^{(1)}$ is better than the design $d^{(2)}$. For comparison, we also take an optimal BLHD generated by maximizing the minimum inter-site distance as a benchmark.

Based on the constructed design, we consider the following model to generate response y as

$$
\begin{aligned}
y = \beta_0 &+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} \beta_{A_{ijk}}(A|S_1 = i, S_2 = j, S_3 = k) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} \beta_{B_{ijk}}(B|S_1 = i, S_2 = j, S_3 = k) \\
&+ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} \beta_{C_{ijk}}(C|S_1 = i, S_2 = j, S_3 = k) \\
&+ \sum_{i=1}^{3} \beta_{S_i} S_i + \sum_{i=1}^{3}\sum_{\substack{j=1 \\ i\neq j}}^{3} \beta_{S_iS_j} S_iS_j \\
&+ \sum_{i=1}^{3} \beta_{AS_i} AS_i + \sum_{i=1}^{3} \beta_{BS_i} BS_i + \sum_{i=1}^{3} \beta_{CS_i} CS_i + \epsilon,
\end{aligned}
$$

where ϵ is the error term following $N(0, \sigma^2)$. Here, we take $\beta_0 = 5$, $\beta_{A_{000}} = 3$, $\beta_{A_{011}} = 2$, $\beta_{B_{010}} = 4$, $\beta_{B_{001}} = 2$, $\beta_{C_{100}} = 10$, $\beta_{C_{101}} = 3$, $\beta_{S_1} = 10$, $\beta_{S_2} = -3$, $\beta_{S_3} = 5$, $\beta_{CS_1} = 5$, $\beta_{S_1S_2} = 5$ and the other coefficients are 0's. We also consider two scenarios of the noise levels as $\epsilon \sim N(0, 1)$ and $\epsilon \sim N(0, 0.5)$.

Figure 3 displays the simulation outcomes based on 100 replications. This figure demonstrates that the two MLSDs accurately estimate the true parameters, as evidenced by the coverage of the true values by all boxplots. Regarding the BLHD, some simulations suggest that $\beta_{A_{011}}$ and $\beta_{B_{001}}$ are statistically insignificant. Although BLHD effectively estimates the slice factors with reasonable accuracy and precision, it exhibits lessrecision and accuracy compared to MLSDs. This is indicated by the means of deviating more from the red line and broader boxplots. This discrepancy is attributed to the BLHD's design, which is more suited for factors with continuous values. When applied to a discrete design, the optimization algorithms face challenges in achieving a global optimum due to the presence of ties, leading to inefficiencies in the algorithm thus, there is no guarantee that the BLHD identified is the optimal one. Supplementary Table S2 indicates the use of $d^{(2)}$ provides slightly better estimates for slice factors than the use of $d^{(1)}$. That is the estimates of β_{S1}, β_{S2}, β_{S3} of $d^{(2)}$ move the mean slightly closer to true values of parameters than that of $d^{(1)}$. Overall, we can find the performance of BLHD is less competitive to MLSD when the layers of design are growing. For $d^{(2)}$, we calculate that the SNR is around 98%, which is much better than that of two two-layer slice designs. Here, we conduct more designs to explore the variability of responses.

6. Case Study

In this section, we use the proposed MLSD to improve AI assurance (Batarseh, Freeman, and Huang 2021; Batarseh and Freeman 2022; Batarseh, Chandrasekaran, and Freeman 2023). In AI assurance, it is important to understand the hyperparameter effects in different models and optimization strategies when training a deep learning model. The combinatorial complexity of hyper-parameters is a common challenge for machine learning practitioners and researchers since it often requires significant computational resources to find a good combination of hyper-parameters for a specific task. Several studies have focused on hyperparameter tuning to enhance the performance of deep neural networks (Pannakkong et al. 2022; Liao et al. 2022). Moreover, a good combination of hyper-parameters can be different for deep learning methods under different models and optimization strategies.

For investigating the effects of hyper-parameters in the deep learning model, practitioners typically rely on their own experience to determine the values of each parameter. They might change one element at a time while keeping the others constant, similar to a one-factor-at-a-time analysis (Wu and

