Balanced and Robust Multi-Treatment Experimental Designs via Randomized Differencing

Qing Chen
Rutgers University

Jing Jia
Rutgers University

Peng Zhang
Rutgers University

Abstract

We introduce GKK+, a new design for multi-arm randomized controlled trials. Standard Bernoulli randomization is robust but often yields poor covariate balance, while existing restricted-randomness designs mainly address two-arm settings. GKK+ extends the Karmarkar–Karp (KK) differencing method to multiple arms. When covariates are smooth and well-behaved, GKK+ achieves an exponentially better covariate balance than the standard Bernoulli design while preserving sufficient randomness. GKK+ improves efficiency in estimating treatment effects and supports standard asymptotic inference. Simulations on synthetic and real datasets demonstrate improved balance and lower estimator variance compared to existing methods.

1 INTRODUCTION

Randomized Controlled Trials (RCTs) are the gold standard for estimating the causal effects of new treatments (Imbens and Rubin, 2015). The *design* of an RCT refers to the probability distribution specifying the random assignment of experimental units into treatment arms.

A good design of an RCT should satisfy two key properties. First, it should *balance covariates* – pretreatment characteristics of the units – so that causal estimates are more precise when covariates are predictive of outcomes. Second, it should remain *robust* against model misspecification when covariates are not predictive. Robustness here means that the treatment effect estimator remains reasonably precise even in worst-case outcome settings. Under the standard Bernoulli design (independent and uniform random assignment), covariates are balanced in expectation, but chance imbalances can be severe when the experiment size is small or moderate. Bernoulli is, however, highly robust in the sense that it minimizes worst-case estimation error (Harshaw et al., 2024).

To improve balance, researchers have proposed designs with restricted randomness, which sacrifice some robustness in exchange for better covariate balance. Examples include randomized blocking (Fisher, 1935), pairwise matching (Greevy et al., 2004), rerandomization (Morgan and Rubin, 2012), and more advanced designs (Krieger et al., 2019; Arbour et al., 2021, 2022; Harshaw et al., 2024; Rao and Zhang, 2025).

Most of these designs focus on the two-arm setting. Extending them to $m > 2$ arms poses significant challenges. Achieving covariate balance requires solving an m-way partitioning problem, which is strongly NP-hard (Garey and Johnson, 1975). Straightforward extensions, such as rerandomization or exhaustive search, can be computationally costly and lack theoretical guarantees as the number of arms increases. However, multi-arm experiments are common in practice: they allow simultaneous evaluation of several interventions, improving efficiency and reducing cost (Dunnett, 1955; Jaki and Wason, 2018; Howard et al., 2018). For example, in clinical trials, a four-arm study may compare three different drugs against one control.

1.1 Our Contributions

We develop a new design for multi-arm RCTs. It achieves strong covariate balance without sacrificing robustness. Technically, our design extends the classic differencing method of Karmarkar and Karp (KK) and its multivariate generalization (GKK) (Karmarkar and Karp, 1982; Turner et al., 2020), and we call it *GKK+*. More concretely, our *contributions* are as follows:

Design. GKK+ extends the recursive grouping framework of KK/GKK to the multi-arm, multi-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

dimensional setting. Methodologically, GKK+ introduces a recursion parameter T that unifies existing designs and captures the balance-robustness tradeoff: $T = 0$ recovers Bernoulli, $T = 1$ yields m-wise matching[1], and $T = \Theta(\log n)$ achieves exponentially smaller imbalance. Computationally, we develop a linear-time implementation of GKK+ for fixed m and covariate dimension.

Theoretical guarantees. Under mild conditions, we prove GKK+ achieves exponentially better coordinate-wise balance than Bernoulli and Lipschitz discrepancy comparable to m-wise matching, and retains robustness close to Bernoulli. Detailed definitions, assumptions, and formal results are given in Section 2 and Theorem 3.2.

Inference. Under GKK+, we study the Horvitz–Thompson estimator for the average treatment effect. We show that the estimator is unbiased and enjoys substantially smaller variance than the Bernoulli design under mild assumptions. We establish asymptotic normality of the estimation error, which supports Wald confidence intervals.

1.2 Related Works

Experimental designs trade off covariate balance and robustness. At one extreme, Bernoulli and complete randomization provide strong robustness but may yield large covariate imbalances in small or moderate samples (Kallus, 2018; Azriel et al., 2022; Harshaw et al., 2024). At the other extreme, optimal assignments minimize imbalance but may sacrifice randomness and robustness (Student, 1938; Bertsimas et al., 2015; Kasy, 2016; Deaton and Cartwright, 2018; Bhat et al., 2020). Many approaches lie in between, including randomized blocking and matching (Greevy et al., 2004; Imai et al., 2009; Bai et al., 2022) and rerandomization (Morgan and Rubin, 2012; Li et al., 2018; Li and Ding, 2020). While generalizations to multiple arms exist, theoretical guarantees and scalable implementations are often largely restricted to two arms.

A more recent line of work connects RCT design with combinatorial discrepancy theory (Matousek, 1999; Chazelle, 2001; Chen et al., 2014), yielding designs with better covariate balance (Harshaw et al., 2024; Turner et al., 2020; Arbour et al., 2022; Rao and Zhang, 2025). Our work extends this connection to RCT designs for multi-arm settings, with improved balance and robustness.

Roadmap. We outline the problem setting in Section 2 and state our main results in Section 3. We then present the GKK+ design in Section 4 and numerical experiments in Section 5. We defer all proofs to the appendix due to space limitations.

[1]That is, find a minimum-weight m-wise matching of the units, and then randomize within each m-group.

2 PROBLEM SETUP

Notations. We use **bold** letters for vectors and matrices and regular letters for scalars. For a positive integer n, let $[n] = \{1, \dots, n\}$.

We study RCTs with n units, m treatment arms, and p-dimensional covariates. Our goal is to design a random assignment that (1) balances covariates across arms and (2) preserves enough randomness to maintain robustness to model misspecification and support valid inference. We follow the Neyman-Rubin potential outcomes framework (Rubin, 2005).

Potential outcome framework. For each unit $i \in [n]$ and arm $k \in [m]$, let Y_{ik} be the potential outcome of unit i under treatment arm k, and let $\boldsymbol{Y}_i = (Y_{i1}, \dots, Y_{im}) \in \mathbb{R}^m$. Only the outcome under the realized assignment is observed. Let $\boldsymbol{x}_i \in \mathbb{R}^p$ be unit i's covariates, and let $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n \times p}$. Throughout the paper, we assume that the units $(\boldsymbol{x}_1, \boldsymbol{Y}_1), \dots, (\boldsymbol{x}_n, \boldsymbol{Y}_n)$ are sampled i.i.d. from the population of interest.

Let $\boldsymbol{W} = (W_1, \dots, W_n) \in [m]^n$ be the random assignment vector, where W_i is the treatment arm assigned to unit i. We assume the following:

Assumption 2.1 (SUTVA (Imbens and Rubin, 2015))**.** *Each unit's potential outcomes are unaffected by the treatment assignments of other units, and, for each unit and each treatment level, there are no alternative versions that could yield different potential outcomes.*

Assumption 2.2 (Unconfoundedness given covariates)**.** *The treatment assignment is independent of the potential outcomes conditional on the covariates, i.e.,* $(\boldsymbol{Y}_1, \dots, \boldsymbol{Y}_n) \perp\!\!\!\perp \boldsymbol{W} \mid (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$.

Assumption 2.3 (Symmetric designs)**.** *The design randomizes treatment labels such that, for every* $i \in [n]$ *and* $k \in [m]$*, the marginal assignment probability satisfies* $\Pr(W_i = k) = 1/m$.

Assumption 2.3 can be satisfied, for example, by partitioning the n units into m groups and then uniformly permuting the treatment arm labels across them.

We want to estimate the average treatment effect (ATE), including both the sample ATE $\tau_{kk'}$ and the population ATE $\tau^*_{kk'}$ for any two treatment arms $k, k' \in [m]$, defined as

$$\tau_{kk'} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} (Y_{ik} - Y_{ik'}), \quad \tau^*_{kk'} \stackrel{\text{def}}{=} \mathbb{E}[Y_{1k} - Y_{1k'}].$$

We use the Horvitz-Thompson (HT) estimator:

$$\hat{\tau}_{kk'} \stackrel{\text{def}}{=} \frac{m}{n}\left(\sum_{i:W_i=k} Y_{ik} - \sum_{i:W_i=k'} Y_{ik'}\right).$$

Lemma 2.4 (Unbiasedness)**.** *The HT estimator under a symmetric design is unbiased: for any $k, k' \in [m]$, $\mathbb{E}[\hat{\tau}_{kk'} \mid \boldsymbol{X}] = \tau_{kk'}$ and $\mathbb{E}[\hat{\tau}_{kk'}] = \tau^*_{kk'}$.*

Our goal is (1) to construct a random $\boldsymbol{W}$ to minimize the worst-case variance of $\hat{\tau}_{kk'}$, which equals mean-squared error since $\hat{\tau}_{kk'}$ is unbiased, and (2) to establish valid confidence intervals for $\tau_{kk'}$ and $\tau^*_{kk'}$.

Covariate balance and robustness. Following Kallus (2018), for each $k \in [m]$ and $i \in [n]$, define

$$f_k(\boldsymbol{x}) \stackrel{\text{def}}{=} \mathbb{E}[Y_{ik} \mid \boldsymbol{x}_i = \boldsymbol{x}], \quad \epsilon_{ik} \stackrel{\text{def}}{=} Y_{ik} - f_k(\boldsymbol{x}_i).$$

Thus, $\mathbb{E}[\epsilon_{ik}|\boldsymbol{x}_i] = 0$. For any function f, let

$$B_{kk'}(\boldsymbol{W}, f) \stackrel{\text{def}}{=} \frac{m}{n}\sum_{i:W_i=k} f(\boldsymbol{x}_i) - \frac{m}{n}\sum_{i:W_i=k'} f(\boldsymbol{x}_i),$$

which is the difference of sums under f between arms k and k' normalized by m/n. Assume f_k belongs to a function class $\mathcal{H}$. Kallus (2018) shows that minimizing the worst-case variance of $\hat{\tau}_{kk'}$ is closely related to finding $\boldsymbol{W}$ that minimizes $\max_{f\in\mathcal{H}} \mathbb{E}[B^2_{kk'}(\boldsymbol{W}, f)]$.

Kallus (2018) derives optimal balanced designs under various modeling assumptions on $\mathcal{H}$. However, when the assumed model is misspecified, the resulting design may be arbitrarily suboptimal (Krieger et al., 2019). This motivates us to develop a design that ensures covariate balance while preserving sufficient randomness to maintain robustness, similar to randomized blocking, pairwise matching, rerandomization, and advanced designs (Krieger et al., 2019; Harshaw et al., 2024).

2.1 Covariate Balance Metrics

We consider two function classes for $\mathcal{H}$: linear functions $\mathcal{H}_{\text{lin}} = \{\boldsymbol{x} \mapsto \boldsymbol{\beta}^\top \boldsymbol{x}\}$ and Lipschitz functions $\mathcal{H}_{\text{Lip}}$, and define corresponding covariate balance metrics. Both function classes and their associated metrics have been extensively studied in the design of RCTs (Kallus, 2018; Krieger et al., 2019; Harshaw et al., 2024). We will later show that GKK+ achieves balance with respect to both metrics simultaneously.

Linear functions and ℓ_∞ discrepancy. Let

$$\boldsymbol{x}^{(k)} \stackrel{\text{def}}{=} \sum_{i:W_i=k} \boldsymbol{x}_i$$

be the covariate sum in arm k. Define the ℓ_∞ *discrepancy*:

$$\mathcal{D}_\infty(\boldsymbol{W}) \quad \stackrel{\text{def}}{=} \quad \max_{k,k'\in[m]} \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k')}\right\|_\infty. \quad (1)$$

Here, $\|\cdot\|_\infty$ denotes the ℓ_∞ norm, i.e., the largest absolute entry of a vector. Thus, $\mathcal{D}_\infty(\boldsymbol{W})$ measures the maximum coordinate imbalance across arms.

This metric bounds the worst-case imbalance for linear functions of covariates with bounded coefficients. Specifically, consider $f(\boldsymbol{x}_i) = \boldsymbol{\beta}^\top \boldsymbol{x}_i$ for some fixed but unknown $\boldsymbol{\beta}$. By Hölder's inequality,

$$\begin{aligned}\frac{n}{m}\left|B_{kk'}(\boldsymbol{W}, f)\right| &= \left|\boldsymbol{\beta}^\top\Big(\sum_{i:W_i=k} \boldsymbol{x}_i - \sum_{i:W_i=k'} \boldsymbol{x}_i\Big)\right| \\ &\le \|\boldsymbol{\beta}\|_1 \left\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k')}\right\|_\infty \le \|\boldsymbol{\beta}\|_1 \mathcal{D}_\infty(\boldsymbol{W}).\end{aligned}$$

Therefore, a small value of $\mathcal{D}_\infty$ ensures a small worst-case imbalance for any linear function.

The ℓ_∞ discrepancy can be generalized to an ℓ_q discrepancy for any $q \in [1, \infty)$, defined by $\mathcal{D}_q(\boldsymbol{W}) \stackrel{\text{def}}{=} \max_{k,k'\in[m]} \|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k')}\|_q$. By norm equivalence, $\mathcal{D}_q(\boldsymbol{W}) \le p^{1/q}\mathcal{D}_\infty(\boldsymbol{W})$.

Lipschitz functions and Lipschitz discrepancy. We next consider $\mathcal{H} = \mathcal{H}_{\text{Lip}}$, the class of Lipschitz functions with bounded Lipschitz constant. The class $\mathcal{H}_{\text{Lip}}$ captures smooth functions that need not be linear. A function $f : \mathbb{R}^p \to \mathbb{R}$ is c_f-Lipschitz if $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \le c_f \|\boldsymbol{x} - \boldsymbol{x}'\|_\infty$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^p$, where c_f is the Lipschitz constant of f.

We define the *Lipschitz discrepancy* as

$$\mathcal{D}_{\text{Lip}}(\boldsymbol{W}) \stackrel{\text{def}}{=} \max_{f\in\mathcal{H}_{\text{Lip}}} \max_{k,k'\in[m]} \Big|\sum_{i:W_i=k} f(\boldsymbol{x}_i) - \sum_{i:W_i=k'} f(\boldsymbol{x}_i)\Big|.$$

This metric measures the worst-case imbalance over Lipschitz functions. For any $f \in \mathcal{H}_{\text{Lip}}$,

$$|B_{kk'}(\boldsymbol{W}, f)| \le \frac{m}{n}\mathcal{D}_{\text{Lip}}(\boldsymbol{W}).$$

Thus, a small $\mathcal{D}_{\text{Lip}}$ guarantees balance for all Lipschitz functions.

For $m = 2$, pairwise matching attains the optimal $\mathcal{D}_{\text{Lip}}$ (Kallus, 2018; Bai, 2022), but may suffer from large $\mathcal{D}_\infty$. Our goal is to develop a design that has a substantially small $\mathcal{D}_\infty$, thereby reducing estimation error under linear outcomes, while achieving a $\mathcal{D}_{\text{Lip}}$ comparable to pairwise matching and its multi-arm generalizations.

2.2 Robustness metric

We introduce a *new* robustness metric that quantifies how much a design deviates from the Bernoulli design through almost t-wise independence (formal definitions follow). The Bernoulli design assigns units *independently*, offering strong robustness to model misspecification (Kallus, 2018; Harshaw et al., 2024), and guaranteeing asymptotic normality of the HT estimator and the validity of Wald-type confidence intervals (CIs). Davezies et al. (2024) further shows that asymptotic *exact* t-wise independence suffices for these properties. However, many commonly used designs, such as pairwise matching, fail to satisfy exact t-wise independence. Thus, we propose a new metric based on *almost* t-wise independence, under which we also establish asymptotic normality.

Our robustness metric $r_t(\boldsymbol{W})$ measures the fraction of t-subsets of units whose assignments do *not* behave independently. Formally, for an integer $t \geq 2$, let

$$r_t(\boldsymbol{W}) \stackrel{\text{def}}{=} \frac{1}{\binom{n}{t}} \sum_{1 \leq i_1 < \ldots < i_t \leq n} \mathbb{1}(W_{i_1}, \ldots, W_{i_t}),$$

where $\mathbb{1}(W_{i_1}, \ldots, W_{i_t}) = 1$ if there exists $w_1, \ldots, w_t \in [m]^t$ such that

$$\Pr(W_{i_1} = w_1, \ldots, W_{i_t} = w_t) \neq \prod_{j=1}^{t} \Pr(W_{i_j} = w_j)$$

and $\mathbb{1}(W_{i_1}, \ldots, W_{i_t}) = 0$ otherwise. Under Bernoulli, $r_t(\boldsymbol{W}_{\text{Bernoulli}}) = 0$ for all t, since all units are assigned independently. When $r_t(\boldsymbol{W})$ is small for all fixed t, $\boldsymbol{W}$ is nearly indistinguishable from Bernoulli, ensuring the standard limit theorems and valid Wald-type CIs.

3 OUR RESULTS

We introduce a new design, GKK+, which achieves both strong covariate balance and robustness in RCTs with multiple arms and multi-dimensional covariates. In our theorems, $O(\cdot), \Theta(\cdot)$, and $o(\cdot)$ hide constants only depending on m.

To achieve exponentially improved $\mathcal{D}_\infty$, we need the following mild assumption on covariate distribution.

Assumption 3.1 (Covariate assumption)**.** *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be i.i.d. random vectors with density ρ. We assume ρ is supported on $[-\Delta, \Delta]^p$, where $\Delta = o(e^{\sqrt{\log n}})$, and ρ is Lipschitz continuous and uniformly bounded above by a constant.*

Assumption 3.1 is mild. The bounded support condition is weak, since the bound is allowed to grow as slowly as $e^{\sqrt{\log n}}$. This ensures that no single unit largely affects the balance metrics. In practice, covariates are usually measured or normalized within bounded ranges. As noted in Turner et al. (2020), this assumption can be relaxed to light-tailed distributions; for example, if $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are drawn from the standard Gaussian, then with high probability they lie in $[-10\sqrt{\log n}, 10\sqrt{\log n}]^p$, so we can truncate the Gaussian to satisfy the bounded support condition. The Lipschitz continuity and uniform upper bound conditions rule out densities that change too abruptly or concentrate too heavily, and they are satisfied by many common distributions such as Gaussian.

3.1 GKK+ Design

Theorem 3.2 (Balance-robustness of GKK+)**.** *Fix $m \geq 2$. Suppose*

$$p \leq \frac{1}{m}\sqrt{\frac{\log n}{5 \log m}}, \quad T = \left\lfloor \frac{0.1 \log n}{mp \log m} \right\rfloor.$$

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be i.i.d. p-dimensional random vectors satisfying Assumption 3.1. Then, in $O(pn)$ time, GKK+ outputs a random assignment $\boldsymbol{W} \in [m]^n$ such that, with probability at least $1 - 1/n$,

1. *Balance: $\mathcal{D}_\infty(\boldsymbol{W}) = n^{-\Theta\left(\frac{\log n}{(mp)^2 \log m}\right)}$ and $\mathcal{D}_{\text{Lip}}(\boldsymbol{W}) = O\left(n^{1 - \frac{1}{5mp}}\right)$.*
2. *Robustness: $r_t(\boldsymbol{W}) = o(1)$ for all fixed $t \geq 2$.*

GKK+ groups similar units[2], compresses each group using a random permutation of arm labels, and then recurses on the compressed groups for T recursions.

The exponential bound for $\mathcal{D}_\infty$ follows because each GKK+ recursion decreases the coordinate-wise sum differences by a factor of about $n^{-\Omega(1/mp)}$, and after T recursions this yields the stated bound. The bound for $\mathcal{D}_{\text{Lip}}$ comes from the first GKK+ recursion, an analog of m-wise matching. For runtime, each GKK+ recursion runs in time proportional to the current problem size and it reduces that size by a factor of m, so the total cost across all recursions is linear.

Balance metric $\mathcal{D}_\infty$. Our bound on $\mathcal{D}_\infty$ matches the classical result of KK up to the constant in the exponent when $p = 1$. For $m = 2$, our bound is slightly weaker than the GKK bound. However, GKK requires knowing the density of $\boldsymbol{x}_i$, an assumption that may not hold in practice. The classical KK bound on $\mathcal{D}_\infty$ is essentially the best achievable by efficient algorithms under standard hardness assumptions (Vafa and Vaikuntanathan, 2025; Mallarapu and Sellke, 2025).

[2]We use *arm* for a treatment arm, and *group* for algorithmic grouping.

The Bernoulli design yields $\mathcal{D}_\infty = \Theta(\sqrt{n})$. A related method, Greedy Pair-Switching (Krieger et al., 2019), attains $\mathcal{D}_\infty = O(n^{-(1+2/p)})$ for $m = 2$ while preserving sufficient randomness. Our result improves these guarantees by an exponential factor.

Balance metric $\mathcal{D}_{\text{Lip}}$. For $m = 2$, Kallus (2018) shows that pairwise matching achieves the optimal $\mathcal{D}_{\text{Lip}}$, given by the total weight of a minimum-weight perfect matching with edge weights defined by ℓ_∞ distances between vectors $\boldsymbol{x}_i$. For $p \geq 3$, the optimal value satisfies $\mathcal{D}_{\text{Lip}} = C_p \cdot n^{1-1/p}$ as $n \to \infty$, where C_d depends only on p (Ledoux, 2022). Our bound on $\mathcal{D}_{\text{Lip}}$ is close to this optimum.

Covariate dimension. Theorem 3.2 assumes that p does not increase too rapidly with n. For high-dimensional cases, we propose a heuristic based on the recursive grouping and compression strategy of GKK+ and show empirically in Section 5 that it outperforms standard designs.

3.2 Statistical Properties

We bound the variance of $\hat{\tau}_{kk'}$, establish convergence, and an asymptotic normality of the estimation error.

Theorem 3.3 (Variance)**.** *Under Assumptions 2.1, 2.2, and 2.3, for any arms $k, k' \in [m]$, the variance*

$$var(\hat{\tau}_{kk'}) = \mathbb{E}[var(D_{kk'} \mid \boldsymbol{X})] + V_n,$$

where

$$D_{kk'} = \frac{1}{m}\sum_{l \neq k} B_{kl}(\boldsymbol{W}, f_k) - \frac{1}{m}\sum_{l \neq k'} B_{k'l}(\boldsymbol{W}, f_{k'}),$$

$$V_n = \frac{1}{n}\mathbb{E}\left[(m-1)(\epsilon_{ik}^2 + \epsilon_{ik'}^2) + 2\epsilon_{ik}\epsilon_{ik'}\right] + var(\tau_{kk'}).$$

Under the GKK+ design and Assumption 3.1, if all $f_k \in \mathcal{H}_{\text{lin}}$ with bounded coefficients, then $var(\hat{\tau}_{kk'}) - V_n = O(n^{-\Theta(\frac{\log n}{(mp)^2 \log m})})$; if all $f_k \in \mathcal{H}_{\text{Lip}}$, then $var(\hat{\tau}_{kk'}) - V_n = O(n^{-\frac{2}{5mp}})$.

The term V_n is design-independent. Theorem 3.3 shows that if covariates and outcomes are uncorrelated, i.e., $\mathbb{E}[Y_{ik} \mid \boldsymbol{x}_i] = \mathbb{E}[Y_{ik}]$, then covariate balancing does not affect variance: the HT estimator has the same variance under Bernoulli and any balancing design (e.g., pairwise matching or GKK+), paralleling Theorem 7 of Kallus (2018) for equal arm size.

Theorem 3.4 (Convergence)**.** *Under Assumptions 2.1, 2.2 and 3.1, with $\mathbb{E}[\epsilon_{ik}^2 \mid \boldsymbol{x}_i] < \infty$ and $f_k \in \mathcal{H}_{\text{Lip}}$ for all k, the GKK+ design guarantees that for any $k, k' \in [m]$, $\hat{\tau}_{kk'} - \tau_{kk'} \to 0$ in probability and $\hat{\tau}_{kk'} - \tau_{kk'} = \mathcal{O}_p(n^{-1/2})$. The same results hold when $\tau_{kk'}$ is replaced by $\tau^*_{kk'}$.*

Theorem 3.5 (Asymptotic normality)**.** *Under Assumptions 2.1, 2.2 and 3.1, with $\mathbb{E}[\epsilon_{ik}^2 \mid \boldsymbol{x}_i] < \infty$ and $f_k \in \mathcal{H}_{\text{Lip}}$ for all k, the GKK+ design guarantees that*

$$\sqrt{n}(\hat{\tau}_{kk'} - \tau_{kk'}) \xrightarrow{d} \mathcal{N}(0, V),$$
$$\sqrt{n}(\hat{\tau}_{kk'} - \tau^*_{kk'}) \xrightarrow{d} \mathcal{N}(0, V^*),$$

where $V = \mathbb{E}\left[(m-1)(\epsilon_{ik}^2 + \epsilon_{ik'}^2) + 2\epsilon_{ik}\epsilon_{ik'}\right]$ and $V^* = var(\hat{\tau}_{kk'})$.

Under standard conditions, GKK+ is asymptotically normal with the same variance as Bernoulli, so higher-moment differences vanish asymptotically.

4 GKK+ DESIGN

In this section, we describe GKK+. Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ and m arms, GKK+ outputs a random assignment $\boldsymbol{W} \in [m]^n$ such that the m-arm sums $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}$ are close in ℓ_∞ norm, which directly bounds the covariate imbalance measure $\mathcal{D}_\infty$. Later, we show that GKK+ also indirectly guarantees small $\mathcal{D}_{\text{Lip}}(\boldsymbol{W})$ and small $r_t(\boldsymbol{W})$ for all fixed t.

GKK+ groups similar units, compresses each group by randomly permuting arm labels, and then recurses on the compressed groups for T steps. We illustrate this grouping-and-compression process with an example.

Example. Consider assigning 8 units with one-dimensional covariates valued in $1, 2, \ldots, 8$ to two arms, A and B. GKK+ sorts the covariates and pairs neighbors: $(1,2), (3,4), (5,6), (7,8)$. In each pair, one unit will be assigned to arm A and the other to B, with the specific assignment decided later. Next, these small pairs are combined into bigger pairs: $(1,2)$ with $(3,4)$ and $(5,6)$ with $(7,8)$. Within each bigger pair, the assignments are coordinated to cancel cross-arm imbalance. For example, if 1 goes to A and 2 to B, then 3 must go to B and 4 to A, resulting both arms with the same total of 5 for the first four numbers. A similar rule applies to the second bigger pair (5–8), which ensures balance there too. Repeating this step once more yields two global assignments $(A, B, B, A, B, A, A, B), (B, A, A, B, A, B, B, A)$, and in both cases each arm sums up to 18. GKK+ outputs each of two assignments with probability 1/2.

The experimenter can choose the recursion depth T. After T recursions, GKK+ chooses uniformly at random among all assignments that satisfy the constraints imposed by those T steps. T controls the balance-randomness trade-off: smaller T retains more randomness, larger T improves balance. For $T = 0$, GKK+ reduces to Bernoulli; for $T = 1$, GKK+ reduces to m-wise matching; for $T = \Theta(\log n)$, it achieves exponentially small $\mathcal{D}_\infty$. In practice, we suggest using

$T = 0$ for maximum randomness, $T = 1$ for small experiments or simple matching, and larger T for larger n to improve balance while preserving near-Bernoulli independence.

The toy example highlights the main idea of GKK+. We now present the formal algorithm for general $m \geq 2$ and arbitrary covariates.

Definitions. Given m vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \in \mathbb{R}^p$, define their m-tuple as $\tilde{\boldsymbol{v}} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m) \in \mathbb{R}^{mp}$ formed by stacking the vectors vertically. Let $\tilde{\boldsymbol{v}}^{(i)} \in \mathbb{R}^p$ denote the i-th component of $\tilde{\boldsymbol{v}}$, which is the vector $\boldsymbol{x}_i \in \mathbb{R}^p$. For a permutation σ on $[m]$, define $(\sigma \circ \tilde{\boldsymbol{v}})^{(k)} = \tilde{\boldsymbol{v}}^{(\sigma(k))}$ for all $k \in [m]$. Thus, $\sigma \circ \tilde{\boldsymbol{v}}$ permutes the components of $\tilde{\boldsymbol{v}}$. For example, given $m = 2$ and $x_1 = 3, x_2 = 4$ (with $p = 1$), their m-tuple is $\tilde{\boldsymbol{v}} = (3, 4)$ with components $\tilde{\boldsymbol{v}}^{(1)} = 3, \tilde{\boldsymbol{v}}^{(2)} = 4$; for $\sigma = (2, 1)$, we have $\sigma \circ \tilde{\boldsymbol{v}} = (4, 3)$.

We use m-tuples to represent assignments of units to arms: the kth block corresponds to arm k. In the above example, $\tilde{\boldsymbol{v}} = (3, 4)$ means unit 1 is assigned to arm 1 and unit 2 to arm 2, while $\sigma \circ \tilde{\boldsymbol{v}} = (4, 3)$ swaps their assignments.

We lift each covariate vector $\boldsymbol{x}_i$ to the m-tuple $\tilde{\boldsymbol{v}}_i = (\boldsymbol{x}_i, \boldsymbol{0}, \ldots, \boldsymbol{0})$, where only the first block is nonzero. A permutation π_i on $[m]$ moves $\boldsymbol{x}_i$ into the block corresponding to the chosen arm $\pi_i(1)$. So, choosing a permutation π_i simply determines which treatment arm unit i is assigned to. In $\sum_{i=1}^{n} \pi_i \circ \tilde{\boldsymbol{v}}_i$, each of the m components equals the sum of the covariate vectors assigned to that arm. Our goal is to choose permutations π_i's so that these m components are close in the ℓ_∞ norm, thereby controlling the imbalance metric $\mathcal{D}_\infty$.

Cyclic differencing. This operator combines m distinct m-tuples in a way that reduces cross-arm imbalance. Let σ be the cyclic shift on $[m]$, defined by $\sigma(1) = m$ and $\sigma(i) = i - 1$ for $i \geq 2$. For m-tuples $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_m \in \mathbb{R}^{mp}$, define

$$\begin{aligned}&\textsc{CycDiff}(\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_m)\\ &= \tilde{\boldsymbol{v}}_1 \;+\; \sigma \circ \tilde{\boldsymbol{v}}_2 \;+\; \sigma^2 \circ \tilde{\boldsymbol{v}}_3 \;+\; \cdots \;+\; \sigma^{m-1} \circ \tilde{\boldsymbol{v}}_m.\end{aligned}$$

That is, each tuple is first rotated by a cyclic shift and then all rotated tuples are summed. When the inputs are similar, the cyclic rotations cause their imbalances across arms to cancel out. For example, $\textsc{CycDiff}((1, 2), (3, 4)) = (1, 2) + (4, 3) = (5, 5)$.

4.1 GKK+ Algorithm

Following KK and GKK (Karmarkar and Karp, 1982; Turner et al., 2020), we refer to a recursion step aforementioned as a *phase*. GKK+ runs for T phases.

At phase t, the input consists of a scalar $\alpha_t > 0$, a set $\mathcal{S}_t \subset [-\alpha_t, \alpha_t]^{mp}$ of m-tuples, and a residual tuple $\tilde{\boldsymbol{b}}_t \in \mathbb{R}^{mp}$. Initially, $\alpha_1 = \alpha$, $\mathcal{S}_1 = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_n\}$, where each $\tilde{\boldsymbol{v}}_i = (\boldsymbol{x}_i, \boldsymbol{0}, \ldots, \boldsymbol{0})$ is a lifted tuple, and $\tilde{\boldsymbol{b}}_1 = \boldsymbol{0}$.

Each phase of GKK+ takes the current collection of tuples, groups together nearby ones, uses cyclic differencing to cancel imbalances, and then reduces any leftovers. Repeating this recursively shrinks imbalance while retaining randomness.

Partition($\alpha_t, \mathcal{S}_t$): partition $[-\alpha_t, \alpha_t]^{mp}$ into smaller subcubes for grouping nearby tuples. Let $l = \lceil |\mathcal{S}_t|^{1/(4mp)} \rceil$ and $\alpha_t' = \alpha_t / l$. Partition the cube $[-\alpha_t, \alpha_t]^{mp}$ into axis-aligned subcubes of side length α_t', collected into $\mathcal{C}_t$. Any two tuples in the same subcube differ by at most $2\alpha_t'$ in ℓ_∞ norm.

Figure 1 illustrates one phase of GKK+ for the case $m = 2$ and $p = 1$. Blue points are 2-tuples. The square is partitioned into small subsquares.

Difference($\mathcal{S}_t, \mathcal{C}_t$): apply cyclic differencing to tuples in each subcube to cancel imbalances. For each subcube containing at least m tuples, repeatedly sample m of them without replacement, say $\tilde{\boldsymbol{v}}_{i_1}, \ldots, \tilde{\boldsymbol{v}}_{i_m}$. Center each tuple by subtracting the subcube center $\boldsymbol{c}$, and form

$$\tilde{\boldsymbol{v}} = \textsc{CycDiff}(\tilde{\boldsymbol{v}}_{i_1} - \boldsymbol{c}, \ldots, \tilde{\boldsymbol{v}}_{i_m} - \boldsymbol{c}) \;\in\; [-C\alpha_t', C\alpha_t']^{mp}$$

for a universal constant C. Collect all such outputs into $\mathcal{S}_t'$. Any leftovers (fewer than m per subcube) form the set $\mathcal{L}_t$.

In Figure 1, points in each subsquare are grouped and cyclically differenced into purple points. The ungrouped point in a subsquare is a leftover.

Clean-up($\mathcal{S}_t', \mathcal{L}_t, \tilde{\boldsymbol{b}}_t, \alpha_t'$): reduce leftovers. *Merge step:* permute each leftover in $\mathcal{L}_t$ and combine with the residual $\tilde{\boldsymbol{b}}_t$ into a single tuple $\tilde{\boldsymbol{v}}^\star$ satisfying $\|\tilde{\boldsymbol{v}}^\star\|_\infty = O(\alpha_t)$. *Shrink step:* repeatedly draw tuples from $\mathcal{S}_t'$ at random without replacement and replace $\tilde{\boldsymbol{v}}^\star$ by a differencing of it with the drawn tuple, stopping once $\|\tilde{\boldsymbol{v}}^\star\|_\infty \leq \gamma\alpha_t'$, where $\gamma = 2m^3p$. Set

$$\mathcal{S}_{t+1} \leftarrow \mathcal{S}_t', \quad \tilde{\boldsymbol{b}}_{t+1} \leftarrow \tilde{\boldsymbol{v}}^\star, \quad \alpha_{t+1} \leftarrow m\alpha_t'/2.$$

In Figure 1, leftovers in each subsquare are reduced into the yellow point, and the procedure recurses on the smaller green square.

Final sampling. After T phases, sample permutations for the tuples in $\mathcal{S}_{T+1} \cup \{\tilde{\boldsymbol{b}}_{T+1}\}$ uniformly at random. Then backtrack through the recursion tree to recover permutations for all units, and hence the final assignment.

Example for $m = 3$. Let $n = 9, m = 3, p = 1$ and covariates $x_i = i$ for $i \in [9]$. Phase 1 forms triple

$(1,2,3),(4,5,6),(7,8,9)$. Within each triple, we place one unit in each arm by cyclically permuting $(x_i,0,0)$ into $(x_i,0,0),(0,x_i,0),(0,0,x_i)$ and summing across the three units. This cancels most of the local imbalance inside each triple. Phase 2 then groups the three resulting 3-tuples and applies another cyclic differencing step, further shrinking cross-arm differences. Backtracking returns the assignment of each unit.

Linear-time implementation. At phase t, the partition and differencing steps can be implemented in time $O(|\mathcal{S}_t|)$; the clean-up step is quadratic in the number of leftovers (which is small) and can be made $o(|\mathcal{S}_t|)$. Since each recursive phase shrinks $|\mathcal{S}_t|$ by a factor of m, the overall runtime is given by the geometric sum $O(n)+O(n/m)+O(n/m^2)+\cdots=O(n)$. Additional details are provided in Section D.3.

Comparison with KK and GKK. GKK+ extends the classical KK and its generalization GKK to a multi-arm and multi-dimensional covariate setting. The *main difference* lies in the CLEAN-UP step. Neither KK nor GKK can be directly applied to multi-arm and multi-dimensional covariates simultaneously. The Merge step generalizes the GKK Reduce algorithm to $m \geq 2$, and we provide a linear-time implementation of this step, which is new. The Shrink step introduces a new algorithm that is different from KK's binary differencing for $p=1$. Due to space constraints, we formally describe our algorithms and their performances in the appendix.

GKK+ heuristic for high-dimensional covariates. GKK+ requires that the covariate dimension p increases slowly relative to n. When p is large, we propose a heuristic based on recursive m-way matching. Rather than dividing a cube into subcubes to group similar vectors, as in PARTITION and DIFFERENCE, we directly cluster units into disjoint groups of size m by minimizing within-group distances. Same as GKK+, we compress each m-group and then recurse. We evaluate this algorithm empirically in Section 5.

Comparison with Arbour et al. (2022). Arbour et al. (2022) presents a general framework for constructing an m arm assignment ($m>2$) by recursively applying a two-arm design that minimizes a *weighted* discrepancy. This recursive decomposition sidesteps the need to directly balance all m groups simultaneously. By contrast, extending KK and its generalizations for two arms to the weighted setting remains challenging, since KK assigns units while treating the two arms symmetrically, preventing the kind of recursive reduction that Arbour et al. (2022) exploits.

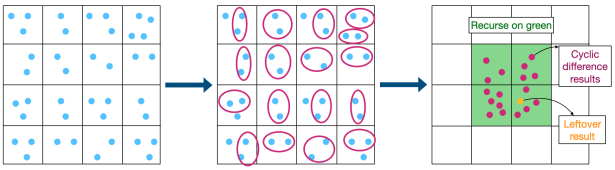

Figure 1: GKK+ phase for $m=2$ and $p=1$.

5 EXPERIMENTS

We compare GKK+ with commonly used designs: Bernoulli, Complete Randomization (CR), Rerandomization (Rerand)[3], Randomized Block Design based on the first two covariates (Azriel et al., 2022), and QuickBlock, which incorporates all covariates (Higgins et al., 2016). Simulation results are presented for four treatment arms ($m=4$) in the main text, with additional results provided in Appendix F. For the two-arm setting, Appendix F also compares GKK+ with Greedy Pair-Switching (Krieger et al., 2019), which is a strong balancing baseline but becomes computationally impractical beyond two arms.

Our code is available at `https://github.com/jjia131/GKK_plus`.

We consider two types of covariates $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$.

Random covariates. Covariate vectors $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n \in \mathbb{R}^p$ are drawn independently from a product distribution, where each coordinate follows the uniform distribution on $[-0.5,0.5]$ (rescaling the support does not affect the results). We test different values of m,p, and n. For each pair (m,p), the sample size n is set to $\lceil 10^k \rceil \times 20m$ with $k \in \{0.25,0.5,0.75,1,1.25,1.5,1.75\}$.

LaLonde dataset. We also conduct simulations with the LaLonde dataset (LaLonde, 1986; Dehejia and Wahba, 1999, 2002), which evaluates the effect of the National Supported Work Demonstration job training program on participant earnings. In our experiment, we do not analyze the original NSW randomized evaluation; instead, we use the CPS control sample from the LaLonde dataset only as a realistic source of baseline covariates. The CPS control dataset contains 15,992 units and 8 covariates, consisting of 4 binary and 4 numeric variables. We stratify on the binary covariates and restrict to the subgroup with binary covariates $(0,0,1,0)$, yielding 7,426 units with 4 numeric covariates. We treat this subset of units as a superpopulation, normalize each covariate to have mean 0 and standard deviation 1, and then generate sample populations by independently drawing n units with replacement.

[3]With a 1% exact acceptance probability and with the Mahalanobis distance.

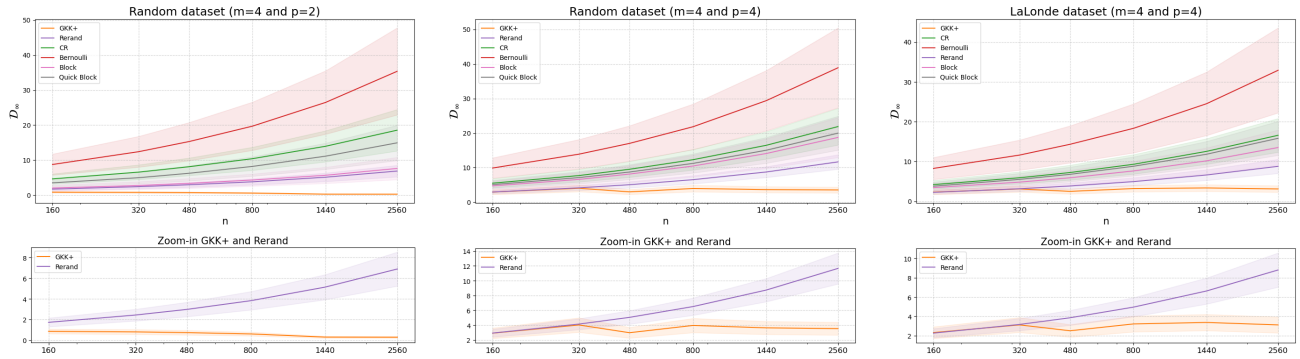

Figure 2: ℓ_∞ Discrepancy $\mathcal{D}_\infty$ for four treatment arms.