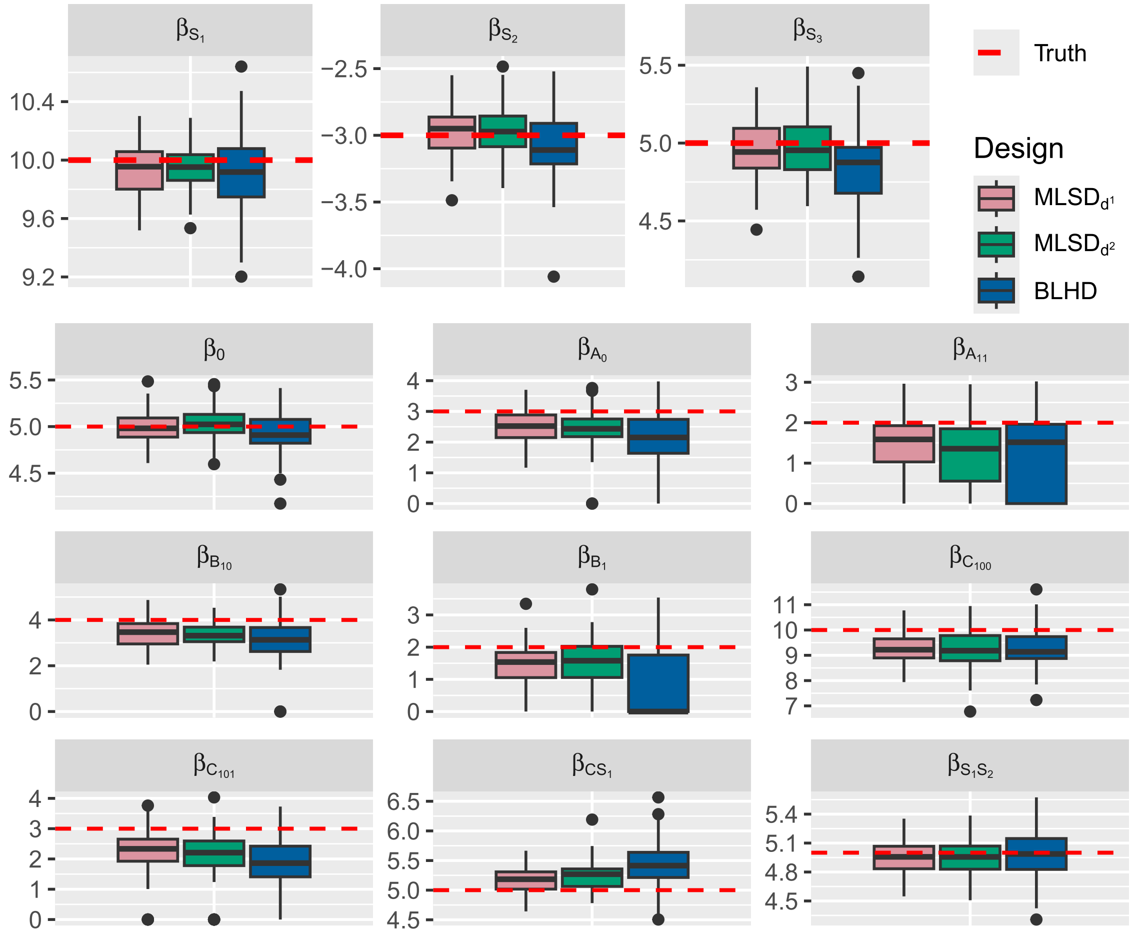

Figure 3. Simulation results for three-layer platform designs: comparative performance of MLSDs and BLHD across 100 replications.

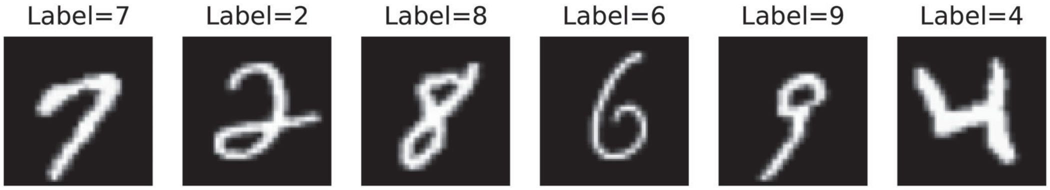

Figure 4. Examples of MNIST Dataset.

Hamada 2021). Based on their results and intuition, they then make a decision about the optimal combination of settings. Such a procedure cannot identify the best combinations of hyper-parameters. Alternatively, practitioners could implement all possible combinations to find the best one, but such an exhaustive search requires costly computational resources. Using the proposed MLSD and analysis method, one can provide a statistical tool to facilitate a more efficient exploration of the hyper-parameter, model, and optimization space for AI assurance.