	Linear outcomes					Quadratic outcomes					Cubic outcomes					Sinusoidal outcomes				
n	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern
160	1.13	3.64	4.18	4.39	4.22	1.45	1.43	1.39	1.44	2.47	0.93	1.83	2.03	2.10	2.01	1.34	3.57	4.15	4.31	4.13
320	1.15	3.48	4.10	4.01	4.03	1.57	1.57	1.54	1.55	2.70	0.93	1.81	1.95	1.98	1.96	1.37	3.46	4.08	4.08	3.97
480	3.17	9.71	11.41	11.78	11.88	2.11	2.10	2.09	2.13	3.64	1.73	3.29	3.74	3.82	3.77	3.18	8.15	9.43	9.61	9.68
800	2.85	9.01	10.63	11.12	10.65	2.36	2.41	2.34	2.27	3.92	1.75	3.37	3.69	3.81	3.78	2.96	7.94	9.17	9.59	9.31
1440	5.84	18.64	22.16	22.61	21.42	2.32	2.39	2.36	2.29	3.87	2.47	4.83	5.38	5.48	5.17	4.51	12.39	14.64	15.17	14.11
2560	10.42	31.95	39.45	37.99	40.10	2.17	2.12	2.16	2.16	3.65	2.83	5.28	6.25	6.00	6.24	5.38	13.80	16.87	16.49	17.03

Table 1: Variance ratios relative to GKK+ for random dataset ($m = 4, p = 4$). The GKK+ column is always 1.00 and is omitted. RR: Rerandomization, Q-Block: Quick Block, CR: Complete Randomization, Bern: Bernoulli.

	Linear outcomes					Quadratic outcomes					Cubic outcomes					Sinusoidal outcomes				
n	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern	RR	Q-Block	Block	CR	Bern
160	1.09	1.80	1.82	1.89	3.74	1.12	1.57	1.56	1.62	2.64	1.12	1.46	1.46	1.53	1.96	1.12	1.74	1.75	1.78	2.74
400	1.30	2.05	2.14	2.14	4.52	1.22	1.65	1.75	1.76	2.97	1.20	1.56	1.66	1.65	2.11	1.26	1.94	1.97	1.95	3.19
640	1.28	2.17	2.00	2.23	4.82	1.29	1.83	1.68	1.87	3.18	1.22	1.70	1.55	1.72	2.24	1.29	2.06	2.03	2.08	3.38
880	1.42	2.33	2.32	2.36	5.21	1.37	1.93	1.90	1.95	3.37	1.33	1.75	1.76	1.78	2.33	1.38	2.17	2.16	2.17	3.61
1120	1.44	2.44	2.36	2.33	5.52	1.41	1.96	1.89	1.90	3.53	1.36	1.80	1.76	1.77	2.47	1.42	2.24	2.29	2.25	3.81

Table 2: Variance ratios relative to GKK+ heuristic ($m = 4, p = 20$).

5.1 Simulation Results

Balance metric ℓ_∞ discrepancy $\mathcal{D}_\infty$. For each (m, p, n), we generate 5,000 independent samples and report $\mathcal{D}_\infty$ (mean $\pm$ standard deviation) in Figure 2. For $p = 2$, Bernoulli and CR perform the worst, Block and Rerand are moderate, and GKK+ yields the lowest $\mathcal{D}_\infty$. For $p = 4$, Bernoulli remains worst, CR and Block are similar, Rerand and GKK+ are similar for small n, but GKK+ performs much better as n grows.

Variance of $\hat{\tau}_{kk'}$. We evaluate the variance of $\hat{\tau}_{kk'} - \tau_{kk'}$ and $\hat{\tau}_{kk'} - \tau^*_{kk'}$, running 5,000 simulations for each data type and (m, p, n) setting.

Outcomes are generated as follows. For random data, each unit $i \in [n]$ and arm $k \in [m]$ has outcome $Y_{ik} = f(\boldsymbol{x}_i) + \epsilon_{ik}$ with $\epsilon_{ik} \sim \mathcal{N}(0, 0.1^2)$, where f depends only on a subset of covariates (Kallus, 2018). For the LaLonde CSP control dataset, only control outcomes are observed. We treat the observed control outcomes as the potential outcomes for one arm. Under a no–treatment-effect assumption, we set the potential outcomes for all other arms equal to these observed outcomes. We then create a synthetic multi-arm trial by randomizing units to arms under different designs (GKK+, Bernoulli, etc.).

Table 1 reports the sampling variance of $\hat{\tau}_{1,2} - \tau_{1,2}$ for synthetic outcomes (other arm pairs (k, k') are similar). We defer the results for $\hat{\tau}_{kk'} - \tau^*_{kk'}$ to the appendix. For better comparison, we divide each variance by the corresponding GKK+ variance for each n, so that GKK+ is always 1 and thus omitted in the table. Across all four outcome types, GKK+ yields the lowest variance, except for small n with cubic out-

n	RR	Q-Block	Block	CR	Bern
160	0.99	1.28	1.50	1.52	1.50
320	1.03	1.38	1.50	1.49	1.50
480	1.10	1.51	1.62	1.63	1.63
800	1.12	1.59	1.67	1.72	1.66
1440	1.19	1.76	1.75	1.73	1.83
2560	1.21	1.87	1.80	1.77	1.83

Table 3: Variance ratios relative to GKK+ for LaLonde ($m = 4$).

comes where Rerand performs slightly better; GKK+ has substantial improvements for linear outcomes. Table 3 shows the results for the LaLonde dataset. Since we assume no treatment effect for any unit, we have $\tau_{kk'} = \tau^*_{kk'}$. The outcomes are normalized to have a mean of zero across the whole population. GKK+ consistently achieves the lowest variance across all values of n, with the exception that for $n = 160$, its variance is nearly identical to that of Rerand. As n increases, GKK+ has more variance reduction. In addition, we conduct a simulation with a large covariate dimension $p = 20$ in Table 2, comparing the GKK+ heuristic against the benchmark methods.

6 DISCUSSION

We introduced GKK+, a new design for multi-arm RCTs that achieves strong covariate balance while preserving robustness to model misspecification. We show that GKK+ yields exponentially smaller imbalance than Bernoulli randomization and supports valid asymptotic inference. Simulations on synthetic and real data confirm variance reduction across diverse outcome models. In higher-dimensional covariate settings, GKK+ may be less effective, but we propose a recursive m-wise matching heuristic that shows promising performance. The current paper focuses on the offline setting; it would be interesting to explore extensions to online or sequential designs.

Acknowledgments

This project was partially supported by NSF Grant CCF-2238682 and an Adobe Data Science Research Award. PZ would also like to thank Guanyang Wang for suggesting the design name GKK+.

References

David Arbour, Drew Dimmery, and Anup Rao. Efficient balanced treatment assignments for experimentation. In *International Conference on Artificial Intelligence and Statistics*, pages 3070–3078. PMLR, 2021.

David Arbour, Drew Dimmery, Tung Mai, and Anup Rao. Online balanced experimental design. In *International Conference on Machine Learning*, pages 844–864. PMLR, 2022.

David Azriel, Abba M Krieger, and Adam Kapelner. The optimality of blocking designs in equally and unequally allocated randomized experiments with general response. *arXiv preprint arXiv:2212.01887*, 2022.

Yuehao Bai. Optimality of matched-pair designs in randomized controlled trials. *American Economic Review*, 112(12):3911–3940, 2022.

Yuehao Bai, Joseph P Romano, and Azeem M Shaikh. Inference in experiments with matched pairs. *Journal of the American Statistical Association*, 117 (540):1726–1737, 2022.

József Beck and Tibor Fiala. "integer-making" theorems. *Discrete Applied Mathematics*, 3(1):1–8, 1981.

Dimitris Bertsimas, Mac Johnson, and Nathan Kallus. The power of optimization over randomization in designing experiments involving small samples. *Operations Research*, 63(4):868–876, 2015.

Nikhil Bhat, Vivek F Farias, Ciamac C Moallemi, and Deeksha Sinha. Near-optimal ab testing. *Management Science*, 66(10):4477–4495, 2020.

Bernard Chazelle. *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.

William Chen, Anand Srivastav, Giancarlo Travaglini, et al. *A panorama of discrepancy theory*, volume 2107. Springer, 2014.

Laurent Davezies, Guillaume Hollard, and Pedro Vergara Merino. Revisiting randomization with the cube method. *arXiv preprint arXiv:2407.13613*, 2024.

Angus Deaton and Nancy Cartwright. Understanding and misunderstanding randomized controlled trials. *Social science & medicine*, 210:2–21, 2018.

Rajeev H Dehejia and Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American statistical Association*, 94(448):1053–1062, 1999.

Rajeev H Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161, 2002.

Richard M Dudley. *Real analysis and probability*. Chapman and Hall/CRC, 2018.

Charles W Dunnett. A multiple comparison procedure for comparing several treatments with a control. *Journal of the American Statistical Association*, 50(272):1096–1121, 1955.

Ronald A Fisher. *The design of experiments*. London: Oliver & Boyd, 1935.

Michael R Garey and David S. Johnson. Complexity results for multiprocessor scheduling under resource constraints. *SIAM journal on Computing*, 4(4):397–411, 1975.

Robert Greevy, Bo Lu, Jeffrey H Silber, and Paul Rosenbaum. Optimal multivariate matching before randomization. *Biostatistics*, 5(2):263–275, 2004.

Allan Gut. *Probability: a graduate course*, volume 200. Springer, 2006.

Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. *Inequalities*. Cambridge university press, 1952.

Christopher Harshaw, Fredrik Sävje, Daniel A Spielman, and Peng Zhang. Balancing covariates in randomized experiments with the Gram-Schmidt walk design. *Journal of the American Statistical Association*, pages 1–13, 2024.

Michael J Higgins, Fredrik Sävje, and Jasjeet S Sekhon. Improving massive experiments with threshold blocking. *Proceedings of the National Academy of Sciences*, 113(27):7369–7376, 2016.

Dena R Howard, Julia M Brown, Susan Todd, and Walter M Gregory. Recommendations on multiple testing adjustment in multi-arm trials with a shared control group. *Statistical methods in medical research*, 27(5):1513–1530, 2018.

Kosuke Imai, Gary King, and Clayton Nall. The essential role of pair matching in cluster-randomized experiments, with application to the mexican universal health insurance evaluation. 2009.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Thomas Jaki and James MS Wason. Multi-arm multistage trials can improve the efficiency of finding effective treatments for stroke: a case study. *BMC cardiovascular disorders*, 18:1–8, 2018.

Nathan Kallus. Optimal a priori balance in the design of controlled experiments. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80 (1):85–112, 2018.

Narendra Karmarkar and Richard M Karp. *The differencing method of set partitioning*. Computer Science Division (EECS), University of California Berkeley, 1982.

Maximilian Kasy. Why experimenters might not always want to randomize, and what they could do instead. *Political Analysis*, 24(3):324–338, 2016.

Abba M Krieger, David Azriel, and Adam Kapelner. Nearly random designs with greatly improved balance. *Biometrika*, 106(3):695–701, 2019.

Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pages 604–620, 1986.

Michel Ledoux. Optimal matching of random samples and rates of convergence of empirical measures. In *Mathematics Going Forward: Collected Mathematical Brushstrokes*, pages 615–627. Springer, 2022.

Xinran Li and Peng Ding. Rerandomization and regression adjustment. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(1): 241–268, 2020.

Xinran Li, Peng Ding, and Donald B Rubin. Asymptotic theory of rerandomization in treatment-control experiments. *Proceedings of the National Academy of Sciences*, 115(37):9157–9162, 2018.

Rushil Mallarapu and Mark Sellke. Strong low degree hardness for the number partitioning problem. *arXiv preprint arXiv:2505.20607*, 2025.

Jiri Matousek. *Geometric discrepancy: An illustrated guide*, volume 18. Springer Science & Business Media, 1999.

Kari Lock Morgan and Donald B Rubin. Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, 2012.

Anup Rao and Peng Zhang. On distributional discrepancy for experimental design with general assignment probabilities. In *International Conference on Artificial Intelligence and Statistics*, pages 1486–1494. PMLR, 2025.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Student. Comparison between balanced and random arrangements of field plots. *Biometrika*, pages 363–378, 1938.

Paxton Turner, Raghu Meka, and Philippe Rigollet. Balancing Gaussian vectors in high dimension. In *Conference on Learning Theory*, pages 3455–3486. PMLR, 2020.

Neekon Vafa and Vinod Vaikuntanathan. Symmetric perceptrons, number partitioning and lattices. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 2191–2202, 2025.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A ROADMAP OF THE APPENDIX

In Section B, we present the CLEAN-UP step of the GKK+ Algorithm. In Sections C and D, we establish covariate balance and robustness of GKK+ and prove Theorem 3.2. In Section E, we analyze statistical properties of GKK+ for estimating the average treatment effect and provide proofs for Theorems 3.3, 3.4, and 3.5. In Section F, we include additional details and results of experiments.

B GKK+ ALGORITHM CLEAN-UP STEP

In our algorithm and analysis, we always let

$$\gamma \stackrel{\text{def}}{=} 2m^3 p. \tag{2}$$

Let N_t be the number of subcubes formed in phase t during the PARTITION step.

In this section, we describe the CLEAN-UP$(\mathcal{S}'_t, \mathcal{L}_t, \tilde{\boldsymbol{b}}_t, \alpha'_t)$ step of the GKK+ algorithm (see Section 4). We omit the subscript t since we focus only on phase t. The inputs are as follows: $\mathcal{S}'$ denotes the set of cyclic difference vectors generated by DIFFERENCE; $\mathcal{L} = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s\}$ is the set of leftover vectors from DIFFERENCE; $\tilde{\boldsymbol{b}} \in \mathbb{R}^{mp}$ represents the residual m-tuple that may have a large norm; and $\alpha' \in \mathbb{R}$ is a parameter ensuring that all elements of $\mathcal{S}'$ lie within $[-\frac{m\alpha'}{2}, \frac{m\alpha'}{2}]^{mp}$. At the end of CLEAN-UP (if not terminated early), all remaining vectors lie within $[-\gamma\alpha', \gamma\alpha']^{mp}$.

CLEAN-UP consists of two steps. *Merge step:* construct an m-tuple $\tilde{\boldsymbol{v}}^*$ by adding $\tilde{\boldsymbol{b}}$ to a sum of permuted leftover vectors from $\mathcal{L}$, ensuring that the norm of the resulting $\tilde{\boldsymbol{v}}^*$ remains comparable to that of $\tilde{\boldsymbol{b}}$. *Shrink step:* reduce the norm of $\tilde{\boldsymbol{v}}^*$ by repeatedly combining it with m-tuples sampled uniformly at random from $\mathcal{S}'$.

B.1 Merge Step

The merge step is presented in Algorithm 1. It generalizes the classical Beck-Fiala algorithm (Beck and Fiala, 1981), which assigns vectors to two arms with low discrepancy, to a setting with multiple arms. Intuitively, determining s permutations for the leftover vectors $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s$ can be formulated as an integer linear program, which is NP-hard in general. Instead of solving this program directly, the algorithm performs a walk within the polytope defined by the corresponding linear constraints and eventually reaches an integer vertex, which is returned as the solution.

Observe that subtracting a common vector from each component of an m-tuple does not change their pairwise differences (and hence their ℓ_∞ discrepancy). Following Karmarkar and Karp (1982), we define the *reduced* form of an m-tuple $\tilde{\boldsymbol{v}}$ as

$$\underline{\tilde{\boldsymbol{v}}} \stackrel{\text{def}}{=} (\tilde{\boldsymbol{v}}^{(1)} - \boldsymbol{x}, \ldots, \tilde{\boldsymbol{v}}^{(m)} - \boldsymbol{x}) \in \mathbb{R}^{mp}_{\geq 0},$$

where $\boldsymbol{x} \in \mathbb{R}^p$ has its ith entry equal to the minimum among the ith entries of $\tilde{\boldsymbol{v}}^{(1)}, \ldots, \tilde{\boldsymbol{v}}^{(m)}$.

Theorem B.1. *Consider m-tuples $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s \in [-\alpha, \alpha]^{mp}$ and $\tilde{\boldsymbol{b}} \in [-\gamma\alpha, \gamma\alpha]^{mp}$. Algorithm 1, when applied to $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s$, produces permutations $\sigma_1, \ldots, \sigma_s$ such that*

$$\underline{\sigma_1 \circ \tilde{\boldsymbol{v}}_1 + \cdots + \sigma_s \circ \tilde{\boldsymbol{v}}_s + \tilde{\boldsymbol{b}}} \in [0,\, 2mp\alpha + 2\gamma\alpha]^{mp}. \tag{3}$$

Moreover, Algorithm 1 can be implemented in $O(p^3 s^2)$ time, where the hidden constants depend only on m.

Algorithm 1 MERGE($\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s$)

Input: m-tuples $\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s \in \mathbb{R}^{mp}$

1: Let $M \leftarrow m!$ and $\mathcal{P}_m$ be the set of all permutations over $[m]$.

2: Let

$$\boldsymbol{U} = \left(\sigma \circ \tilde{\boldsymbol{v}}_i : \sigma \in \mathcal{P}_m, i \in [s]\right) \in \mathbb{R}^{(mp)\times(Ms)}$$
$$\boldsymbol{D} = \mathrm{DIAG}(\boldsymbol{1}_M^\top, \ldots, \boldsymbol{1}_M^\top) \in \mathbb{R}^{s\times(Ms)}$$
$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{D} \end{pmatrix} \in \mathbb{R}^{(mp+s)\times(Ms)}$$

3: Let $\boldsymbol{\lambda} \leftarrow \frac{1}{M}\boldsymbol{1} \in \mathbb{R}^{Ms}$ and $\mathcal{A} \leftarrow [Ms]$. ▷ alive varaibles

4: **while** $\mathcal{A}$ is non-empty **do**

5: Let $\boldsymbol{y} \in \mathbb{R}^{\mathcal{A}} \setminus \{0\}$ satisfy $\boldsymbol{W}(:, \mathcal{A})\boldsymbol{y} = \boldsymbol{0}$. Break the while-iteration if no such $\boldsymbol{y}$ exists.

6: Let $\mu^* \leftarrow \max\{\mu \in \mathbb{R} : \boldsymbol{\lambda}(\mathcal{A}) + \mu\boldsymbol{y} \in [0,1]^{\mathcal{A}}\}$. Let $\boldsymbol{\lambda}(\mathcal{A}) \leftarrow \boldsymbol{\lambda}(\mathcal{A}) + \mu^*\boldsymbol{y}$.

7: Let $\mathcal{A} \leftarrow \{i \in [Ms] : \lambda_i \in (0,1)\}$.

8: **end while**

9: Assign $\{\lambda_i : i \in \mathcal{A}\}$ arbitrarily to $\{0,1\}$, subject to $\boldsymbol{D}\boldsymbol{\lambda} = \boldsymbol{1}$.

10: For each $i \in [n], j \in [M]$, let σ_i be the jth permutation in $\mathcal{P}_m$ if $\lambda_{(i-1)M+j} = 1$.

Return: $\sigma_1, \ldots, \sigma_s$.

Remark. Algorithm 1 does not run in linear time. However, in each phase t, the number of leftover vectors s satisfies

$$s \le mN_t \le m \cdot 2^{mp} \cdot n_t^{1/4}.$$

The runtime of the Merge step is bounded by

$$O(1) \cdot p^3 s^2 \le O(1) \cdot m^2 p^3 2^{2mp} n_t^{1/2} = o(n_t).$$

Hence, the total runtime of the Merge step across all phases is at most $o(n)$.

Proof. We first prove Eq. (3) and then present a fast implementation of Algorithm 1.

Low discrepancy. We claim that when the while-loop on line 4 of Algorithm 1 terminates, there are at most $2mp$ alive variables in $\boldsymbol{\lambda}$. The loop can terminate in two ways: either $\mathcal{A}$ becomes empty (i.e., no alive variables remain), or no vector $\boldsymbol{y}$ satisfies the condition on line 5. The claim holds trivially in the first case. It remains to consider the second case, which happens only if $|\mathcal{A}| \le r_{\mathcal{A}}$, the number of nonzero rows in $\boldsymbol{W}(:, \mathcal{A})$, denoted by $r_{\mathcal{A}}$. To upper bound $r_{\mathcal{A}}$, consider each m-tuple $\tilde{\boldsymbol{v}}_i$ for $i \in [s]$. We say $\tilde{\boldsymbol{v}}_i$ is alive if there exists some $j \in [M]$ such that $\lambda_{(i-1)M+j}$ is alive. Since the algorithm enforces $\sum_{j\in[M]} \lambda_{(i-1)M+j} = 1$ and $\lambda \in [0,1]^{Ms}$, the fact that $\tilde{\boldsymbol{v}}_i$ is alive implies that at least two variables in $\{\lambda_{(i-1)M+j} : j \in [M]\}$ must be alive. Therefore, the number of alive m-tuples (i.e., the number of nonzero rows in $D_{\mathcal{A}}$) is at most $|\mathcal{A}|/2$. Combining this with the contribution from $U_{\mathcal{A}}$, we obtain the bound $r_{\mathcal{A}} \le |\mathcal{A}|/2 + mp$. Thus, the while-iteration terminates at line 5 only if

$$\frac{|\mathcal{A}|}{2} + mp \ge |\mathcal{A}| \iff |\mathcal{A}| \le 2mp.$$

Now, we are ready to prove the theorem. Before the while-iteration terminates, Algorithm 1 always guarantees that $\boldsymbol{U}\boldsymbol{\lambda} = \frac{1}{M}\boldsymbol{U}\boldsymbol{1}$. At the termination of the while-iteration, there are at most mp alive m-tuples. By the assignment for these alive m-tuples, we have

$$\boldsymbol{U}\boldsymbol{\lambda} - \frac{1}{M}\boldsymbol{U}\boldsymbol{1} \in [-mp\alpha, mp\alpha]^{mp}.$$

Therefore, the permutations $\sigma_1, \ldots, \sigma_s$ induced by $\boldsymbol{\lambda}$ satisfy

$$\underline{\sigma_1 \circ \tilde{\boldsymbol{v}}_1 + \cdots + \sigma_n \circ \tilde{\boldsymbol{v}}_s} \in [0, 2mp\alpha]^{mp}.$$

Together with the range of $\tilde{\boldsymbol{b}}$, Eq. (3) holds.