Specifically, we consider the classification deep learning algorithm for a popular dataset named MNIST (Deng 2012), which is composed of handwritten digits formatted as 28 × 28 pixel monochrome images. Figure 4 illustrates some image examples. The objective of the deep learning algorithm is to accurately classify the images and achieve good prediction accuracy. There are several key factors, including model architecture, optimization strategies, batch sizes, and epoch and learning rates, for consideration in the deep learning algorithm. For model architecture, two popular models used in image classification problems are the Convolutional Neural Network (CNN) and Multi-layer Perceptron (MLP). In general, CNN is the preferred choice for image classification tasks over MLP due to its ability to capture spatial hierarchies and local patterns in images, leading to better performance. However, MLP typically requires fewer computational resources compared to CNN. If computational resources are limited, an MLP might be a more feasible choice. In addition, MLP is generally simpler and faster to train than CNN, especially for small datasets or simple image classification tasks where the spatial hierarchies are not as critical. Therefore, when the performance of MLP is comparable to that of CNN without significant sacrifices in accuracy, MLP becomes an attractive option. Optimizers are the tools we use to estimate the parameters by minimizing a loss function. Some well-known optimizers are adaptive gradients, such as AdamW (Loshchilov and Hutter 2017) and Adagrad (Lydia and Francis 2019). The learning rate can control how much model weights should be updated. As deep learning algorithms often require significant computational resources, it may not be feasible to optimize a model using all available data at once. As a result, the data are split into smaller subsets or batches, and the model estimation is iterated over each batch. Thus, the number of data points in each batch is known as the batch size, and the number of times the algorithm runs on the entire training dataset is known as the number of epochs.

To investigate the effects of the above key factors in the deep learning algorithm, we cast this parameter selection problem as a two-layer sliced design. The model and optimizer can be considered as the slice factors, which are crucial components in deep learning. The remaining factors, including the number of epochs, batch size, and learning rate, can be considered design factors. While some design factors are continuous, users of deep learning algorithms typically choose discrete levels, for example, 32, 64, or 128 for the batch size. By using the proposed $2^{2+(3-1)}$ two-layer sliced design, we can effectively investigate the effects of these factors on the performance of the deep learning algorithm. Table 3 lists the factors and their respective levels in this study.

Table 3. Five factors and their levels.

	Slice factors		Design factors		
	Model (S_1)	Optimizer (S_2)	Epoch (A)	Batch size (B)	Learning rate (C)
Level 0	MLP	AdamW	20	32	10^{-3}
Level 1	CNN	Adagrad	50	64	10^{-4}

In this two-layer sliced design, we consider two optimal designs with 16 runs as described in Table 2 of Section 5.1. For each design, we take the prediction accuracy as our response. The corresponding results for $d^{(1)}$, $d^{(2)}$, and the BLHD are reported in Table 4.

In the Figure 5, we present the main effects for slice factors S_1 and S_2, as well as the design factor C within a subdesign. For slice factor S_1, the main effects on D_1 and D_2 are similar and both positive. However, in BLHD, S_1 exhibits slightly negative main effects. Factors S_2 and C_{11} exhibit similar effects across all three designs. Figure 6 illustrates interactions across the three designs, all of which display similar patterns. According to these two figures, we observed BLHD shows a different effect for S_1.

After collecting all the responses (y), we conduct parameter estimations and summarize the significant parameters for all strategies in Table 5. All other parameters not reported in

Table 4. The classification accuracy (%) for each design strategy in comparison.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D_1	95.36	94.81	93.15	91.23	98.54	98.34	98.44	96.72	88.05	89.42	78.28	69.56	94.25	95.92	77.97	62.45
D_2	94.96	90.42	92.99	94.76	74.6	88.51	89.52	74.24	97.23	98.24	98.79	97.98	94.25	66.78	78.58	95.31
BLHD	66.83	90.73	91.68	95.46	89.42	75.45	98.69	97.78	88.86	98.14	63.36	98.44	88.51	94.96	92.94	78.53

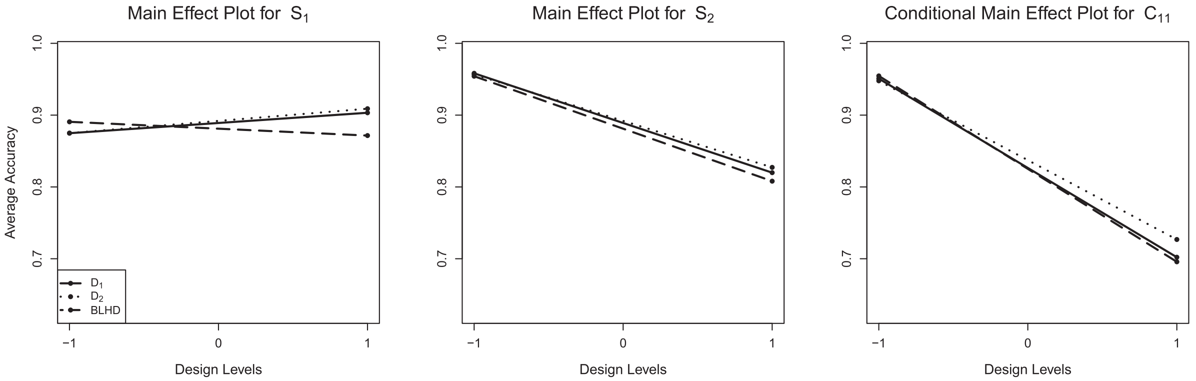

Figure 5. Main effects for S_1, S_2 and C_{11} across different designs.