Fast implementation. The main computational bottleneck in implementing Algorithm 1 is at line 5, which requires finding a vector $\boldsymbol{y}$ orthogonal to the rows of $\boldsymbol{W}(:,\mathcal{A}) \in \mathbb{R}^{(mp+s)\times O(Ms)}$, or reporting that no such $\boldsymbol{y}$ exists. A naive implementation takes $\Omega(s^2)$ time in the worst case, leading to a total runtime of $\Omega(s^3)$. We show that, by exploiting the special structure of $\boldsymbol{D}$, which forms part of $\boldsymbol{W}$, line 5 can be implemented in $O(s)$ time (presented in Algorithm 2).

Forming the matrices $\boldsymbol{U}$ and $\boldsymbol{D}$ requires $O(ps)$ time. The number of while-loop iterations is at most $O(s)$, since each iteration deactivates at least one alive variable. Within each iteration, line 5 can be implemented in $O(p^3 s)$ time using Algorithm 2, and the remaining steps also run in $O(ps)$ time. Therefore, the overall runtime is $O(p^3 s^2)$. □

Algorithm 2 NullVec($\boldsymbol{U}_{\mathcal{A}}, \boldsymbol{D}_{\mathcal{A}}$)

Input: matrices $\boldsymbol{U}_{\mathcal{A}} \in \mathbb{R}^{(mp)\times|\mathcal{A}|}$ and $\boldsymbol{D}_{\mathcal{A}} \in \mathbb{R}^{s'\times|\mathcal{A}|}$, where $\boldsymbol{D}_{\mathcal{A}} = \text{diag}(\mathbf{1}_{M_1}^\top, \ldots, \mathbf{1}_{M_{s'}}^\top)$ is block diagonal with each diagonal block being a single row vector.

1: $\mathcal{U} \leftarrow \emptyset$.
2: **for** each row vector $\boldsymbol{a}$ of $\boldsymbol{U}_{\mathcal{A}}$ **do** ▷ Gram-Schmidt orthogonalization
3: $\quad \boldsymbol{a} \leftarrow \boldsymbol{a} - \text{Proj}(\boldsymbol{D}_{\mathcal{A}}, \mathcal{U}, \boldsymbol{a})$.
4: $\quad \mathcal{U} \leftarrow \mathcal{U} \cup \{\boldsymbol{a}/\|\boldsymbol{a}\|\}$.
5: **end for**
6: Let $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{A}|}$ be a random vector.

Return: $\boldsymbol{y} - \text{Proj}(\boldsymbol{D}_{\mathcal{A}}, \mathcal{U}, \boldsymbol{y})$.

7: **function** Proj($\boldsymbol{D}_{\mathcal{A}}, \mathcal{U}, \boldsymbol{a}$)
8: $\quad$ Let $\boldsymbol{a}' \in \mathbb{R}^{|\mathcal{A}|}$ be a vector to be determined.
9: $\quad$ For each $i \in [s']$, let $j_1 = \sum_{j<i} M_j + 1$ and $j_2 = j_1 + M_i - 1$, and let

$$\boldsymbol{a}'(j_1 : j_2) \leftarrow \frac{\langle \boldsymbol{a}(j_1 : j_2), \mathbf{1}_{M_i} \rangle}{M_i} \mathbf{1}_{M_i}.$$

10: $\quad$ For each $\boldsymbol{u} \in \mathcal{U}$, let $\boldsymbol{a}' \leftarrow \boldsymbol{a}' + \langle \boldsymbol{a}, \boldsymbol{u} \rangle \boldsymbol{u}$.
11: $\quad$ Return $\boldsymbol{a}'$.
12: **end function**

B.2 Shrink Step

Let

$$\tilde{\boldsymbol{b}} \leftarrow \underline{\sigma_1 \circ \tilde{\boldsymbol{v}}_1 + \cdots + \sigma_s \circ \tilde{\boldsymbol{v}}_s + \tilde{\boldsymbol{b}}},$$

where $\sigma_1, \ldots, \sigma_s$ are returned by Algorithm 1 Merge($\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_s$). We will reduce the length of $\tilde{\boldsymbol{b}}$ by elements randomly drawn from $\mathcal{S}'$. A pseudo-code is presented in Algorithm 3.

Algorithm 3 Shrink($\tilde{\boldsymbol{b}}, \mathcal{S}', \alpha', \gamma$)

Input: m-tuples $\tilde{\boldsymbol{b}} \in \mathbb{R}^{mp}$, $\mathcal{S}' \subset \mathbb{R}^{mp}$, and $\alpha', \gamma > 0$

1: Let $\tilde{\boldsymbol{u}}_1, \ldots, \tilde{\boldsymbol{u}}_s$ be a list of m-tuples obtained by randomly permuting the elements in $\mathcal{S}'$.
2: $i \leftarrow 1$.
3: **while** $\|\tilde{\boldsymbol{b}}\|_\infty > \gamma\alpha'$ and $i \le s$ **do**
4: $\quad$ If $\langle \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{u}}_i \rangle < 0$, let $\tilde{\boldsymbol{b}} \leftarrow \tilde{\boldsymbol{b}} + \tilde{\boldsymbol{u}}_i$ and $\mathcal{S}' \leftarrow \mathcal{S}' \setminus \{\tilde{\boldsymbol{u}}_i\}$.
5: $\quad i \leftarrow i + 1$.
6: **end while**

Return: $\tilde{\boldsymbol{b}}, \mathcal{S}'$.

Algorithm 3 is similar to step 4(b) of the Clean-Up procedure in Turner et al. (2020). However, unlike Turner et al. (2020), where both addition and subtraction of $\tilde{\boldsymbol{u}}_i$ are allowed, here we are restricted to adding $\tilde{\boldsymbol{u}}_i$ to $\tilde{\boldsymbol{b}}$. In order to decrease the norm of $\tilde{\boldsymbol{b}}$, we only add those $\tilde{\boldsymbol{u}}_i$ satisfying $\langle \tilde{\boldsymbol{b}}, \tilde{\boldsymbol{u}}_i \rangle < 0$.

We further claim that the while-loops in Algorithm 3 use only a small fraction of the vectors in $\mathcal{S}'$. The proof of this claim relies on the distributional properties of vectors in $\mathcal{S}'$, which we defer to Section C.4.

C INDUCTIVE ASSERTIONS ABOUT GKK+

In this section and the next, we present a sequence of lemmas that lead to the proof of Theorem 3.2. Our analysis builds upon the framework introduced in Karmarkar and Karp (1982) and further developed in Turner et al. (2020), which is based on a sequence of inductive assertions. However, because of our modified Clean-Up step and the different distribution of cyclic difference vectors, several parts of our proofs differ from those in the earlier works.

We formally present Algorithm GKK+ in Algorithm 4.

Algorithm 4 GKK+(T)

Input: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in [-\Delta, \Delta]^p$, $m \geq 2$, and $T \leq \lfloor \frac{0.1 \log n}{mp \log m} \rfloor$.

1: For each $i \in [n]$, lift $\boldsymbol{x}_i$ to an m-tuple $\tilde{\boldsymbol{v}}_i = (\boldsymbol{x}_i, \mathbf{0}, \ldots, \mathbf{0})$.

2: $\mathcal{S}_1 \leftarrow \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_n\}, \alpha_1 = \Delta, \tilde{\boldsymbol{b}}_1 = \mathbf{0}$

3: **for** $t = 1, \ldots, T$ **do**

4: Let

$$\mathcal{C}_t, \alpha'_t \leftarrow \textsc{Partition}(\mathcal{S}_t, \alpha_t)$$
$$\mathcal{S}'_t, \mathcal{L}_t \leftarrow \textsc{Difference}(\mathcal{S}_t, \mathcal{C}_t)$$
$$\mathcal{S}_{t+1}, \tilde{\boldsymbol{b}}_{t+1} \leftarrow \textsc{Clean-Up}(\mathcal{S}'_t, \mathcal{L}_t, \tilde{\boldsymbol{b}}_t, \alpha'_t)$$

5: If $\mathcal{S}_{t+1} = \emptyset$, return $\{\tilde{\boldsymbol{b}}_{t+1}\}$.

6: Let $\alpha_{t+1} \leftarrow \frac{m\alpha'_t}{2}$.

7: **end for**

8: Let $\sigma_{\tilde{\boldsymbol{v}}}$'s be i.i.d. random permutations over $[m]$, and let $\tilde{\boldsymbol{u}} \leftarrow \sum_{\tilde{\boldsymbol{v}} \in \mathcal{S}_{T+1} \cup \{\tilde{\boldsymbol{b}}_{T+1}\}} \sigma_{\tilde{\boldsymbol{v}}} \circ \tilde{\boldsymbol{v}}$.

9: Using standard backtracking, obtain the assignment $\boldsymbol{W} \in [m]^n$ for the n units.

Return: $\boldsymbol{W}$.

Proof intuition. We begin with a simplified and idealized setting under the following assumptions: (1) GKK+(T) runs for T iterations (i.e., the algorithm does not terminate at line 5); (2) in every phase $t \leq T$, the algorithm reduces the lengths of all vectors in $\mathcal{S}_t$ by a factor of $n^{-\Omega(1/(mp))}$.

After T phases, the remaining vectors have length at most $n^{-\Omega(T/(mp))}$. Assigning these remaining vectors at random yields an ℓ_∞ discrepancy $\mathcal{D}_\infty$ of at most $n^{1/2-\Omega(T/(mp))}$. In particular, if we take $T = \Theta\Big(\frac{\log n}{mp \log m}\Big)$, the resulting ℓ_∞ discrepancy $\mathcal{D}_\infty$ is $n^{-\Theta\left(\frac{\log n}{(mp)^2 \log m}\right)}$.

The central task is therefore to prove that, with high probability, the GKK+(T) algorithm successfully completes T phases, which guarantees the desired reduction in $\mathcal{D}_\infty$.

C.1 A Conceptual Resampling Step

We introduce a *conceptual* Resampling step between the Partition and Difference steps to ensure that, at the beginning of each phase t, a large fraction of the vectors in $\mathcal{S}_t$ are conditionally independent and possess sufficiently smooth densities. This Resampling step is implicit in Karmarkar and Karp (1982) and was made explicit in Turner et al. (2020). The latter assumes knowledge of the input distribution of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, but reduces the exponent in the ℓ_∞ discrepancy by a factor of mp.

Resampling. For each phase t, we label the vectors in $\mathcal{S}_t$ as "good" and "bad", which partitions $\mathcal{S}_t = \mathcal{G}_t \cup \mathcal{B}_t$ where $\mathcal{G}_t$ is the set of all the good vectors and $\mathcal{B}_t$ the bad vectors. Initially, all vectors in $\mathcal{S}_1$ are good (i.e., $\mathcal{G}_1 = \mathcal{S}_1$). For $t \geq 1$, let $f_t : [-\alpha_t, \alpha_t]^{mp} \to \mathbb{R}$ be the density of the vectors in $\mathcal{G}_t$ (in particular, f_1 is the density

function of the m-tuple $(\mathbf{0}, \ldots, \mathbf{0}, \boldsymbol{x}_1)$). For each $C \in \mathcal{C}_t$ where $\min_{\tilde{\boldsymbol{v}} \in C} f_t(\tilde{\boldsymbol{v}}) > 0$, let

$$f_{t,C} \stackrel{\text{def}}{=} \min_{\tilde{\boldsymbol{v}} \in C : f_t(\tilde{\boldsymbol{v}}) > 0} f_t(\tilde{\boldsymbol{v}}).$$

Let

$$g_t(\tilde{\boldsymbol{v}}) \stackrel{\text{def}}{=} \begin{cases} f_{t,C}, & \text{if } \tilde{\boldsymbol{v}} \in C \text{ for some } C \in \mathcal{C}_t \text{ where } f_{t,C} \text{ is defined} \\ 0, & \text{otherwise} \end{cases}$$

For each $\tilde{\boldsymbol{v}} \in \mathcal{G}_t$, we independently and randomly relabel it as bad with probability $1 - \frac{g_t(\tilde{\boldsymbol{v}})}{f_t(\tilde{\boldsymbol{v}})}$. A bad vector always remains bad.

In the DIFFERENCE step, if a vector is a cyclic difference of m good vectors, we label it as good; otherwise, we label it as bad.

Notations. We keep track of the good vectors and bad vectors during each phase t. Let $\mathcal{G}_t, \mathcal{G}_t', \mathcal{G}_t''$ be the sets of good vectors at the beginning of phase t, after conceptual RESAMPLING, and after DIFFERENCE, respectively. Note that $\mathcal{G}_t$ is also the set of good vectors after CLEAN-UP at phase $t-1$ for $t \geq 2$. Let $\mathcal{B}_t, \mathcal{B}_t', \mathcal{B}_t''$ be the corresponding sets for bad vectors. In addition, let $n_t = |\mathcal{G}_t \cup \mathcal{B}_t| = |\mathcal{S}_t|$. When the context is clear, we drop the subscript t.

C.1.1 Distribution of Good Vectors after Resampling

Turner et al. (2020) proved the following lemma for the distribution of the vectors in $\mathcal{G}_t'$ after (conceptual) RESAMPLING. It states that after RESAMPLING, good vectors are conditionally independent and uniform.

Lemma C.1 (Lemma 12 in Turner et al. (2020))**.** *Suppose that, conditional on an event $\mathcal{F}$, the random vectors in $\mathcal{G} = \{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_n\} \subset [-\alpha, \alpha]^{mp}$ are i.i.d., each with conditional joint density f. Let $\mathcal{B} \subset [-\alpha, \alpha]^{mp}$ be another set of vectors. Let $\mathcal{C}$ be the collection of subcubes returned by* PARTITION*$(\alpha, \mathcal{G} \cup \mathcal{B})$. Define $\mathcal{T}_{\mathcal{G},\alpha,f}$ as the configuration that records how the vectors in $\mathcal{G}$ are assigned to the subcubes in $\mathcal{C}$. Let $\mathcal{G}' \subset \mathcal{G}$ be the subset of vectors relabeled as good by* RESAMPLING*. Then, conditional on $\mathcal{F}$ and $\mathcal{T}_{\mathcal{G},\alpha,f}$, the following hold:*

1. *The vectors in $\mathcal{G}'$ are mutually independent and independent of those in $\mathcal{G} \setminus \mathcal{G}'$.*
2. *For each subcube $C \in \mathcal{C}$, a given good vector in $C \cap \mathcal{G}'$ is uniformly distributed over $C \cap \mathit{supp}(f)$.*

Lemma C.1 implies the following. In the first phase, for each $C \in \mathcal{C}_1$ such that $C \cap \mathrm{supp}(f_1) \neq \emptyset$, we have

$$C \cap \mathrm{supp}(f_1) = ([0, \alpha_1')^p + \boldsymbol{z}) \times \{0\}^{(m-1)p},$$

where each entry of $\boldsymbol{z} \in \mathbb{R}^p$ is a multiple of α_1'. Conditional on earlier steps, the vectors in $\mathcal{G}_1' \cap C$ are i.i.d. and uniformly distributed over $C \cap \mathrm{supp}(f_1)$. In phase $t \geq 2$, Lemmas C.2 and C.3 (in the next subsection) show that for each $C \in \mathcal{C}_t$, $C \cap \mathrm{supp}(f_t) = C$. Therefore, conditional on previous steps, the vectors in $\mathcal{G}_t' \cap C$ are i.i.d. and uniformly distributed over C.

C.2 Distribution of Random Cyclic Difference of Good Vectors

Given Lemma C.1, we can characterize the distributions of the random cyclic differences of good vectors after applying DIFFERENCE. For simplicity, we separate the first phase and the remaining phases.

Lemma C.2. *In the first phase, conditional on* PARTITION *and conceptual* RESAMPLING*, run* DIFFERENCE*, the vectors in $\mathcal{G}_1''$ are i.i.d. random vectors in $[-\frac{\alpha_1'}{2}, \frac{\alpha_1'}{2}]^{mp}$ whose entries are independently and uniformly distributed over $[-\frac{\alpha_1'}{2}, \frac{\alpha_1'}{2}]$.*

Proof. The proof follows that, conditional on previous steps, the difference vectors in the first phase have the form $(\boldsymbol{x}_{i_1} - \boldsymbol{c}, \ldots, \boldsymbol{x}_{i_m} - \boldsymbol{c})$ where $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_m}$ are independently and uniformly distributed over $[0, \alpha_1']^p$ and $\boldsymbol{c} = \{-\frac{\alpha_1'}{2}\}^p$. □

In phase $t \geq 2$, we show that in DIFFERENCE, the cyclic difference of m good vectors has entries that are i.i.d. samples from a scaled and shifted version of the *Irwin–Hall (IH) distribution*. The IH distribution with parameter m is the continuous distribution of the sum of m i.i.d. random variables uniformly distributed over $(0, 1)$. Its density is given by

$$f_{IH}(x) = \begin{cases} \frac{1}{(m-1)!} \sum_{k=0}^{\lfloor x \rfloor} (-1)^k \binom{m}{k} (x-k)^{m-1}, & 0 \leq x \leq m \\ 0, & \text{otherwise} \end{cases}$$

For any fixed m, the density f_{IH} is Lipschitz, with the Lipschitz constant depending only on m. Moreover, there exists a constant $D = D(m)$ such that $f_{IH}(x) \leq D$ for all $x \in [0, m]$.

Lemma C.3. *For each $t \geq 2$, conditional on* PARTITION *and* RESAMPLING*, run* DIFFERENCE*, the vectors in $\mathcal{G}_t''$ are i.i.d. random vectors in $[-\frac{m\alpha_t'}{2}, \frac{m\alpha_t'}{2}]^{mp}$ with a common density $\rho^{\otimes mp}$ where $\rho : [-\frac{m\alpha_t'}{2}, \frac{m\alpha_t'}{2}] \to \mathbb{R}$ with $\rho(y) = \frac{1}{\alpha_t'} f_{IH}(\frac{y}{\alpha_t'} + \frac{m}{2})$. We refer to the distribution with density ρ as* the centered IH distribution with parameters m, α.

The proof of Lemma C.3 follows directly from Lemma C.1 and the IH distribution.

C.3 Only a Few Good Points are Relabeled as Bad in Resampling

We claim that after RESAMPLING only relabels a small portion of good vectors as bad.

Lemma C.4. *For $t \geq 1$, conditional on $|\mathcal{G}_t| \geq n^{\frac{1}{mp}}$, run conceptual* RESAMPLING*,*

$$\Pr\left(|\mathcal{G}_t'| \geq (1 - 2n_t^{-\frac{1}{4mp}}) |\mathcal{G}_t| \right) \geq 1 - n^{-10}.$$

We need the following lemma on the product of Lipschitz functions.