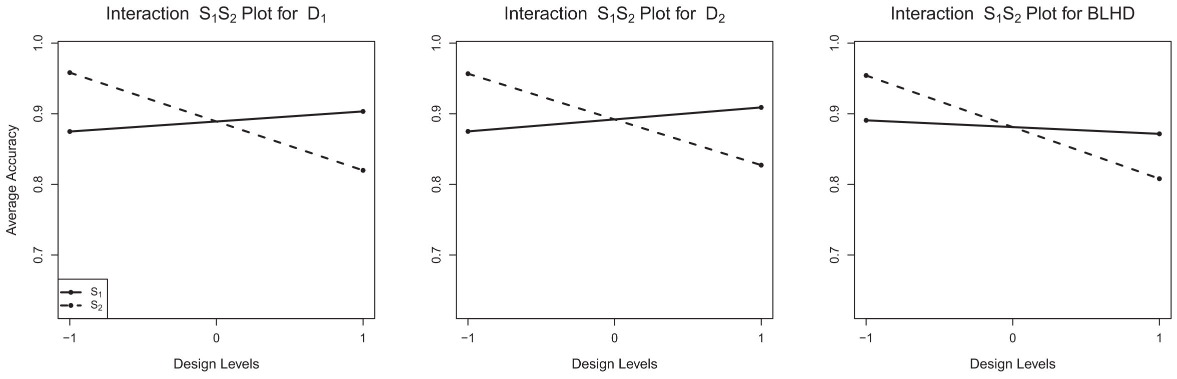

Figure 6. Interaction effect between S_1 and S_2 across different designs.

Table 5. Parameters estimates in two-layer sliced design.

	β_0	$\beta_{A_{01}}$	$\beta_{A_{11}}$	$\beta_{B_{01}}$	$\beta_{B_{11}}$	$\beta_{C_{00}}$	$\beta_{C_{01}}$	$\beta_{C_{11}}$	β_{S_1}	β_{S_2}	$\beta_{S_1S_2}$
D_1	88.90	2.48	4.26	−1.80	−3.43	−1.41	−7.37	−12.40	1.41	−6.90	−0.75
D_2	89.19	0	2.87	0	−2.27	−1.24	−6.95	−10.7	1.61	−6.39	−0.61
BLHD	88.59	0	6.02	0	−1.09	0	−6.19	−9.38	0.80	−6.94	−1.66

the table are zero across all three designs. Table 5 compares results for different designs. The coefficients estimation of I, β_{S_1}, β_{S_2}, $\beta_{S_1S_2}$, $\beta_{C_{00}}$, and $\beta_{C_{01}}$ in $d^{(1)}$ and $d^{(2)}$ are very close. The minor differences observed between the estimates from $d^{(1)}$ and $d^{(2)}$ can be attributed to random noise. Specifically, two primary sources contribute to this noise. The first is run-to-run variability inherent in training CNNs, influenced by factors such as random initialization of weights, the ordering of mini-batches during stochastic gradient descent, and adaptive learning rates. The second source of noise arises from the selection of tuning parameter λ, introducing additional variability into the model estimation process and affecting the accuracy of the coefficient estimates. As a result, there is a slight estimation gap between $d^{(1)}$ and $d^{(2)}$. Among all the coefficients, the signs of each detected effect in the two optimal strategies are consistent. However, for the BLHD, the magnitude of S_1 and S_1S_2 are different from the coefficients of $d^{(1)}$ and $d^{(2)}$. Moreover, the BLHD design fails to identify the effects of $\beta_{C_{00}}$, indicating a potential limitation in its detection capability.

Based on the result for $d^{(1)}$, the average accuracy of the deep learning algorithm is 88.90%. The estimate of β_{S_1} is 1.41, which means that changing the model from MLP to CNN can improve the accuracy by 1.41%. This is consistent with the general observation that CNN works better in computer vision tasks than MLP. In terms of β_{S_2}, we can infer that the optimizer AdamW performs better than the Adagrad and it can enhance the accuracy by 6.9%. The estimate of $\beta_{S_1S_2}$ shows that the model and optimizer interact with each other and the interaction harms the prediction accuracy. The estimate of $\beta_{A_{00}}$ presents improving epochs from 20 to 50 can increase 2.48% accuracy when we use the MLP model and AdamW optimizer.

The estimate of $\beta_{A_{11}}$ implies that improving epochs from 20 to 50 increases 4.26% accuracy when we use the CNN model and Adagrad optimizer. Overall, the main effect of epoch (i.e., design factor A) in every slice factor plays a positive impact on accuracy. This is because the model does not converge at 20 epochs and it needs more epochs to improve the model's performance. The effects of batch size (i.e., design factor B) always hurt the prediction. This can be explained by using a larger batch size means fewer model updates in each model epoch. In addition, the significant effect of the learning rate (i.e., design factor C) shows that prediction accuracy can benefit from the bigger learning rate since it can accelerate the model learning process. The results of $d^{(2)}$ can be interpreted similarly.

From the analysis, one can obtain several insightful observations, guiding the selection of factors for model performance. Notably, the first slice factor indicates a positive sign, suggesting that CNN models outperform MLP models in the classification of handwritten digital images, enhancing accuracy by approximately 1.5%. If computational resources are limited and some decrease in accuracy is acceptable, MLP models may be considered a viable alternative. Moreover, the choice of optimization strategy plays a crucial role in performance, with the AdamW optimizer boosting accuracy by about 6.5%, as demonstrated by the second slice factor. Based on the insights from the slice factors, coupled with considerations such as budget and computation time, researchers can make informed decisions regarding the most suitable model and optimization method. Furthermore, under specific combinations of slice factors, experimenters can identify optimal hyperparameter settings. In a short summary, the proposed approach is useful to facilitate simultaneous model selection, optimization strategies choosing, and hyperparameter tuning within a single fractional experimental design, enabling efficient and effective exploration of model configurations.

7. Discussion

We proposed a multi-layer sliced design to quantify the effects of slice factors and design factors to account for design factors with different effects under different level combinations of slice factors. We also developed a criterion for finding the minimum aberration design in this new situation. Moreover, we developed an effective analysis method to estimate the effects of these factors and test their significance. It enhances the reduction of estimation bias through the combination of sub-model estimations. The application of the proposed design to AI assurance is particularly important in practice as it can effectively detect the effects of hyper-parameters affecting the performance of AI algorithms.

The proposed method can also be adapted for online experiments and other AI applications. In online experiments, slice factors can be different mediums where an experiment is conducted, and they can significantly influence the results of the study. For example, these factors can include device types such as laptops and cellphones, web browsers such as Google Chrome and Safari, and e-commerce platforms like Amazon and eBay. The importance of a slice factor can vary depending on the context and objectives of the study. Here is a scenario where one slice factor is more important than another. Consider conducting a study to understand online shopping behavior, with a focus on comparing user interactions and purchase decisions on laptops versus cellphones. In this experiment, it involves two slice factors: device types (cellphone and laptop) and web browsers (Google Chrome and Firefox). The device type can be more important than the web browser for several reasons. First, users might use cellphones for quick purchases or while on the go whereas desktop shopping might be associated with more extensive research and comparison. Understanding these differences is crucial for the study's objectives. Second, cellphones might offer features like push notifications and personalized recommendations that can increase user engagement and conversion rates. Analyzing the differences in shopping behavior between laptops and cellphones can provide valuable insights for e-commerce businesses. Notably, one should be careful when making assumptions about the importance sequence of slice factors. Inappropriate assumptions might introduce estimation bias and impact the estimation efficiency. In practice, to mitigate such risks, one should conduct a pilot study to assess the relative importance of factors when possible. It is also good to check with practitioners to set reasonable assumptions based on their experience. In the future, it would be interesting to take into account slice factors of varying importance and apply the proposed MLSD to online shopping experiments.

The proposed estimation method can estimate the effects of interest simultaneously. An interesting finding is that in the case of only one slice factor, the conditional value X_1 given S_1 can be viewed from the angle of conditional main effects (Mak and Wu 2019). Note that there is a close connection between interaction effects and conditional main effects. It is important to remark that we only consider the conditional main effects on the slice factors, which differs from the conventional conditional main effect model. In addition, it is important to clarify the differences between the slice factor and the blocking factor in experimental design. While the design strategy for constructing these two types of designs may have some similarities, their underlying mechanisms are distinct. In the case of the blocking factor, it is essential to control for the variation it introduces. While for the slice factor, it is crucial to accurately detect its effect, which is often the primary focus. Additionally, when constructing the design, priority should be given to accurate estimation of the slice factor's effect.