Lemma C.5 (Lemma 9 of Turner et al. (2020))**.** *Let $\rho : [-\alpha, \alpha] \to \mathcal{R}$ be a probability density function that is Lipschitz with constant L and bounded above by some constant $D > 0$. Let $g = \rho^{\otimes m} : [-\alpha, \alpha]^m \to \mathbb{R}$ be the density of m i.i.d. random variables with common density ρ. Then, g is Lipschitz with parameter LmD^{m-1} in the ℓ_∞ norm: $|g(\boldsymbol{x}) - g(\boldsymbol{y})| \leq LmD^{m-1} \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$ for any $\boldsymbol{x}, \boldsymbol{y} \in [-\alpha, \alpha]^m$.*

Proof of Lemma C.4. We first calculate the probability that a given vector $\tilde{\boldsymbol{v}} \in \mathcal{G}_t$ is relabeled as bad at RESAMPLING and then apply Hoeffding's inequality to all the vectors in $\mathcal{G}_t$ as they are independently relabeled.

$$\begin{aligned} \Pr(\tilde{\boldsymbol{v}} \text{ relabeled bad} \mid \tilde{\boldsymbol{v}} \in \mathcal{G}_t) &= \sum_{C \in \mathcal{C}_t} \int_{C \cap \mathrm{supp}(f_t)} f_t(\tilde{\boldsymbol{v}}) \left(1 - \frac{f_{t,C}}{f_t(\tilde{\boldsymbol{v}})}\right) d\tilde{\boldsymbol{v}} \\ &= \sum_{C \in \mathcal{C}_t} \int_{C \cap \mathrm{supp}(f_t)} (f_t(\tilde{\boldsymbol{v}}) - f_{t,C})\, d\tilde{\boldsymbol{v}}. \end{aligned} \tag{4}$$

For $t = 1$, we have $|f_1(\tilde{\boldsymbol{v}}) - f_{1,C}| \leq O(\alpha_1')$. Thus,

$$\Pr(\tilde{\boldsymbol{v}} \text{ relabeled bad} \mid \tilde{\boldsymbol{v}} \in \mathcal{G}_1) \leq \sum_{C \in \mathcal{C}_1} \mathrm{Vol}(C \cap \mathrm{supp}(f_1)) \cdot O(\alpha_1') \leq O(2^p n^{-\frac{1}{4mp}}).$$

For $t \geq 2$, by Lemma C.3, $f_t = \rho_t^{\otimes(mp)}$ where $\rho_t(x) = \frac{1}{\alpha_{t-1}'} f_{IH}(\frac{x}{\alpha_{t-1}'} + \frac{m}{2})$ for $x \in [-\frac{m\alpha_{t-1}'}{2}, \frac{m\alpha_{t-1}'}{2}]$ and $\rho_t(x) = 0$ otherwise. We can check that for any $x, y \in [-\frac{m\alpha_{t-1}'}{2}, \frac{m\alpha_{t-1}'}{2}]$,

$$|\rho_t(x) - \rho_t(y)| \leq \max_z |\rho_t'(z)| \cdot |x - y| \leq \frac{1}{\Omega((\alpha_{t-1}')^2)} |x - y|,$$

where $\Omega(\cdot)$ hides constant only depending on m. In addition, for any $x \in [-\frac{m\alpha_{t-1}'}{2}, \frac{m\alpha_{t-1}'}{2}]$,

$$\rho_t(x) = \frac{1}{\Omega(\alpha_{t-1}')}.$$

Thus, by Lemma C.5, for each $C \in \mathcal{C}_t$,

$$\forall \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{u}} \in C,\ |f_t(\tilde{\boldsymbol{v}}) - f_t(\tilde{\boldsymbol{u}})| \le O(mp(\alpha'_{t-1})^{-(mp+1)}) \left\|\tilde{\boldsymbol{v}} - \tilde{\boldsymbol{u}}\right\|_\infty \le O\left(\frac{mp\alpha'_t}{(\alpha'_{t-1})^{mp+1}}\right).$$

Plugging into Eq. (4),

$$\Pr(\tilde{\boldsymbol{v}} \text{ relabeled bad} \mid \tilde{\boldsymbol{v}} \in \mathcal{G}_t) \le O(1) \cdot (m\alpha'_{t-1})^{mp} \cdot \frac{mp\alpha'_t}{(\alpha'_{t-1})^{mp+1}} = O\left(\frac{\alpha'_t}{\alpha'_{t-1}}\right) = O\left(n_t^{-\frac{1}{4mp}}\right),$$

where n_t is the number of input vectors for phase t. The statement holds by applying Hoeffding's inequality. □

C.4 Clean-Up Shrink Step

We claim that the Shrink step in Clean-Up does not use many vectors from $\mathcal{S}'_t$.

Lemma C.6. *Assume $|\mathcal{B}''_t| = o(1)\,|\mathcal{G}''_t|$. Let $\tilde{\boldsymbol{b}} \in [0, 2mp\alpha_t + 2\gamma\alpha_t]^{mp}$. Define a sequence of random vectors: $\tilde{\boldsymbol{b}}_0 = \tilde{\boldsymbol{b}}$, and for $k = 1, 2, \ldots$, let $\tilde{\boldsymbol{b}}_k$ the vector $\tilde{\boldsymbol{b}}$ at the kth iteration of Algorithm 3. Fix $K = Cn_t^{\frac{1}{2mp}}$ where $C = C(m, p)$ is a sufficiently large constant. With probability at least $1 - \exp(-\Theta(n_t^{\frac{1}{3mp}}))$, there exists $i \le K$ such that $\|\tilde{\boldsymbol{b}}_k\|_\infty \le \gamma\alpha'_t$, where $\gamma = 2m^3p$.*

The proof is a modification of Lemma 11 in Turner et al. (2020), with the additional consideration of vectors drawn from the set $\mathcal{B}''_t$ and the vectors $\boldsymbol{u}_i$ that are not combined with $\tilde{\boldsymbol{b}}$.

Proof. Let $\mathcal{E}_1$ be the event that $\mathcal{I}_K = \{1 \le k \le K : \boldsymbol{u}_k \in \mathcal{G}''_t\}$ has cardinality $(1 - o(1))K$. By Hoeffding's inequality,

$$\Pr(\mathcal{E}_1) \ge 1 - \exp(-n_t^{\frac{1}{3mp}}).$$

We condition on $\mathcal{E}_1$ in the rest of the proof.

For $k = 1, \ldots, K$, let $\mathbf{1}_k$ be the indicator such that $\mathbf{1}_k = 1$ if $\langle \tilde{\boldsymbol{b}}_{k-1}, \boldsymbol{u}_k \rangle < 0$ and $\mathbf{1}_k = 0$ otherwise. Then,

$$\begin{aligned} 0 \le \|\tilde{\boldsymbol{b}}_K\|_2^2 &= \|\tilde{\boldsymbol{b}}_0\|_2^2 - 2\sum_{k=1}^{K} \mathbf{1}_k \cdot \left|\langle \tilde{\boldsymbol{b}}_{k-1}, \boldsymbol{u}_k \rangle\right| + \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_2^2 \\ &\le \|\tilde{\boldsymbol{b}}_0\|_2^2 - 2\sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \mathbf{1}_k \left|\langle \tilde{\boldsymbol{b}}_{k-1}, \boldsymbol{u}_k \rangle\right| + \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_2^2. \end{aligned}$$

Rearranging the above inequality and using the fact that $\boldsymbol{u}_k \in [-\frac{m\alpha'_t}{2}, \frac{m\alpha'_t}{2}]^{mp}$,

$$\sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \left|\langle \tilde{\boldsymbol{b}}_{k-1}, \boldsymbol{u}_k \rangle\right| \le \frac{1}{2}\left(\|\tilde{\boldsymbol{b}}_0\|_2^2 + \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_2^2\right) \le 4\alpha_t^2(mp + \gamma)^2 mp + \frac{1}{8}K(m\alpha'_t)^2 mp. \tag{5}$$

Let $\mathcal{E}_2$ denote the event that for all $k \le K$, $\|\tilde{\boldsymbol{b}}_k\|_2 > \gamma\alpha'_t$. For each k, let $\tilde{\boldsymbol{b}}'_k = \tilde{\boldsymbol{b}}_k / \|\tilde{\boldsymbol{b}}_k\|_2$. Assuming $\mathcal{E}_2$,

$$\sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \mathbf{1}_k \cdot \left|\langle \tilde{\boldsymbol{b}}_{k-1}, \boldsymbol{u}_k \rangle\right| = \sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \mathbf{1}_k \cdot \left|\langle \tilde{\boldsymbol{b}}'_{k-1}, \boldsymbol{u}_k \rangle\right| \cdot \|\tilde{\boldsymbol{b}}_k\|_2 > \gamma\alpha' \sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \mathbf{1}_k \cdot \left|\langle \tilde{\boldsymbol{b}}'_{k-1}, \boldsymbol{u}_k \rangle\right|.$$

Combining the above inequality and Eq. (5),

$$\sum_{\substack{1 \le k \le K \\ \boldsymbol{u}_k \in \mathcal{G}''_t}} \mathbf{1}_k \cdot \left|\langle \tilde{\boldsymbol{b}}'_{k-1}, \boldsymbol{u}_k \rangle\right| < \frac{4\alpha_t^2(mp + \gamma)^2 mp}{\gamma\alpha'_t} + \frac{K\alpha'_t m^3 p}{8\gamma} \stackrel{\text{def}}{=} \kappa. \tag{6}$$

Index the elements in $\mathcal{I}_K$ by $k_1, \ldots, k_s$. Consider sampling the vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ by first sampling the set $\mathcal{I}_K$ and then for each k, sampling $\boldsymbol{u}_k$ from $\mathcal{G}_t''$ if $k \in \mathcal{I}_K$ and from $\mathcal{B}_t''$ if $k \notin \mathcal{I}_K$. For each $1 \le j \le s$, let

$$Y_j \stackrel{\text{def}}{=} \sum_{l=1}^{j} \mathbf{1}_{k_l} \cdot \left| \langle \tilde{\boldsymbol{b}}'_{k_l - 1}, \boldsymbol{u}_{k_l} \rangle \right| - \frac{1}{2} c \alpha'_t j,$$

where $c = \frac{1}{6}$ is the constant in Claim C.7. Thus,

$$\begin{aligned} \mathbb{E}[Y_j - Y_{j-1} \mid \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k_j - 1}, \mathcal{I}_K] &= \mathbb{E}\left[\mathbf{1}_{k_j} \cdot \left| \langle \tilde{\boldsymbol{b}}'_{k_j - 1}, \boldsymbol{u}_{k_j} \rangle \right| - \frac{1}{2} c \alpha'_t \mid \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k_j - 1}, \mathcal{I}_K \right] \\ &= -\frac{1}{4} c \alpha'_t + \frac{1}{4} c \alpha'_t \ge 0. \end{aligned}$$

The last inequality follows from the symmetry of the distribution of $\boldsymbol{u}_{k_j}$ about the origin. Thus, $Y_0 = 0, Y_1, Y_2, \ldots$ form a submartingale. In addition,

$$|Y_j - Y_{j-1}| \le \left\| \boldsymbol{u}_{k_j} \right\|_2 \le \sqrt{mp} \cdot \frac{m \alpha'_t}{2}.$$

By the Azuma-Hoeffding inequality, for any $\epsilon > 0$,

$$\Pr\left(\sum_{l=1}^{|\mathcal{I}_K|} \mathbf{1}_{k_l} \cdot \left| \langle \tilde{\boldsymbol{b}}'_{k_l - 1}, \boldsymbol{u}_{k_l} \rangle \right| \le \frac{1}{2} c \alpha'_t \left| \mathcal{I}_K \right| - \epsilon \mid \mathcal{I}_K, \mathcal{E}_1 \right) \le \exp\left(- \frac{2\epsilon^2}{\left| \mathcal{I}_K \right| (\alpha'_t)^2 m^3 p} \right).$$

We take

$$\epsilon = \frac{1}{2} c \alpha'_t \left| \mathcal{I}_K \right| - \kappa = \left(\frac{1}{2} c - \frac{(1 + o(1)) m^3 p}{8\gamma} \right) \alpha'_t \left| \mathcal{I}_K \right| - \frac{4 \alpha_t^2 (mp + \gamma)^2 mp}{\gamma \alpha'_t}. \tag{7}$$

Given $\gamma = 2m^3 p$, the first term in the above right-hand side is positive. Note that

$$\frac{\alpha_t^2}{\alpha'_t} = \alpha'_t \cdot n_t^{\frac{1}{2mp}}.$$

By our choice of $K = C n_t^{\frac{1}{2mp}}$ for a sufficiently large constant C, the first term in the right-hand side of Eq. (7) dominates. Thus,

$$\Pr\left(\sum_{l=1}^{|\mathcal{I}_K|} \mathbf{1}_{k_l} \cdot \left| \langle \tilde{\boldsymbol{b}}'_{k_l - 1}, \boldsymbol{u}_{k_l} \rangle \right| \le \kappa \mid \mathcal{I}_K, \mathcal{E}_1 \right) \le \exp\left(-\Theta(|\mathcal{I}_K|) \right) = \exp\left(-\Theta(n_t^{\frac{1}{2mp}}) \right).$$

Combining all the above arguments together,

$$\Pr(\mathcal{E}_2) \le \Pr\left(\sum_{l=1}^{|\mathcal{I}_K|} \mathbf{1}_{k_l} \cdot \left| \langle \tilde{\boldsymbol{b}}'_{k_l - 1}, \boldsymbol{u}_{k_l} \rangle \right| \le \kappa \mid \mathcal{E}_1 \right) + \Pr(\overline{\mathcal{E}_1}) \le \exp\left(-\Theta(n_t^{\frac{1}{3mp}}) \right).$$

□

Claim C.7. *Let $\boldsymbol{v} \in \mathbb{R}^{mp}$ be an arbitrarily fixed vector with $\|\boldsymbol{v}\|_2 = 1$. Let $\boldsymbol{u} \in [-\frac{\alpha}{2}, \frac{\alpha}{2}]^{mp}$ be a random vector whose entries are i.i.d. random variables drawn from distribution Q_u. If Q_u is the centered IH distribution with parameters m and α, then $\mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u} \rangle|] \ge \frac{\sqrt{m}\alpha}{6}$. If Q_u is the uniform distribution over $[-\frac{\alpha}{2}, \frac{\alpha}{2}]$, then $\mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u} \rangle|] \ge \frac{\alpha}{6}$.*

Proof. By Holder's inequality (Ref: Equation 17a in Hardy et al. (1952)),

$$\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u} \rangle^2] = \mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u} \rangle|^{2/3} |\langle \boldsymbol{v}, \boldsymbol{u} \rangle|^{4/3}] \le \mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u} \rangle|]^{2/3} \cdot \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u} \rangle^4]^{1/3}.$$

Rearranging the above inequality,

$$\mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|] \geq \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^2]^{3/2} \cdot \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^4]^{-1/2}. \tag{8}$$

We expand the two expectations on the right-hand side:

$$\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^2] = \sum_{i=1}^{mp} \boldsymbol{v}(i)^2 \mathbb{E}[\boldsymbol{u}(i)^2],$$
$$\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^4] = \sum_{i=1}^{mp} \boldsymbol{v}(i)^4 \mathbb{E}[\boldsymbol{u}(i)^4] + 6 \sum_{1\leq i<j\leq mp} \boldsymbol{v}(i)^2 \boldsymbol{v}(j)^2 \mathbb{E}[\boldsymbol{u}(i)^2]\mathbb{E}[\boldsymbol{u}(j)^2].$$

Note that $\boldsymbol{u}(i)$'s are independent and have a common density $\frac{1}{\alpha} f_{IH}(\frac{y}{\alpha} + \frac{m}{2})$. The second moment of $\boldsymbol{u}(i)$:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{u}(i)^2] &= \int_{-m\alpha/2}^{m\alpha/2} y^2 \cdot \frac{1}{\alpha} f_{IH}(\frac{y}{\alpha} + \frac{m}{2}) dy \\
&= \int_0^m (x - \frac{m}{2})^2 \alpha^2 f_{IH}(x) dx && \text{(change of variable: } x = \tfrac{y}{\alpha} + \tfrac{m}{2}) \\
&= \frac{m\alpha^2}{12} && \text{(the IH distribution has variance } \tfrac{m}{12})
\end{aligned}$$

Similarly, the fourth moment of $\boldsymbol{u}(i)$:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{u}(i)^4] &= \int_{-m\alpha/2}^{m\alpha/2} y^4 \cdot \frac{1}{\alpha} f_{IH}(\frac{y}{\alpha} + \frac{m}{2}) dy \\
&= \frac{\alpha^4 m}{24}(\frac{m}{2} - \frac{1}{5}) && \text{(the IH distribution has kurtosis } \tfrac{m^2}{48} - \tfrac{m}{120})
\end{aligned}$$

Thus,

$$\mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^2] = \frac{m\alpha^2}{12}, \quad \mathbb{E}[\langle \boldsymbol{v}, \boldsymbol{u}\rangle^4] \leq \frac{m^2\alpha^4}{48}.$$

and Plugging into Equation (8),

$$\mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|] \geq \frac{\sqrt{m}\alpha}{6}.$$

By a similar argument, we can bound $\mathbb{E}[|\langle \boldsymbol{v}, \boldsymbol{u}\rangle|] \geq \frac{\alpha}{6}$ if $\boldsymbol{u}$ is uniformly distributed over $[-\frac{\alpha}{2}, \frac{\alpha}{2}]^{mp}$. □

D ANALYSIS OF GKK+ AND PROOF OF THEOREM 3.2

In this section, we prove Theorem 3.2. We prove the discrepancy bounds in Section D.1 and the robustness bound in Section D.2.

D.1 Discrepancy

We start by bounding the key parameters in each phase. W.h.p., at the beginning of phase t, the number of vectors is approximately $\left(\frac{1}{m}\right)^{t-1} n$, with only an $o(1)$ fraction labeled as bad. This guarantees that Algorithm 4 GKK+ can run for a sufficiently large number of phases.

Lemma D.1. *Assume $m \geq 2$ is fixed and $p \leq \frac{1}{m}\sqrt{\frac{\log n}{5 \log m}}$. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in [-\Delta, \Delta]^p$ be i.i.d. random vectors drawn from a continuous distribution over $[-\Delta, \Delta]^p$ with a Lipschitz density. Let $c_1 = 0.1$ and $c_2 < \frac{\log m}{2 \log n}$. Then, for any positive integer $T \leq \frac{c_1 \log n}{mp \log m}$, with probability at least $1 - n^{-9}$, the following hold for each $t \leq T$:*

$$\begin{aligned}
&(\frac{1}{m} - n^{-c_2})^{t-1} n \leq |\mathcal{G}_t| \leq (\frac{1}{m})^{t-1} n, \\
&|\mathcal{B}_t| \leq \sum_{s=1}^{t-2} (\frac{1}{m^{3/4}})^s \cdot 2n^{1 - \frac{1}{4mp}} + t - 1.
\end{aligned} \tag{9}$$

By the choice of T, the bounds above imply that for each $t \le T$,

$$|\mathcal{B}_t| = O(n^{1-\frac{1}{4mp}}) = o(1)\,|\mathcal{G}_t|\,. \tag{10}$$

Proof. We prove Eq. (9) by induction on $t \le T$. Then, Eq. (10) follows immediately.

Base case $t = 1$. For $t = 1$, $\mathcal{G}_1 = \mathcal{S}_1$, and thus $|\mathcal{G}_1| = n$ and $|\mathcal{B}_1| = 0$. The induction hypothesis in Eq. (9) holds.

Inductive step. Assume Eq. (9) holds for $t < T$. Since each cyclic differencing operation combines m vectors into one vector and all leftover vectors are combined with $\tilde{\boldsymbol{b}}$ (labeled as bad), we have

$$|\mathcal{G}_{t+1}| \le \frac{|\mathcal{G}_t|}{m}.$$

To prove the lower bound for $|\mathcal{G}_{t+1}|$ and the upper bound for $|\mathcal{B}_{t+1}|$, we track the number of good and bad vectors after each step of Algorithm 4 GKK+.

At the end of (conceptual) RESAMPLING, by Lemma C.4, with probability n^{-10}, the following holds:

$$|\mathcal{G}'_t| \ge (1 - 2n_t^{-\frac{1}{4mp}})\,|\mathcal{G}_t|\,.$$

Conditional on this event,

$$\begin{aligned}
|\mathcal{B}'_t| - |\mathcal{B}_t| &= |\mathcal{G}_t| - |\mathcal{G}'_t| \\
&= 2n_t^{-\frac{1}{4mp}}\,|\mathcal{G}_t| \\
&\le 2\,|\mathcal{G}_t|^{1-\frac{1}{4mp}} \\
&\le 2\left(\frac{1}{m}\right)^{(t-1)(1-\frac{1}{4mp})} n^{1-\frac{1}{4mp}} && \text{(by induction hypothesis)} \\
&\le 2\left(\frac{1}{m^{3/4}}\right)^{t-1} n^{1-\frac{1}{4mp}} && \text{(since } mp \ge 1\text{)}
\end{aligned}$$

Combining with the induction hypothesis for $|\mathcal{B}_t|$, we have

$$\begin{aligned}
|\mathcal{B}'_t| &\le \sum_{s=1}^{t-2}\left(\frac{1}{m^{3/4}}\right)^s \cdot 2n^{1-\frac{1}{4mp}} + t - 1 + 2\left(\frac{1}{m^{3/4}}\right)^{t-1} n^{1-\frac{1}{4mp}} \\
&\le \sum_{s=1}^{t-1}\left(\frac{1}{m^{3/4}}\right)^s \cdot 2n^{1-\frac{1}{4mp}} + t - 1 \qquad (11) \\
&= O(n^{1-\frac{1}{4mp}}).
\end{aligned}$$

At the end DIFFERENCE,

$$\begin{aligned}
|\mathcal{G}''_t| &\ge \frac{|\mathcal{G}'_t| - (m-1)N_t - (m-1)\,|\mathcal{B}'_t|}{m} \\
&\ge \frac{(1 - 2n_t^{-\frac{1}{4mp}})\,|\mathcal{G}_t| - (m-1)2^{mp}n_t^{\frac{1}{4}} - O(n^{1-\frac{1}{4mp}})}{m}.
\end{aligned}$$

By the induction hypothesis in Eq. (9),

$$\begin{aligned}
|\mathcal{G}_t| &\ge \left(\frac{1}{m} - n^{-c_2}\right)^{\frac{c_1 \log n}{mp \log m}} n = (1 - o(1))\,n^{1-\frac{c_1}{mp}}, \\
|\mathcal{G}_t| &\ge (1 - o(1))n_t.
\end{aligned}$$

Thus,

$$\begin{aligned}|\mathcal{G}_t''| &\geq \frac{|\mathcal{G}_t|}{m}\left(1-2n_t^{-\frac{1}{4mp}}-(1+o(1))(m-1)2^{mp}n_t^{-\frac{3}{4}}-(1+o(1))n^{-\frac{1-4c_1}{4mp}}\right)\\ &= \frac{|\mathcal{G}_t|}{m}\left(1-O(n^{-\frac{3}{4}(1-\frac{c_1}{4mp})}+n^{-\frac{1-4c_1}{4mp}})\right)\\ &= |\mathcal{G}_t|\left(\frac{1}{m}-n^{-c_2}\right) \qquad \text{(since } c_1+c_2mp<\tfrac{1}{4}\text{)}\end{aligned}$$

At the end of CLEAN-UP, by Theorem B.1 and Lemma C.6, with probability at least $1-n^{-10}$,

$$|\mathcal{G}_{t+1}| \geq |\mathcal{G}_t''| - C\,|\mathcal{G}_t''|^{\frac{1}{2mp}} = (1-n^{-c_2})\,|\mathcal{G}_t''|\,,$$

where C is a sufficiently large constant in Lemma C.6. By the induction hypothesis,

$$|\mathcal{G}_{t+1}| \geq (\frac{1}{m}-n^{-c_2})\,|\mathcal{G}_t| \geq (\frac{1}{m}-n^{-c_2})^t n.$$

In addition, combining Eq. (11),

$$|\mathcal{B}_{t+1}| \leq |\mathcal{B}_t''|+1 \leq \sum_{s=1}^{t-1}(\frac{1}{m^{3/4}})^s\cdot 2n^{1-\frac{1}{4mp}}+t.$$

Therefore, the induction hypothesis in Eq. (9) holds for $t+1$. □

D.1.1 ℓ_∞ Discrepancy Bound

We prove the ℓ_∞ discrepancy bound stated in Theorem 3.2. We restate the bound in the following lemma.

Lemma D.2. *Under the same assumptions as in Lemma D.1, Algorithm 4 GKK+ (together with a standard backtracking) returns a random assignment $\boldsymbol{W}\in[m]^n$ such that, with probability at least $1-n^{-8}$,*

$$\mathcal{D}_\infty(\boldsymbol{W}) = n^{-\Theta(\frac{\log n}{(mp)^2\log m})}.$$

Proof. By Lemma D.1, with probability at least $1-n^{-9}$, Algorithm 4 can run $T=\frac{c_1\log n}{mp\log m}$ phases.

For each $t\leq T$, at the end of phase t, all m-tuples except $\tilde{\boldsymbol{b}}$ are in $[-\alpha_{t+1},\alpha_{t+1}]$, and $\tilde{\boldsymbol{b}}\in[-\gamma\alpha_{t+1},\gamma\alpha_{t+1}]$ where γ is a constant that only depends on m and p. Let

$$\theta \stackrel{\text{def}}{=} \frac{1}{m}-n^{-c_2} \tag{12}$$

be the parameter in Eq. (9). By our choice of parameters in Algorithm 4,

$$\begin{aligned}&\alpha_1=\Delta,\\ &\alpha_{t+1}\leq m\alpha_t\cdot n_t^{-\frac{1}{4mp}}\leq m\alpha_t\cdot(\theta^{t-1}n)^{-\frac{1}{4mp}},\ \forall t\geq 1\end{aligned}$$

Thus,

$$\begin{aligned}\alpha_{t+1} &= \frac{\alpha_{t+1}}{\alpha_t}\times\frac{\alpha_t}{\alpha_{t-1}}\times\cdots\times\frac{\alpha_2}{\alpha_1}\times\alpha_1\\ &\leq m^t\cdot\theta^{-\frac{t^2-t}{8mp}}\cdot n^{-\frac{t}{4mp}}\cdot\Delta.\end{aligned} \tag{13}$$

Let $\tilde{\boldsymbol{u}}$ be the m-tuple returned Algorithm 4 GKK+ with input $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$. The ℓ_∞ discrepancy of the m groups by a standard backtracking equals

$$\max_{k,k'\in[m]}\left\|\tilde{\boldsymbol{u}}^{(k)}-\tilde{\boldsymbol{u}}^{(k')}\right\|_\infty$$

where, for $k \in [m]$, $\tilde{\boldsymbol{u}}^{(k)} = \tilde{\boldsymbol{u}}((k-1)p+1 : kp)$ is the kth component of $\tilde{\boldsymbol{u}}$. The above number is at most

$$2 \max_{k \in [m]} \left\| \tilde{\boldsymbol{u}}^{(k)} - \boldsymbol{w} \right\|_\infty,$$

where $\boldsymbol{w} \in \mathbb{R}^p$ is the average of the m components in $\tilde{\boldsymbol{u}}$.

At the end of Algorithm 4, we randomly permute all the m-tuples in $\mathcal{S}_{T+1} \cup \{\tilde{\boldsymbol{b}}\}$ and add them together with $\tilde{\boldsymbol{b}}$ to obtain $\tilde{\boldsymbol{u}}$. We first bound a single entry of $\tilde{\boldsymbol{u}}$ and then apply a union bound over all its entries. Consider the first entry of $\tilde{\boldsymbol{u}}$, which corresponds to the first entry of the p-dimensional vector in the first group. The same argument holds for all other entries of $\tilde{\boldsymbol{u}}$. Note that

$$\tilde{\boldsymbol{u}}(1) = \sum_{\tilde{\boldsymbol{v}} \in \mathcal{S}_{T+1} \cup \{\tilde{\boldsymbol{b}}\}} \tilde{\boldsymbol{v}}^{(\sigma_{\tilde{\boldsymbol{v}}}(1))}(1),$$

where $\sigma_{\tilde{\boldsymbol{v}}}$'s are random permutations and independent of each other. Observe that

$$\mathbb{E}[\tilde{\boldsymbol{u}}(1)] = \boldsymbol{w}(1).$$

By Hoeffding's inequality, for a sufficiently large constant C,

$$\Pr\left(|\tilde{\boldsymbol{u}}(1) - \boldsymbol{w}(1)| \geq C\alpha_{T+1}\sqrt{n_{T+1}\log n} \right) \leq n^{-10}.$$

That is, with probability at least $1 - n^{-10}$,

$$\begin{aligned}
|\tilde{\boldsymbol{u}}(1) - \boldsymbol{w}(1)| &\leq C\alpha_{T+1}\sqrt{n_{T+1}\log n} \\
&\leq Cm^T \cdot \theta^{-\frac{T^2-T}{8mp}} \cdot n^{-\frac{T}{4mp}} \cdot \sqrt{m^{-T+1} n \log n} \cdot \Delta && \text{(by Eq. (13))} \\
&\leq O(1) \cdot m^{\frac{T+1}{2} + \frac{T^2-T}{8mp}} n^{\frac{1}{2} - \frac{T}{4mp}} \log^{\frac{1}{2}} n \cdot \Delta && \text{(by Eq. (12))} \\
&\leq O(1) \cdot n^{\frac{\log m}{\log n}(\frac{T^2}{8mp} + T) + \frac{1}{2} - \frac{T}{4mp}} \log^{\frac{1}{2}} n \cdot \Delta \\
&\leq O(1) \cdot n^{\frac{1}{2} - T(\frac{1}{4mp} - \frac{c_1}{8(mp)^2} - \frac{\log m}{\log n})} \log^{\frac{1}{2}} n \cdot \Delta && \text{(since } T \leq \tfrac{c_1 \log n}{mp \log m}\text{)} \\
&\leq n^{\frac{1}{2} - \frac{T}{5mp}} \cdot \Delta && \text{(since } c_1 = 0.1 \text{ is sufficiently small)}
\end{aligned}$$

Taking a union bound over all the mp entries of $\tilde{\boldsymbol{u}}$, with probability at least $1 - mpn^{-10}$, the ℓ_∞ discrepancy of the m-group assignment is at most

$$O(1) \cdot n^{\frac{1}{2} - \frac{T}{5mp}}.$$

Taking $T = \frac{c_1 \log n}{mp \log m}$ and $\Delta = e^{o(\sqrt{\log n})}$, we have the claimed bound. □

D.1.2 Lipschitz Discrepancy Bound

We establish the Lipschitz discrepancy bound stated in Theorem 3.2.

In the first phase, the GKK+ algorithm groups all but an $o(1)$ fraction of the leftover input vectors into sets of size m, ensuring that the vectors within each group are close to one another. Within each group, the vectors are randomly assigned to m distinct arms. This extends the classical pairwise matching design from two arms to m arms, yielding a low Lipschitz discrepancy. The contribution of the ungrouped leftover vectors to the overall Lipschitz discrepancy is negligible.

Lemma D.3. *Assume $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in [-\Delta, \Delta]^p$ and GKK+ runs for at least one phase. Then, GKK+ (together with a standard backtracking) returns a random assignment $\boldsymbol{W} \in [m]^n$ such that*

$$\mathcal{D}_{\mathrm{Lip}}(\boldsymbol{W}) = O(n^{1-1/(4mp)}).$$

Proof. Let $f : \mathbb{R}^p \to \mathbb{R}$ be an arbitrary Lipschitz function. For each pair $k, k' \in [m]$ where $k \neq k'$, we define

$$M_{k,k'} \stackrel{\text{def}}{=} \left| \sum_{i:W_i=k} f(\boldsymbol{x}_i) - \sum_{i:W_i=k'} f(\boldsymbol{x}_i) \right|.$$

$M_{k,k'}$ is determined in phase 1. In phase 1, a unit is either grouped with $m-1$ others for cyclic differencing or is a leftover. If (i, j) with $W_i = k, W_j = k'$ are grouped together and cyclic differenced, their contribution to $M_{k,k'}$ is at most $O(\alpha_1')$, and there are at most n/m such pairs. A leftover unit contributes at most $|f(\mathbf{0})| + O(\alpha_1)$. Since the number of subcubes in phase 1 is

$$N_1 = (2n^{1/(4mp)})^p = 2^p n^{1/(4m)},$$

there are at most $(m-1)N_1$ leftovers. Hence,

$$M_{k,k'} \leq \frac{n}{m} \cdot \frac{\Delta}{n^{1/(4mp)}} + (m-1) \cdot 2^p n^{1/(4m)} \cdot (\Delta + O(1)) \leq n^{1-1/(4mp)} \cdot \Delta.$$

Since $\Delta = e^{o(\sqrt{\log n})}$, we have the claimed bound. □

D.2 Robustness Bound

We prove the robustness bound stated in Theorem 3.2. Intuitively, after T phases of Algorithm 4, still a large number of m-tuples are left. They are randomly permuted and combined. This ensures that a large portion of the assignment exhibits independence.

Lemma D.4. *Under the same assumptions as in Lemma D.1, Algorithm 4 GKK+ (together with a standard backtracking) returns a random assignment $\boldsymbol{W} \in [m]^n$ such that, with probability at least $1 - n^{-9}$, the following holds: for any fixed integer $l > 0$, at most $o(1)$ fraction of $\{(W_{i_1}, \ldots, W_{i_l}) : 1 \leq i_1 < \ldots < i_l \leq n\}$ are* not *mutually independent (i.e., $r_l(\boldsymbol{W}) = o(1)$).*

Proof. Algorithm 4 GKK+ runs for at most $T \leq \frac{0.1 \log n}{mp \log m}$ phases. After T phases, by Lemma D.1, with probability at least $1 - n^{-9}$, Algorithm 4 has grouped the n input vectors – excluding those that are leftovers or used in the second step of Clean-Up – into

$$g = (1 - o(1)) \cdot \frac{n}{m^T}$$

groups, each of size m^T, as established in Lemma D.1. For any fixed integer l, if units $i_1, \ldots, i_l$ belong to l distinct such groups, then their assignments are mutually independent. The fraction of such mutually independent l-tuples among all possible l-tuples is at least

$$\begin{aligned}
\frac{\binom{g}{l}(m^T)^l}{\binom{n}{l}} &= \prod_{i=0}^{l-1} \frac{(g-i)m^T}{n-i} \\
&= \prod_{i=0}^{l-1} \frac{(1-o(1))n - im^T}{n-i} \\
&= \prod_{i=0}^{l-1} \frac{1 - o(1) - im^T/n}{1 - i/n} \\
&\geq \prod_{i=0}^{l-1} \left(1 - o(1) - \frac{im^T}{n}\right) \\
&= 1 - o(1) \qquad \text{(since } T \leq \tfrac{0.1 \log n}{mp \log m}\text{)}
\end{aligned}$$

□

D.3 Runtime

We show that, with probability at least $1 - n^{-9}$, Algorithm 4 GKK+ runs in time $O(pn)$, where the hidden constants depend only on m.

In phase t, both Partition and Difference take $O(pn_t)$ time. By Theorem B.1, the first step of Clean-Up requires $O(p^3(mN_t)^2) = O(n_t)$ time, while the second step takes $O(n_t)$. Hence, each phase has runtime $O(n_t)$. Moreover, Lemma D.1 ensures that, with probability at least $1 - n^{-9}$, we have $n_{t+1} = \Theta(n_t/m)$ for all $t \geq 1$. Therefore, the runtime is dominated by the first phase and totals $O(pn)$.

E STATISTICAL PROPERTIES OF GKK+

In this section, we establish statistical properties of GKK+ for estimating the average treatment effect (ATE). We prove Theorems 3.3, 3.4, and 3.5.

E.1 Variance and Convergence

In this section, we prove Theorems 3.3 and 3.4. We first calculate the estimation error.

Claim E.1. *For any $k, k' \in [m]$, the estimation error*

$$\hat{\tau}_{kk'} - \tau_{kk'} = \frac{m}{n} \sum_{i:W_i=k} f_k(\boldsymbol{x}_i) - \bar{f}_k - \left(\frac{m}{n} \sum_{i:W_i=k'} f_{k'}(\boldsymbol{x}_i) - \bar{f}_{k'} \right) + \frac{1}{n} \sum_{i=1}^{n} (mw_{ik} - 1)\epsilon_{ik} - (mw_{ik'} - 1)\epsilon_{ik'},$$

where

$$\bar{f}_k \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_k(\boldsymbol{x}_i), \quad \forall k \in [m].$$

Proof. By definition,

$$\begin{aligned}
\hat{\tau}_{kk'} - \tau_{kk'} =& \frac{m}{n} \left(\sum_{i:W_i=k} Y_{ik} - \sum_{i:W_i=k'} Y_{ik'} \right) - \frac{1}{n} \sum_{i=1}^{n} (Y_{ik} - Y_{ik'}) \\
=& \frac{m}{n} \sum_{i:W_i=k} f_k(x_i) - \bar{f}_k - \left(\frac{m}{n} \sum_{i:W_i=k'} f_{k'}(x_i) - \bar{f}_{k'} \right) + \frac{1}{n} \sum_{i=1}^{n} (mw_{ik} - 1)\epsilon_{ik} - (mw_{ik'} - 1)\epsilon_{ik'}.
\end{aligned}$$

□

The first part of Theorem 3.3 is restated in the following claim.

Claim E.2. *For any $k, k' \in [m]$,*

$$var(\hat{\tau}_{kk'} \mid \boldsymbol{X}) = \frac{1}{n^2} \mathbb{E} \left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'} \right)^2 \mid \boldsymbol{X} \right] + \frac{m-1}{n^2} \sum_{i=1}^{n} \mathbb{E} \left[\epsilon_{ik}^2 + \epsilon_{ik'}^2 + \frac{2}{m-1} \epsilon_{ik}\epsilon_{ik'} \mid \boldsymbol{x}_i \right].$$

where

$$F_{ik} \stackrel{\text{def}}{=} (mw_{ik} - 1) f_k(x_i), \ \forall i, k$$

and

$$var(\hat{\tau}_{kk'}) = \frac{1}{n^2} \mathbb{E} \left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'} \right)^2 \right] + \frac{m-1}{n} \mathbb{E} \left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1} \epsilon_{1k}\epsilon_{1k'} \right] + var(\tau_{kk'}).$$

Proof. Since for each i and k, $\Pr(W_i = k) = 1/m$, the expectation:

$$\mathbb{E}[\hat{\tau}_{kk'}] = \frac{1}{n} \sum_{i=1}^{n} (Y_{ik} - Y_{ik'}) = \tau_{kk'}.$$

Let

$$E_{ik} = (mw_{ik} - 1)\epsilon_{ik},\ \forall i, k$$

Conditional on $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, the variance (multiplying n^2):

$$\begin{aligned}
&n^2 \cdot \mathrm{var}(\hat{\tau}_{kk'} \mid \boldsymbol{X}) \\
=&\mathbb{E}\left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'} + \sum_{i=1}^{n} E_{ik} - E_{ik'}\right)^2 \mid \boldsymbol{X}\right] \\
=&\mathbb{E}\left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'}\right)^2 \mid \boldsymbol{X}\right] + \mathbb{E}\left[\left(\sum_{i=1}^{n} E_{ik} - E_{ik'}\right)^2 \mid \boldsymbol{X}\right] + 2\mathbb{E}\left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'}\right)\left(\sum_{i=1}^{n} E_{ik} - E_{ik'}\right) \mid \boldsymbol{X}\right]
\end{aligned}$$

In the rightmost-hand side of the above equation, the third term (aka, the crossing term) can be written as

$$\begin{aligned}
&2\sum_{i,j} \mathbb{E}\left[(F_{ik} - F_{ik'})(E_{jk} - E_{jk'}) \mid \boldsymbol{X}\right] \\
=&2\sum_{i,j} \mathbb{E}\left[(F_{ik} - F_{ik'})(mw_{jk} - 1)|\boldsymbol{X}\right] \cdot \mathbb{E}[\epsilon_{jk} \mid \boldsymbol{x}_j] - \mathbb{E}\left[(F_{ik} - F_{ik'})(mw_{jk'} - 1) \mid \boldsymbol{X}\right] \cdot \mathbb{E}[\epsilon_{jk'} \mid \boldsymbol{x}_j] \\
=&0 \qquad\qquad (\text{since } \mathbb{E}[\epsilon_{ik} \mid \boldsymbol{x}_i] = 0,\ \forall i, k)
\end{aligned}$$

Similarly, the second term can be written as

$$\begin{aligned}
&\sum_{i=1}^{n} \mathbb{E}\left[(E_{ik} - E_{ik'})^2 \mid \boldsymbol{X}\right] \\
=&\sum_{i=1}^{n} \mathbb{E}\left[(mw_{ik} - 1)^2\epsilon_{ik}^2 + (mw_{ik} - 1)^2\epsilon_{ik'}^2 - 2(mw_{ik} - 1)(mw_{ik'} - 1)\epsilon_{ik}\epsilon_{ik'} \mid \boldsymbol{X}\right] \\
=&(m-1)\sum_{i=1}^{n} \mathbb{E}\left[\epsilon_{ik}^2 + \epsilon_{ik'}^2 + \frac{2}{m-1}\epsilon_{ik}\epsilon_{ik'} \mid \boldsymbol{x}_i\right].
\end{aligned}$$

So,

$$n^2 \cdot \mathrm{var}(\hat{\tau}_{kk'} \mid \boldsymbol{X}) = \mathbb{E}\left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'}\right)^2 \mid \boldsymbol{X}\right] + (m-1)\sum_{i=1}^{n} \mathbb{E}\left[\epsilon_{ik}^2 + \epsilon_{ik'}^2 + \frac{2}{m-1}\epsilon_{ik}\epsilon_{ik'} \mid \boldsymbol{x}_i\right].$$

Without conditional on $\boldsymbol{X}$,

$$\mathrm{var}(\hat{\tau}_{kk'}) = \mathrm{var}(\hat{\tau}_{kk'} - \tau_{kk'}) + \mathrm{var}(\tau_{kk'} - \tau^*_{kk'}) + \mathrm{Cov}(\hat{\tau}_{kk'} - \tau_{kk'}, \tau_{kk'} - \tau^*_{kk'}).$$

The first term:

$$\mathrm{var}(\hat{\tau}_{kk'} - \tau_{kk'}) = \frac{1}{n^2}\mathbb{E}\left[\left(\sum_{i=1}^{n} F_{ik} - F_{ik'}\right)^2\right] + \frac{m-1}{n}\mathbb{E}\left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1}\epsilon_{1k}\epsilon_{1k'}\right].$$

The second term:

$$\mathrm{var}(\tau_{kk'} - \tau^*_{kk'}) = \mathrm{var}(\tau_{kk'}).$$

The third term:

$$\begin{aligned}
\mathrm{Cov}(\hat{\tau}_{kk'} - \tau_{kk'}, \tau_{kk'} - \tau^*_{kk'}) &= \mathbb{E}[(\hat{\tau}_{kk'} - \tau_{kk'})(\tau_{kk'} - \tau^*_{kk'})] \\
&= \mathbb{E}\left[\mathbb{E}[(\hat{\tau}_{kk'} - \tau_{kk'})(\tau_{kk'} - \tau^*_{kk'})|\boldsymbol{X}]\right] \\
&= 0.
\end{aligned}$$

Combining the above three terms, we obtain the formula for the variance of $\hat{\tau}_{kk'}$ in the statement. □

Proof of Theorem 3.3. The first part follows from Claim E.2 using the formula of $B_{kl}(\boldsymbol{W}, f)$ defined in Section 2, and the second from Theorem 3.2. □

Proof of Theorem 3.4. Under the assumptions, by Theorem 3.3,

$$\begin{aligned}\operatorname{var}(\hat{\tau}_{kk'} - \tau_{kk'}) &= \frac{m-1}{n}\mathbb{E}[g(\epsilon_{ik}, \epsilon_{ik'})] = O(\frac{1}{n}),\\ \operatorname{var}(\hat{\tau}_{kk'}) &= O(\frac{1}{n}).\end{aligned}$$

By Chebyshev's inequality, $\hat{\tau}_{kk'} - \tau_{kk'} \to 0$ in probability and $\hat{\tau}_{kk'} - \tau_{kk'} = \mathcal{O}_p(n^{-1/2})$; $\hat{\tau}_{kk'} - \tau^*_{kk'} \to 0$ in probability and $\hat{\tau}_{kk'} - \tau^*_{kk'} = \mathcal{O}_p(n^{-1/2})$. □

E.2 Asymptotic Normality

In this section, we prove Theorem 3.5. The proof of the first part is given in Section E.2.1, and the second part is in Section E.2.2.