We would like to remark that exploring the potential of experimental design in modern applications is a promising direction and is gaining increasing attention. Some recent works have shown that experimental design can help improve the performance of AI. Lim et al. (2020) employed experimental design to enhance the efficiency of AI-driven optimization for complex disease treatments using a minimal number of experiments. Lian et al. (2021, 2022) explored the design of experiments to improve the robustness and assurance of AI algorithms. On the other hand, other researchers study using AI to improve the experimental design. Kleinegesse and Gutmann (2020, 2021) and Guo et al. (2022) used contrastive variational mutual information estimators to better find the optimal design. Ren et al. (2021) considered the smart device to collect the most informative data by optimizing the knowledge graph of the customer. Our study presents useful results demonstrating that the application of statistical experimental design can enhance AI research. In the future, one can consider conducting experiments with the Amazon Mechanical Turk platform to gather data for a multi-layer sliced design and analysis. Additionally, we identify several directions for future exploration. First, it would be valuable to examine sliced design under nonnormal response scenarios, extending to both design construction and analysis methodologies. Second, researchers could consider linking the slice aberration criterion to estimation capability with statistical foundation and using alternative design criteria that incorporate prior information about design factors to construct experimental designs, such as the I-WLP criterion (Li, Mee, and Zhou 2019). Third, it is interesting to explore the minimum aberration design under the constructed estimation model (Mukerjee, Wu, and Chang 2017; Chang 2023).

Supplementary Materials

The *mlsd_supplementary.pdf* contains proofs of Theorem 1 and Proposition 1, and additional simulation results. The *code.zip* provides all the code for reproducing the simulations (Section 5) and case study (Section 6).

Acknowledgments

The authors thank the Editor, the Associate Editor, and anonymous referees for their insightful comments and constructive suggestions that greatly improved the quality and clarity of this work.

Disclosure Statement

The authors report no conflict of interest.

ORCID

Xinwei Deng http://orcid.org/0000-0002-1560-2405

References

Bardenet, R., Brendel, M., Kégl, B., and Sebag, M. (2013), "Collaborative Hyperparameter Tuning," in *International Conference on Machine Learning*, pp. 199–207, PMLR. [1]

Batarseh, F. A., and Freeman, L. (2022), *AI Assurance: Towards Trustworthy, Explainable, Safe, and Ethical AI*, Amsterdam: Elsevier. [1,8]

Batarseh, F. A., Freeman, L., and Huang, C.-H. (2021), "A Survey on Artificial Intelligence Assurance," *Journal of Big Data*, 8, 60. [1,8]

Batarseh, F. A., Chandrasekaran, J., and Freeman, L. J. (2023), "An Introduction to AI Assurance," in *AI Assurance*, eds. F. A. Batarseh and L. J. Freeman, pp. 3–12, Amsterdam: Elsevier. [1,8]

Bergstra, J., and Bengio, Y. (2012), "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, 13, 281–305. [1,2]

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011), "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems* (Vol. 24). [2]

Box, G. E. P., and Hunter, J. S. (1961), "The 2 k-p Fractional Factorial Designs," *Technometrics*, 3, 311–351. [2,4]

Box, C. E. P., Hunter, W. H., and Hunter, S. (1978), *Statistics for Experimenters* (Vol. 664), New York: Wiley. [1]

Brown, B. M., and Wang, Y.-G. (2005), "Standard Errors and Covariance Matrices for Smoothed Rank Estimators," *Biometrika*, 92, 149–158. [6]

Chang, M.-C. (2022), "A Unified Framework for Minimum Aberration," *Statistica Sinica*, 32, 251–69. [2]

——— (2023), "Bayesian-Inspired Minimum Contamination Designs under a Double-Pair Conditional Effect Model," *Statistical Theory and Related Fields*, 7, 336–349. [5,12]

Cheng, C.-S., Steinberg, D. M., and Sun, D. X. (1999), "Minimum Aberration and Model Robustness for Two-Level Fractional Factorial Designs," *Journal of the Royal Statistical Society*, Series B, 61, 85–93. [2]

Cheng, C.-S. (2016), *Theory of Factorial Design*, London: Chapman and Hall/CRC. [2,3]

Cilluffo, G., Sottile, G., La Grutta, S., and Muggeo, V. M. R. (2020), "The Induced Smoothed Lasso: A Practical Framework for Hypothesis Testing in High Dimensional Regression," *Statistical Methods in Medical Research*, 29, 765–777. [2,6]

Deng, L. (2012), "The Mnist Database of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine*, 29, 141–142. [9]

Dougherty, S., Simpson, J. R., Hill, R. R., Pignatiello, J. J., and White, E. D. (2015), "Effect of Heredity and Sparsity on Second-Order Screening Design Performance," *Quality and Reliability Engineering International*, 31, 355–368. [1]

Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K., et al. (2013), "Towards an Empirical Foundation for Asessing Bayesian Optimization of Hyperparameters," in *NIPS Workshop on Bayesian Optimization in Theory and Practice* (Vol. 10). [2]

Feurer, M., Springenberg, J., and Hutter, F. (2015, "Initializing Bayesian Hyperparameter Optimization via Meta-Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 29), AAAI. [2]

Fisher, R. A. (1970), "Statistical Methods for Research Workers," in *Breakthroughs in Statistics: Methodology and Distribution*, eds. S. Kotz and N. L. Johnson, pp. 66–70, New York: Springer. [2]

Fries, A., and Hunter, W. G. (1980), "Minimum Aberration 2 k–p Designs," *Technometrics*, 22, 601–608. [2,4]

Gramacy, R. B., Sauer, A., and Wycoff, N. (2022), "Triangulation Candidates for Bayesian Optimization," in *Advances in Neural Information Processing Systems* (Vol. 35), pp. 35933–35945. [2]

Gunst, R. F., and Mason, R. L. (2009), "Fractional Factorial Design," *Wiley Interdisciplinary Reviews: Computational Statistics*, 1, 234–244. [2]

Guo, Q., Chen, J., Wang, D., Yang, Y., Deng, X., Huang, J., Carin, L., Li, F., and Tao, C. (2022), "Tight Mutual Information Estimation with Contrastive Fenchel-Legendre Optimization," in *Advances in Neural Information Processing Systems* (Vol. 35), pp. 28319–28334. [12]

Hung, Y., Joseph, V. R., and Melkote, S. N. (2009), "Design and Analysis of Computer Experiments with Branching and Nested Factors," *Technometrics*, 51, 354–365. [2,7]

Hutter, F., Kotthoff, L., and Vanschoren, J. (2019), *Automated Machine Learning: Methods, Systems, Challenges*, Cham: Springer Nature. [2]

Jones, B., and Nachtsheim, C. J. (2009), "Split-Plot Designs: What, Why, and How," *Journal of Quality Technology*, 41, 340–361. [2]

Joy, T. R., Rana, S., Gupta, S., and Venkatesh, S. (2016), "Hyperparameter Tuning for Big Data Using Bayesian Optimisation," in *2016 23rd International Conference on Pattern Recognition*, pp. 2574–2579, IEEE. [2]

Kittitharayada, P., Buddhakulsomsiri, J., Pannakkong, W., et al. (2021), "Using Design of Experiments during the Process of Tuning Hyperparameters in Machine Learning Algorithms," PhD thesis, Thammasat University. [2]

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. (2017), "Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets," in *Artificial Intelligence and Statistics*, pp. 528–536, PMLR. [2]

Kleinegesse, S., and Gutmann, M. U. (2020), "Bayesian Experimental Design for Implicit Models by Mutual Information Neural Estimation," in *International Conference on Machine Learning*, pp. 5316–5326, PMLR. [12]

——— (2021), "Gradient-based Bayesian Experimental Design for Implicit Models Using Mutual Information Lower Bounds," arXiv preprint arXiv:2105.04379. [12]

Lee, W.-Y., Park, S.-M., and Sim, K.-B. (2018), "Optimal Hyperparameter Tuning of Convolutional Neural Networks based on the Parameter-Setting-Free Harmony Search Algorithm," *Optik*, 172, 359–367. [2]

Lenth, R. V. (1989), "Quick and Easy Analysis of Unreplicated Factorials," *Technometrics*, 31, 469–473. [6]

Lessmann, S., Stahlbock, R., and Crone, S. F. (2005), "Optimizing Hyperparameters of Support Vector Machines by Genetic Algorithms," in *IC-AI* (Vol. 74), pp. 82, ACM. [2]

Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., and Soatto, S. (2020a), "Rethinking the Hyperparameters for Fine-Tuning," arXiv preprint arXiv:2002.11770. [1]

Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B., and Talwalkar, A. (2020b), "A System for Massively Parallel Hyperparameter Tuning," *Proceedings of Machine Learning and Systems*, 2, 230–246. [1]

Li, W., Mee, R. W., and Zhou, Q. (2019), "Using Individual Factor Information in Fractional Factorial Designs," *Technometrics*, 61, 38–49. [12]