E.2.1 Asymptotic Normality of $\sqrt{n}(\hat{\tau}_{kk'} - \tau_{kk'})$

In this section, we prove the first part of Theorem 3.5:

$$\sqrt{n}(\hat{\tau}_{kk'} - \tau_{kk'}) \xrightarrow{d} \mathcal{N}(0, V) \tag{14}$$

where

$$V = (m-1)\mathbb{E}\left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1}\epsilon_{1k}\epsilon_{1k'}\right].$$

By Lemma E.1 and Theorem 3.2, with probability at least $1 - n^{-9}$,

$$\begin{aligned}\sqrt{n}(\hat{\tau}_{kk'} - \tau_{kk'}) &= o(1) + \frac{1}{\sqrt{n}}\sum_{i=1}^{n}((mw_{ik}-1)\epsilon_{ik} - (mw_{ik'}-1)\epsilon_{ik'})\\ &= o(1) + \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik} - E_{ik'}),\end{aligned}$$

where

$$E_{ik} \stackrel{\text{def}}{=} (mw_{ik}-1)\epsilon_{ik}, \quad \forall i \in [n],\ k \in [m]$$

By Slutsky's theorem (Ref: Theorem 11.4 of Gut (2006)), it suffices to show that

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik} - E_{ik'}) \xrightarrow{d} \mathcal{N}(0, V) \tag{15}$$

We first analyze the case in which ϵ_{ik} is bounded for all $i \in [n]$ and $k \in [m]$. Subsequently, we remove this assumption and prove Theorem 3.5.

Proof of the asymptotic normality of $\hat{\tau}_{kk'} - \tau_{kk'}$ assuming $|\epsilon_{ik}| \le K$ for all i, k. We will show that for any positive integer t, the expectation of the t-th power of the left-hand side of Eq. (15) converges almost surely to the t-th moment of a normal random variable with mean zero and variance V. This establishes Eq. (15).

For a positive integer $s \le t$, let

$$\mathcal{A}_{s,t} \stackrel{\text{def}}{=} \{\boldsymbol{a} = (a_1, \ldots, a_s) \in \mathbb{Z}^s_{>0} : a_1 + \cdots + a_s = t\}.$$

Then,

$$\mathbb{E}\left[\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik}-E_{ik'})\right)^t\right] = n^{-t/2}\sum_{1\le i_1,\ldots,i_t\le n}\mathbb{E}\left[\prod_{j=1}^{t}\left(E_{i_jk}-E_{i_jk'}\right)\right]$$
$$= n^{-t/2}\sum_{s=1}^{t}\sum_{1\le i_1<\ldots<i_s\le n}\sum_{\boldsymbol{a}\in\mathcal{A}_{s,t}} c_{\boldsymbol{a},t}\cdot\mathbb{E}\left[\prod_{j=1}^{s}(E_{i_jk}-E_{i_jk'})^{a_j}\right], \tag{16}$$

where $c_{\boldsymbol{a},t}=\frac{t!}{\prod_{j=1}^{s}a_j!}$. We consider three cases of s: $s<t/2$, $s>t/2$, and $s=t/2$ (the third case happens only if t is even). For each s, let

$$A_s \stackrel{\text{def}}{=} \sum_{1\le i_1<\ldots<i_s\le n}\sum_{\boldsymbol{a}\in\mathcal{A}_{s,t}} c_{\boldsymbol{a},t}\cdot\mathbb{E}\left[\prod_{j=1}^{s}(E_{i_jk}-E_{i_jk'})^{a_j}\right].$$

Case 1: $s<t/2$**.** In this case,

$$\mathbb{E}\left[\prod_{j=1}^{s}(E_{i_jk}-E_{i_jk'})^{a_j}\mid\boldsymbol{X}\right] \le \mathbb{E}\left[\prod_{j=1}^{s}2^{a_j}\left(\left|E_{i_jk}\right|^{a_j}+\left|E_{i_jk'}\right|^{a_j}\right)\mid\boldsymbol{X}\right]$$
$$\le \mathbb{E}\left[\prod_{j=1}^{s}2^{a_j}(m-1)^{a_j}\left(\left|\epsilon_{i_jk}\right|^{a_j}+\left|\epsilon_{i_jk'}\right|^{a_j}\right)\mid\boldsymbol{X}\right]$$
$$\le 2^t(m-1)^t\prod_{j=1}^{s}\mathbb{E}\left[\left|\epsilon_{i_jk}\right|^{a_j}+\left|\epsilon_{i_jk'}\right|^{a_j}\mid\boldsymbol{x}_{i_j}\right].$$

By Assumption that $|\epsilon_{ik}|\le K$ for all i,k and taking expectation over $\boldsymbol{x}_i$'s,

$$\mathbb{E}\left[\prod_{j=1}^{s}(E_{i_jk}-E_{i_jk'})^{a_j}\right]\le 2^{t+s}(m-1)^tK^t\le(4mK)^t.$$

Plugging into the definition of A_s:

$$A_s\le\binom{n}{s}C_{t,s}(4mK)^t\le n^sC_{t,s}(4mK)^t,$$

where $C_{t,s}=\sum_{\boldsymbol{a}\in\mathcal{A}_{s,t}}c_{\boldsymbol{a},t}$ is a constant that only depends on s,t. Since $s<t/2$, we have

$$n^{-t/2}A_s=o(1),$$

that is, each term with $s<t/2$ contributes at most $o(1)$ to the leftmost-hand side of Eq. (16).

Case 2: $s>t/2$**.** For every $\boldsymbol{a}\in\mathcal{A}s,t$, there exists an entry, say a_{j^*}, satisfies $a_{j^*}=1$. Under Assumptions that $|\epsilon_{ik}|\le K$ for all i,k and Assumption 2.2, and conditional on the covariates $\boldsymbol{X}$, the variables ϵ_{ik} are mutually independent and also independent of the assignment vector $\boldsymbol{W}$. Thus,

$$\mathbb{E}\left[\prod_{j=1}^{s}(E_{i_jk}-E_{i_jk'})^{a_j}\mid\boldsymbol{X}\right]$$
$$=\mathbb{E}[\epsilon_{i_{j^*}k}\mid\boldsymbol{x}_{i_{j^*}}]\cdot\mathbb{E}\left[(mw_{i_{j^*}k}-1)\prod_{j\ne j^*}(E_{i_jk}-E_{i_jk'})^{a_j}\mid\boldsymbol{X}\right]$$
$$-\mathbb{E}[\epsilon_{i_{j^*}k'}\mid\boldsymbol{x}_{i_{j^*}}]\cdot\mathbb{E}\left[(mw_{i_{j^*}k'}-1)\prod_{j\ne j^*}(E_{i_jk}-E_{i_jk'})^{a_j}\mid\boldsymbol{X}\right]$$
$$=0.$$

Thus, $A_s=0$, that is, each term with $s>t/2$ contributes 0 to the leftmost-hand side of Eq. (16).

Case 3: $s = t/2$. It suffices to consider $a_1 = \cdots = a_s = 2$; otherwise, some entry of $\boldsymbol{a}$ must be 1 and the term contributes 0 to A_s. Thus,

$$A_s = \sum_{1 \le i_1 < \ldots < i_{t/2} \le n} \frac{t!}{2^{t/2}} \mathbb{E}\left[\prod_{j=1}^{t/2} (E_{i_j k} - E_{i_j k'})^2\right].$$

Let $\mathcal{T}_{t/2} = \{(i_1, \ldots, i_{t/2}) : 1 \le i_1 < \ldots < i_{t/2} \le n\}$. Let $\mathcal{B} \subset \mathcal{T}_{t/2}$ be the subset consisting of all $(i_1, \ldots, i_{t/2})$ for which $(W_{i_1}, \ldots, W_{i_{t/2}})$ are not mutually independent. By Theorem 3.2, with probability at least $1 - n^{-9}$,

$$|\mathcal{B}| = o(1)|\mathcal{T}_{t/2}|. \tag{17}$$

For each $(i_1, \ldots, i_{t/2}) \in \mathcal{T}_{t/2} \setminus \mathcal{B}$,

$$\begin{aligned}\mathbb{E}\left[\prod_{j=1}^{t/2} (E_{i_j k} - E_{i_j k'})^2 \mid \boldsymbol{X}\right] &= \prod_{j=1}^{t/2} \mathbb{E}\left[(E_{i_j k} - E_{i_j k'})^2 \mid \boldsymbol{x}_{i_j}\right] \\ &= (m-1)^{t/2} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'} \mid \boldsymbol{x}_{i_j}\right]\end{aligned}$$

For each $(i_1, \ldots, i_{t/2}) \in \mathcal{B}$,

$$0 \le \mathbb{E}\left[\prod_{j=1}^{t/2} (E_{i_j k} - E_{i_j k'})^2 \mid \boldsymbol{X}\right] \le 2^{t/2}(m-1)^t \cdot \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 \mid \boldsymbol{x}_{i_j}\right].$$

Thus,

$$\begin{aligned}&\mathbb{E}\left[\prod_{j=1}^{t/2} (E_{i_j k} - E_{i_j k'})^2 \mid \boldsymbol{X}\right] - (m-1)^{t/2} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'} \mid \boldsymbol{x}_{i_j}\right] \\ \ge &-(m-1)^{t/2} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'} \mid \boldsymbol{x}_{i_j}\right] \\ \ge &-m^{t/2} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 \mid \boldsymbol{x}_{i_j}\right],\end{aligned}$$

and

$$\begin{aligned}&\mathbb{E}\left[\prod_{j=1}^{t/2} (E_{i_j k} - E_{i_j k'})^2 \mid \boldsymbol{X}\right] - (m-1)^{t/2} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'} \mid \boldsymbol{x}_{i_j}\right] \\ \le &\left(2^{t/2}(m-1)^t - (m-2)^{t/2}\right) \cdot \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 \mid \boldsymbol{x}_{i_j}\right].\end{aligned}$$

Therefore, conditional on $\boldsymbol{X}$,

$$\begin{aligned}&\frac{2^{t/2}}{t!} \cdot A_s - (m-1)^{t/2} \sum_{(i_1, \ldots, i_{t/2}) \in \mathcal{T}_{t/2}} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'} \mid \boldsymbol{x}_{i_j}\right] \\ \in &\left[-m^{t/2}, 2^{t/2}(m-1)^t - (m-2)^{t/2}\right] \sum_{(i_1, \ldots, i_{t/2}) \in \mathcal{B}} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 \mid \boldsymbol{x}_{i_j}\right].\end{aligned}$$

By Assumption that $|\epsilon_{ik}| \le K$ for all i, k and Eq. (17), taking the expectation over $\boldsymbol{X}$, we have

$$\frac{2^{t/2}}{t!} \cdot A_s = (m-1)^{t/2} \sum_{(i_1,\ldots,i_{t/2}) \in \mathcal{T}_{t/2}} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'}\right] + o(n^{t/2}).$$

Thus, the contribution of each term with $s = t/2$ to the leftmost-hand side of Eq. (16) is

$$\begin{aligned}
n^{-t/2} A_{t/2} &= \frac{t!}{2^{t/2}} \cdot n^{-t/2} \cdot (m-1)^{t/2} \sum_{(i_1,\ldots,i_{t/2}) \in \mathcal{T}_{t/2}} \prod_{j=1}^{t/2} \mathbb{E}\left[\epsilon_{i_j k}^2 + \epsilon_{i_j k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'}\right] + o(1) \\
&= \frac{t!}{2^{t/2}} \cdot n^{-t/2} \cdot \binom{n}{t/2} \cdot (m-1)^{t/2} \mathbb{E}\left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'}\right]^{t/2} + o(1) \\
&= (t-1)!! \cdot (m-1)^{t/2} \mathbb{E}\left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1}\epsilon_{i_j k}\epsilon_{i_j k'}\right]^{t/2} + o(1).
\end{aligned}$$

Combining all the three cases of s and applying to Eq. (16),

$$\mathbb{E}\left[\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik} - E_{ik'})\right)^t\right] = (t-1)!! \cdot V^{t/2} + o(1).$$

Therefore, Eq. (15) is satisfied, which completes the proof of the statement. □

Next, we drop the assumption that $|\epsilon_{ik}| \le K$ for all i, k.

Proof of the asymptotic normality of $\hat{\tau}_{kk'} - \tau_{kk'}$ (Proof of Theorem 3.5). Note that E_{ik} is either $(m-1)\epsilon_{ik}$ or $-\epsilon_{ik}$. Thus, bounded ϵ_{ik} implies bounded E_{ik}.

For $M > 0$ and any $i \in [n],\ k \in [m]$, let

$$E_{\le M,ik} \stackrel{\text{def}}{=} E_{ik}\mathbb{1}\left\{|E_{ik}| \le M\right\}, \quad E_{>M,ik} \stackrel{\text{def}}{=} E_{ik}\mathbb{1}\left\{|E_{ik}| > M\right\}.$$

Meanwhile, let

$$\tilde{E}_{\le M,ik} \stackrel{\text{def}}{=} E_{\le M,ik} - \mathbb{E}\left[E_{\le M,ik} \mid \boldsymbol{X}\right], \quad \tilde{E}_{>M,ik} \stackrel{\text{def}}{=} E_{>M,ik} - \mathbb{E}\left[E_{>M,ik} \mid \boldsymbol{X}\right].$$

We have:

$$\begin{aligned}
&\mathbb{E}\left[\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})\right|^2 \mid \boldsymbol{X}\right] \\
=&\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right] + \frac{1}{n}\sum_{i\ne j}\mathbb{E}\left[(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})(\tilde{E}_{>M,jk} - \tilde{E}_{>M,jk'}) \mid \boldsymbol{X}\right] \\
=&\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right] + \frac{1}{n}\sum_{i\ne j}\Big(\mathbb{E}\left[\tilde{E}_{>M,ik}\tilde{E}_{>M,jk} \mid \boldsymbol{X}\right] \\
&+\mathbb{E}\left[\tilde{E}_{>M,ik'}\tilde{E}_{>M,jk'} \mid \boldsymbol{X}\right] - \mathbb{E}\left[\tilde{E}_{>M,ik}\tilde{E}_{>M,jk'} \mid \boldsymbol{X}\right] - \mathbb{E}\left[\tilde{E}_{>M,ik'}\tilde{E}_{>M,jk} \mid \boldsymbol{X}\right]\Big) \\
=&\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right] \\
&+\frac{1}{n}\sum_{i\ne j}\Big(\mathbb{E}\left[\tilde{E}_{>M,ik} \mid \boldsymbol{X}\right]\mathbb{E}\left[\tilde{E}_{>M,jk} \mid \boldsymbol{X}\right] + \mathbb{E}\left[\tilde{E}_{>M,ik'} \mid \boldsymbol{X}\right]\mathbb{E}\left[\tilde{E}_{>M,jk'} \mid \boldsymbol{X}\right] \\
&\qquad -\mathbb{E}\left[\tilde{E}_{>M,ik} \mid \boldsymbol{X}\right]\mathbb{E}\left[\tilde{E}_{>M,jk'} \mid \boldsymbol{X}\right] - \mathbb{E}\left[\tilde{E}_{>M,ik'} \mid \boldsymbol{X}\right]\mathbb{E}\left[\tilde{E}_{>M,jk} \mid \boldsymbol{X}\right]\Big) \qquad \text{(by independence)} \\
=&\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik} - \tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right]. \qquad (18)
\end{aligned}$$

By the Strong Law of Large Numbers, $\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik}-\tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right]$ converges almost surely to $\mathbb{E}\left[(\tilde{E}_{>M,1k}-\tilde{E}_{>M,1k'})^2\right]$. By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}\limsup_n \mathbb{E}\left[\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(\tilde{E}_{>M,ik}-\tilde{E}_{>M,ik'})\right| \mid \boldsymbol{X}\right] &\le \limsup_n \mathbb{E}\left[\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(\tilde{E}_{>M,ik}-\tilde{E}_{>M,ik'})\right|^2 \mid \boldsymbol{X}\right]^{1/2}\\ &= \limsup_n \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[(\tilde{E}_{>M,ik}-\tilde{E}_{>M,ik'})^2 \mid \boldsymbol{X}\right]\right)^{1/2} \quad \text{(by Eq. (18))}\\ &= \mathbb{E}\left[(\tilde{E}_{>M,1k}-\tilde{E}_{>M,1k'})^2\right]^{1/2}. \qquad (19)\end{aligned}$$

Let h be a bounded Lipschitz function of constant c_h, that is, $|h(x)-h(x')|\le c_h|x-x'|$ for all $x,x'\in\mathbb{R}$. Let

$$V_{\le M} \stackrel{\text{def}}{=} \mathbb{E}\left[(\tilde{E}_{\le M,1k}-\tilde{E}_{\le M,1k'})^2\right], \quad N\sim\mathcal{N}(0,1).$$

We have:

$$\begin{aligned}&\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik}-E_{ik'})\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(V^{1/2}N\right)\right]\right|\\ \le&\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik}-E_{ik'})\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(\tilde{E}_{\le M,1k}-\tilde{E}_{\le M,1k'}\right)\right) \mid \boldsymbol{X}\right]\right|\\ &+\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(\tilde{E}_{\le M,1k}-\tilde{E}_{\le M,1k'}\right)\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(V_{\le M}^{1/2}N\right)\right]\right|\\ &+\left|\mathbb{E}\left[h\left(V_{\le M}^{1/2}N\right)\right]-\mathbb{E}\left[h\left(V^{1/2}N\right)\right]\right|\\ \le& c_h\,\mathbb{E}\left[\left|\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(\tilde{E}_{>M,ik}-\tilde{E}_{>M,ik'})\right| \mid \boldsymbol{X}\right]\\ &+\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(\tilde{E}_{\le M,1k}-\tilde{E}_{\le M,1k'}\right)\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(V_{\le M}^{1/2}N\right)\right]\right|\\ &+c_h\left|V_{\le M}^{1/2}-V^{1/2}\right|\mathbb{E}[|N|].\end{aligned}$$

By the first part of the proof for bounded E_{ik} and Eq. (19), we have

$$\begin{aligned}&\limsup_n\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik}-E_{ik'})\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(V^{1/2}N\right)\right]\right|\\ &\le c_h\,\mathbb{E}\left[(\tilde{E}_{>M,1k}-\tilde{E}_{>M,1k'})^2\right]^{1/2}+0+c_h\left|V_{\le M}^{1/2}-V^{1/2}\right|\mathbb{E}[|N|].\end{aligned}$$

Under Assumption $\mathbb{E}[\epsilon_{ik}^2 \mid \boldsymbol{x}_i]<\infty$, the right-hand side in the above equation goes to 0 as $M\to\infty$. Thus,

$$\limsup_n\left|\mathbb{E}\left[h\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(E_{ik}-E_{ik'})\right) \mid \boldsymbol{X}\right]-\mathbb{E}\left[h\left(V^{1/2}N\right)\right]\right|=0,$$

and Eq. (15) is satisfied (Ref: Theorem 11.3.3 of Dudley (2018)). □

E.2.2 Asymptotic Normality of $\sqrt{n}(\hat{\tau}_{kk'}-\tau^*_{kk'})$

In this section, we prove the second part of Theorem 3.5:

$$\sqrt{n}(\hat{\tau}_{kk'}-\tau^*_{kk'}) \xrightarrow{d} \mathcal{N}(0,V^*)$$

where

$$V^* = (m-1)\left[\epsilon_{1k}^2 + \epsilon_{1k'}^2 + \frac{2}{m-1}\epsilon_{1k}\epsilon_{1k'}\right] + n \cdot \mathrm{var}(\tau_{kk'}).$$

Proof. We write

$$\sqrt{n}(\hat{\tau}_{kk'} - \tau^*_{kk'}) = \sqrt{n}(\hat{\tau}_{kk'} - \tau_{kk'}) + \sqrt{n}(\tau_{kk'} - \tau^*_{kk'}).$$

By Eq. (14) and Slutsky's theorem, it suffices to show that

$$\sqrt{n}(\tau_{kk'} - \tau^*_{kk'}) \xrightarrow{d} \mathcal{N}(0,\ n \cdot \mathrm{var}(\tau_{kk'})). \tag{20}$$

By definition,

$$\sqrt{n}(\tau_{kk'} - \tau^*_{kk'}) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \left(Y_{ik} - Y_{ik'} - \tau^*_{kk'}\right).$$

Note that $Y_{ik} - Y_{ik'} - \tau^*_{kk'}$'s are i.i.d. random variables with expectation 0 and bounded variance. By the central limit theorem (Lindeberg-Lévy CLT) and the fact that $n \cdot \mathrm{var}(\tau_{kk'}) = \mathrm{var}(Y_{1k} - Y_{1k'})$, Eq. (20) holds. □

F ADDITIONAL EXPERIMENTAL DETAILS AND RESULTS

F.1 ℓ_∞ Discrepancy

In this section, we present our experimental results for $m = 2$ and for $m = 4$, $p = 20$.