Lian, J., Choi, K., Veeramani, B., Hu, A., Freeman, L., Bowen, E., and Deng, X. (2022), "Do-Aiq: A Design-of-Experiment Approach to Quality Evaluation of AI Mislabel Detection Algorithm," arXiv preprint arXiv:2208.09953. [12]

Lian, J., Freeman, L., Hong, Y., and Deng, X. (2021), "Robustness with Respect to Class Imbalance in Artificial Intelligence Classification Algorithms," *Journal of Quality Technology*, 53, 505–525. [12]

Liao, L., Li, H., Shang, W., and Ma, L. (2022), "An Empirical Study of the Impact of Hyperparameter Tuning and Model Optimization on the Performance Properties of Deep Neural Networks," *ACM Transactions on Software Engineering and Methodology*, 31, 1–40. [8]

Lim, J. J., Goh, J., Rashid, M. B. M. E., and Chow, E. K.-H. (2020), "Maximizing Efficiency of Artificial Intelligence-Driven Drug Combination Optimization through Minimal Resolution Experimental Design," *Advanced Therapeutics*, 3, 1900122. [12]

Lorenzo, P. R., Nalepa, J., Kawulok, M., Ramos, L. S., and Pastor, J. R. (2017), "Particle Swarm Optimization for Hyper-Parameter Selection in Deep Neural Networks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 481–488, ACM. [2]

Loshchilov, I., and Hutter, F. (2017), "Decoupled Weight Decay Regularization," arXiv preprint arXiv:1711.05101. [9]

Lydia, A., and Francis, S. (2019), "Adagrad-an Optimizer for Stochastic Gradient Descent," *International Journal of Information and Computing Science*, 6, 566–568. [9]

Mak, S., and Wu, C. F. J. (2019), "cmenet: A New Method for Bi-Level Variable Selection of Conditional Main Effects," *Journal of the American Statistical Association*, 114, 844–856. [12]

Mantovani, R. G., Horváth, T., Cerri, R., Vanschoren, J., and de Carvalho, A. C. P. L. F. (2016), "Hyper-Parameter Tuning of a Decision Tree Induction Algorithm," in *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 37–42, IEEE. [2]

Rahul Mukerjee, Wu, C. F. J., and Chang, M.-C. (2017), "Two-Level Minimum Aberration Designs under a Conditional Model with a Pair of Conditional and Conditioning Factors," *Statistica Sinica*, 27, 997–1016. [5,12]

Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P., and Buddhakulsomsiri, J. (2022), "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Sudy of ann, svm, and dbn," *Mathematical Problems in Engineering*, 2022, 8513719. [8]

Phadke, M. S. (1995), *Quality Engineering Using Robust Design*, Hoboken, NJ: Prentice Hall PTR. [2]

Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019), "Hyperparameters and Tuning Strategies for Random Forest," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9, e1301. [2]

Ren, X., Yin, H., Chen, T., Wang, H., Huang, Z., and Zheng, K. (2021), "Learning to Ask Appropriate Questions in Conversational Recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 808–817, ACM. [12]

Sadeghi, S., Chien, P., and Arora, N. (2020), "Sliced Designs for Multi-Platform Online Experiments," *Technometrics*, 62, 387–402. [1,2,4]

Seeger, M., Steinke, F., and Tsuda, K. (2007), "Bayesian Inference and Optimal Design in the Sparse Linear Model," in *Artificial Intelligence and Statistics*, pp. 444–451, PMLR. [1]

Shao, J. (1997), "An Asymptotic Theory for Linear Model Selection," *Statistica Sinica*, 7, 221–242. [6]

Snoek, J., Larochelle, H., and Adams, R. P. (2012), "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems* (Vol. 25). [1]

Taguchi, G. (1987), *System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs*, Bingham Farms, MI: American Suppliers Institute. [2]

Tang, B., and Wu, C. F. J. (1996), "Characterization of Minimum Aberration 2n-k Designs in Terms of their Complementary Designs," *The Annals of Statistics*, 24, 2549–2559. [2]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Series B, 58, 267–288. [6]

Wu, C. F. J., and Hamada, M. S. (2021), *Experiments: Planning, Analysis, and Optimization*, New York: Wiley. [1,2,3,5,6,9]

Yogatama, D., and Mann, G. (2014), "Efficient Transfer Learning Method for Automatic Hyperparameter Tuning," in *Artificial Intelligence and Statistics*, pp. 1077–1085, PMLR. [2]

Yuan, M., Roshan Joseph, V., and Lin, Y. (2007), "An Efficient Variable Selection Approach for Analyzing Designed Experiments," *Technometrics*, 49, 430–439. [1]