Two arms $m = 2$. For the two-arm case ($m = 2$), we compare GKK+ with Bernoulli, Complete Randomization (CR), Rerandomization (Rerand), Blocking (Block), Quick Block, and an additional design known as Greedy Pair-Switching (GreedySwitch) in Krieger et al. (2019). GreedySwitch starts with a random assignment generated under Complete Randomization and then iteratively performs greedy swaps of unit pairs between arms, chosen from all n^2 possible pairs, to reduce the discrepancy. This process continues until no further improvement is possible. GreedySwitch has a time complexity of $\Omega(n^m)$, making it computationally infeasible for $m > 2$. Hence, we exclude it from comparisons involving more than two arms. In addition, because of the slow runtime of GreedySwitch, we evaluate all designs using 1,000 independent samples for $m = 2$. Our simulation results for $m = 2$ are shown in Figure 3.

When $m = 2$, Bernoulli and CR yield the highest discrepancies, while Block, Quick Block, and Rerand perform moderately better. GKK+ and GreedySwitch achieve the lowest discrepancies. Focusing on GKK+ and GreedySwitch, we observe that their ℓ_∞ discrepancies decrease rapidly as n increases, with GKK+ consistently achieving smaller discrepancies than GreedySwitch.

Figure 3: ℓ_∞ Discrepancy $\mathcal{D}_\infty$ for two arms.

Four arms $m = 4$ and covariate dimension $p = 20$. For $p = 20$ covariates, we compare the GKK+ heuristic, which performs recursive m-way matching, against common benchmark methods. We set the number of recursion steps to $T = 2$. The simulation results are shown in Figure 4. The GKK+ heuristic achieves the lowest discrepancy $\mathcal{D}_\infty$ across all tested values of n.

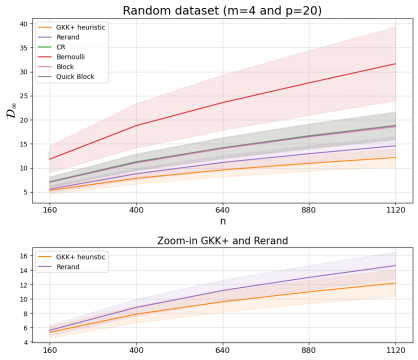

Figure 4: ℓ_∞ Discrepancy $\mathcal{D}_\infty$ for four arms and $p = 20$ covariates.

F.2 Variance of $\hat{\tau}_{kk'}$

F.2.1 Outcome Generating Process for Random Dataset

We provide additional details about the potential outcomes for randomly generated covariates.

$p = 4$ covariates. For each unit $i \in [n]$ and arm $k \in [m]$, the outcome is given by

$$Y_{ik} = f(\boldsymbol{x}_i) + \epsilon_{ik}, \quad \text{where } \epsilon_{ik} \sim \mathcal{N}(0, \sigma^2). \tag{21}$$

By default, we set $\sigma = 0.01$. The value of σ controls the extent to which the outcomes are uncorrelated with the covariates. We also consider larger noise levels with $\sigma = 0.1$ and $\sigma = 1$.

We consider four choices of the function f, each depending only on the third and fourth covariates, following Kallus (2018).

1. Linear: $f(\boldsymbol{x}) = \boldsymbol{x}(3) - \boldsymbol{x}(4)$;
2. Quadratic: $f(\boldsymbol{x}) = (\boldsymbol{x}(3) - \boldsymbol{x}(4))^2$;
3. Cubic: $f(\boldsymbol{x}) = (\boldsymbol{x}(3) - \boldsymbol{x}(4))^3$;
4. Sinusoidal: $f(\boldsymbol{x}) = \sin(\pi/3 + \pi\boldsymbol{x}(3)/3 - 2\pi\boldsymbol{x}(4)/3) - 6\sin(\pi\boldsymbol{x}(3)/3 + \pi\boldsymbol{x}(4)/4) + 6\sin(\pi\boldsymbol{x}(3)/3 + \pi\boldsymbol{x}(4)/6)$.

We define $f(\boldsymbol{x})$ to depend only on the third and fourth covariates, so that Blocking based on the first two covariates cannot effectively reduce the variance. In contrast, Quick Block utilizes all covariates and therefore does not suffer from this limitation.

$p = 20$ covariates. In the 20-dimensional covariate setting, we customize the outcome function f to depend on 10 covariates with indices from 6 to 15. Let $\tilde{\boldsymbol{x}} = (\boldsymbol{x}(6), \ldots, \boldsymbol{x}(15))^\top$. We define a linear combination $L(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}}^\top \boldsymbol{\beta}$, where $\boldsymbol{\beta} = (0.60, -0.50, 0.40, -0.30, 0.25, 0.20, -0.15, 0.10, 0.08, -0.05)^\top$. The weights $\boldsymbol{\beta}$ are chosen arbitrarily. Then, we define the four functions:

1. Linear: $f(\mathbf{x}) = L(\tilde{\boldsymbol{x}})$;
2. Quadratic: $f(\mathbf{x}) = L(\tilde{\boldsymbol{x}})^2$;
3. Cubic: $f(\mathbf{x}) = L(\tilde{\boldsymbol{x}})^3$;

4. Sinusoidal: $f(\mathbf{x}) = \sin(t_1) + 0.8\sin(t_2)$, where

$$t_1 = \frac{\pi}{3}\left(\boldsymbol{x}(6) - 2\boldsymbol{x}(7) + 0.5\boldsymbol{x}(8) - \tfrac{1}{3}\boldsymbol{x}(9) + \boldsymbol{x}(10) - 0.5\boldsymbol{x}(11) + \tfrac{1}{3}\boldsymbol{x}(12) - 0.25\boldsymbol{x}(13) + 0.2\boldsymbol{x}(14) - 0.15\boldsymbol{x}(15)\right),$$
$$t_2 = \pi\left(\tfrac{1}{3}\boldsymbol{x}(6) + \tfrac{1}{4}\boldsymbol{x}(7) - \tfrac{1}{5}\boldsymbol{x}(8) + \tfrac{1}{6}\boldsymbol{x}(9) - \tfrac{1}{7}\boldsymbol{x}(10) + \tfrac{1}{8}\boldsymbol{x}(11)\right).$$

F.2.2 Experimental Results

Variance var$(\hat{\tau}_{kk'} - \tau^*_{kk'})$ for four arms. We evaluate the sample variance of $\hat{\tau}_{kk'} - \tau_{kk'}$ and $\hat{\tau}_{kk'} - \tau^*_{kk'}$. Table 1 in Section 5 reports var$(\hat{\tau}_{12} - \tau_{12})$ for $m = 4$ and $p = 4$, while Table 4 in this section presents var$(\hat{\tau}_{12} - \tau^*_{12})$ for the same setting. The two tables are nearly identical because most sample ATEs are very close to the population ATE. GKK+ consistently achieves the lowest variance across all scenarios, with one exception: under cubic outcomes with small n, Rerandomization slightly outperforms GKK+. For linear outcomes, the variances under Bernoulli, Complete Randomization (CR), Blocking, and Quick Block are approximately 4 to 41 times larger than that of GKK+, and Rerandomization varies from 1.2 to 10 times. For quadratic outcomes, all benchmark methods exhibit variances between 1.4 and 4.1 times that of GKK+. For cubic outcomes, Rerandomization remains within 1.6 to 2.6 times for slightly large n, while Blocking, Quick Block, and simple randomization methods are 1.8 - 5.7 times. Under sinusoidal outcomes, Bernoulli, CR, and Blocking exceed a factor of 4 to 18, Quick Block is slightly better, and Rerandomization exceeds 1.3 to 5.5. These results indicate that GKK+ maintains the lowest variance across a range of outcome complexities and is robust to increasing nonlinearity in the outcome function.

	Linear outcomes						Quadratic outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	1.13	3.64	4.18	4.39	4.22	1.00	1.45	1.42	1.39	1.44	2.47
320	1.00	1.15	3.47	4.10	4.01	4.03	1.00	1.57	1.57	1.54	1.55	2.70
480	1.00	3.17	9.71	11.41	11.78	11.88	1.00	2.11	2.10	2.09	2.13	3.63
800	1.00	2.85	9.01	10.63	11.12	10.65	1.00	2.36	2.41	2.34	2.26	3.92
1440	1.00	5.84	18.64	22.18	22.63	21.43	1.00	2.32	2.39	2.36	2.29	3.87
2560	1.00	10.42	31.94	39.45	37.98	40.10	1.00	2.17	2.12	2.16	2.16	3.65
	Cubic outcomes						Sinusoidal outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	0.93	1.83	2.03	2.10	2.01	1.00	1.34	3.57	4.15	4.31	4.13
320	1.00	0.93	1.81	1.95	1.98	1.96	1.00	1.37	3.45	4.08	4.08	3.97
480	1.00	1.73	3.29	3.74	3.82	3.77	1.00	3.18	8.16	9.44	9.61	9.69
800	1.00	1.75	3.37	3.69	3.81	3.78	1.00	2.96	7.94	9.17	9.59	9.31
1440	1.00	2.47	4.83	5.37	5.48	5.17	1.00	4.51	12.39	14.65	15.18	14.12
2560	1.00	2.83	5.28	6.25	5.99	6.24	1.00	5.38	13.79	16.87	16.49	17.03

Table 4: Variance var$(\hat{\tau}_{1,2} - \tau^*_{1,2})$ ratios relative to GKK+ for a random dataset with $m = 4$ and $p = 4$, where all variances are scaled so that GKK+ column always has 1.00.

Different noise levels σ in the outcome generating process in Eq. (21). We evaluate the effect of different values of σ in Eq. (21), where larger σ corresponds to higher noise levels. Tables 5 and 6 report the variance results for $\sigma = 0.1$ and $\sigma = 1$, respectively.

For the low-noise setting ($\sigma = 0.1$), GKK+ consistently outperforms common benchmark designs. An exception occurs for small n under cubic outcomes, where Rerandomization performs slightly better but GKK+ remains comparable. In the high-noise setting ($\sigma = 1$), the noise term ϵ_{ik} in Eq. (21) dominates the signal term $f(\boldsymbol{x}_i)$ due to the covariate distribution. In this case, GKK+ performs comparably to the benchmark designs, indicating that covariate balancing in GKK+ does not harm performance.

By comparing the variance results across different noise levels $\sigma = 0.01, 0.1$, and 1 (Tables 1, 5, and 6), we observe that GKK+ achieves greater variance reduction when the noise level is smaller.

	Linear outcomes						Quadratic outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	1.11	3.13	3.57	3.77	3.60	1.00	1.33	1.27	1.25	1.29	2.04
320	1.00	1.12	3.00	3.48	3.39	3.41	1.00	1.43	1.40	1.40	1.44	2.29
480	1.00	2.29	6.05	7.06	7.31	7.35	1.00	1.69	1.71	1.67	1.74	2.67
800	1.00	2.09	5.90	6.87	7.28	6.82	1.00	1.83	1.86	1.84	1.79	2.76
1440	1.00	3.07	8.64	10.27	10.44	9.82	1.00	1.82	1.86	1.85	1.85	2.76
2560	1.00	3.82	10.42	12.79	12.13	12.93	1.00	1.78	1.72	1.80	1.76	2.78
	Cubic outcomes						Sinusoidal outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	0.95	1.53	1.67	1.69	1.65	1.00	1.31	3.37	3.90	4.06	3.88
320	1.00	0.94	1.48	1.58	1.63	1.61	1.00	1.34	3.31	3.88	3.90	3.77
480	1.00	1.35	2.12	2.37	2.42	2.37	1.00	2.86	7.11	8.27	8.40	8.42
800	1.00	1.35	2.12	2.23	2.32	2.31	1.00	2.75	7.08	8.12	8.52	8.27
1440	1.00	1.59	2.52	2.71	2.86	2.67	1.00	3.85	10.26	12.13	12.53	11.67
2560	1.00	1.66	2.52	2.94	2.72	2.87	1.00	4.40	10.82	13.17	12.85	13.25

Table 5: Noise level $\sigma = 0.1$. Variance $\text{var}(\hat{\tau}_{1,2} - \tau_{1,2})$ ratios relative to GKK+ for a random dataset with $m = 4$ and $p = 4$, where all variances are scaled so that GKK+ column always has 1.00.

	Linear outcomes						Quadratic outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	1.00	1.11	1.12	1.17	1.12	1.00	0.98	0.98	0.96	0.96	1.00
320	1.00	1.01	1.09	1.05	1.04	1.08	1.00	1.03	1.03	1.01	1.02	1.09
480	1.00	1.05	1.11	1.13	1.16	1.14	1.00	0.95	1.05	1.02	1.00	0.99
800	1.00	1.02	1.11	1.15	1.19	1.12	1.00	1.03	1.03	1.04	1.04	1.03
1440	1.00	1.02	1.11	1.15	1.18	1.13	1.00	1.03	1.02	1.01	1.01	1.03
2560	1.00	1.02	1.15	1.18	1.14	1.17	1.00	1.03	1.00	1.03	1.04	1.03
	Cubic outcomes						Sinusoidal outcomes					
n	GKK+	Rerand	Quick Block	Block	CR	Bernoulli	GKK+	Rerand	Quick Block	Block	CR	Bernoulli
160	1.00	1.00	1.02	1.02	1.01	1.01	1.00	1.05	1.30	1.37	1.41	1.36
320	1.00	0.99	0.95	1.01	1.02	1.05	1.00	1.01	1.29	1.38	1.40	1.34
480	1.00	1.03	1.07	1.08	1.09	1.07	1.00	1.17	1.40	1.54	1.55	1.54
800	1.00	0.97	0.96	1.02	1.00	1.00	1.00	1.16	1.44	1.49	1.54	1.50
1440	1.00	1.02	1.01	0.98	1.03	1.01	1.00	1.12	1.47	1.60	1.61	1.56
2560	1.00	1.00	1.01	1.01	0.96	1.01	1.00	1.19	1.45	1.59	1.52	1.53

Table 6: Noise level $\sigma = 1$. Variance $\text{var}(\hat{\tau}_{1,2} - \tau_{1,2})$ ratios relative to GKK+ for a random dataset with $m = 4$ and $p = 4$, where all variances are scaled so that GKK+ column always has 1.00.

Two arms. Tables 7 and 8 present results for the two-arm setting with six random covariates, and Table 9 presents the two-arm setting for the LaLonde dataset. For linear outcomes, Bernoulli, Complete Randomization (CR), and Blocking yield variances that are hundreds to thousands of times larger than that of GKK+; Rerandomization performs better, and GreedySwitch achieves variances only slightly higher than GKK+, typically between 1.1 and 2 times. Under quadratic outcomes, all methods range from approximately 1.7 to 6 times the variance of GKK+. In the cubic case, Rerandomization maintains variance between 1.5 and 3.2 times GKK+, GreedySwitch between 1.1 and 2.5, while Bernoulli, CR, and Blocking exceed 4 to 8 times. For sinusoidal outcomes, simple randomization methods reach variances between 20 and 56 times that of GKK+, Rerandomization limits this to 4 to 12, and GreedySwitch remains closest to GKK+ at 1.6 to 4.5 times. The same relative ordering holds for τ^*_{12}, with slightly lower ratios but consistent patterns. On the LaLonde dataset, GKK+ slightly outperforms GreedySwitch, followed by Rerandomization. Bernoulli, CR, and Blocking show variances 1.8 to 2.2 times higher than GKK+. Overall, GKK+ provides the most consistent and substantial variance reduction across all types of outcomes.

	Linear outcomes						Quadratic outcomes					
n	GKK+	GS	Rerand	Block	CR	Bernoulli	GKK+	GS	Rerand	Block	CR	Bernoulli
80	1.00	1.14	28.18	211.03	214.38	204.40	1.00	1.73	1.72	1.75	1.73	2.90
160	1.00	1.69	124.96	956.03	955.39	955.46	1.00	1.96	2.04	1.98	1.96	3.43
240	1.00	2.01	253.31	1934.45	1899.91	1874.76	1.00	2.31	2.24	2.33	2.27	3.85
400	1.00	1.79	395.07	2997.75	2892.08	3073.06	1.00	2.50	2.54	2.58	2.55	4.37
720	1.00	1.35	423.09	3238.09	3219.97	3173.23	1.00	3.31	3.24	3.07	3.10	5.20
1280	1.00	1.14	476.32	3228.11	3363.23	3418.58	1.00	3.96	3.84	3.98	3.82	6.48
	Cubic outcomes						Sinusoidal outcomes					
n	GKK+	GS	Rerand	Block	CR	Bernoulli	GKK+	GS	Rerand	Block	CR	Bernoulli
80	1.00	1.14	1.55	4.41	4.37	4.23	1.00	1.60	4.04	20.38	20.54	19.98
160	1.00	1.27	1.68	4.92	4.98	4.93	1.00	1.94	4.90	24.57	24.64	25.19
240	1.00	1.43	1.94	5.51	5.45	5.38	1.00	2.51	6.31	33.11	31.78	31.17
400	1.00	1.72	2.22	6.23	6.11	6.44	1.00	2.70	6.83	35.19	34.54	35.74
720	1.00	2.00	2.47	6.84	6.86	6.92	1.00	3.55	8.69	44.04	43.62	42.51
1280	1.00	2.53	3.16	8.15	8.34	8.43	1.00	4.45	11.76	53.83	56.48	56.50

Table 7: Variance $\text{var}(\hat{\tau}_{1,2} - \tau_{1,2})$ ratios relative to GKK+ for a random dataset with $m = 2$ and $p = 6$, where all variances are scaled so that the GKK+ column equals 1.00. GS: GreedySwitch.

	Linear outcomes						Quadratic outcomes					
n	GKK+	GS	Rerand	Block	CR	Bernoulli	GKK+	GS	Rerand	Block	CR	Bernoulli
80	1.00	1.15	26.75	199.68	202.89	193.34	1.00	1.73	1.72	1.75	1.73	2.90
160	1.00	1.56	99.72	760.44	759.63	759.40	1.00	1.96	2.03	1.98	1.96	3.41
240	1.00	1.70	165.00	1258.40	1237.03	1219.31	1.00	2.30	2.23	2.32	2.26	3.84
400	1.00	1.40	210.29	1591.07	1535.94	1632.14	1.00	2.49	2.53	2.58	2.54	4.35
720	1.00	1.20	220.15	1681.36	1671.52	1647.82	1.00	3.31	3.23	3.07	3.10	5.20
1280	1.00	1.09	242.02	1638.23	1705.57	1732.14	1.00	3.94	3.82	3.96	3.81	6.43
	Cubic outcomes						Sinusoidal outcomes					
n	GKK+	GS	Rerand	Block	CR	Bernoulli	GKK+	GS	Rerand	Block	CR	Bernoulli
80	1.00	1.14	1.55	4.39	4.35	4.19	1.00	1.60	4.03	20.32	20.49	19.93
160	1.00	1.27	1.68	4.89	4.96	4.92	1.00	1.94	4.89	24.54	24.60	25.14
240	1.00	1.42	1.93	5.48	5.41	5.34	1.00	2.50	6.30	33.06	31.75	31.13
400	1.00	1.71	2.20	6.16	6.05	6.37	1.00	2.69	6.81	35.09	34.46	35.65
720	1.00	1.98	2.46	6.76	6.78	6.83	1.00	3.53	8.66	43.90	43.49	42.39
1280	1.00	2.52	3.14	8.07	8.26	8.35	1.00	4.42	11.69	53.46	56.08	56.14

Table 8: Variance $\text{var}(\hat{\tau}_{1,2} - \tau^*_{1,2})$ ratios relative to GKK+ for a random dataset with $m = 2$ and $p = 6$, where all variances are scaled so that the GKK+ column equals 1.00. GS: GreedySwitch.

n	GKK+	GS	Rerand	Quick Block	Block	CR	Bernoulli
80	1.00	0.99	1.05	1.84	2.09	1.93	1.72
160	1.00	0.99	1.09	1.96	1.90	2.01	1.85
240	1.00	1.08	1.11	1.83	1.78	1.86	1.99
400	1.00	1.02	1.07	1.80	2.00	2.05	1.93
720	1.00	1.08	1.12	1.87	1.86	1.89	2.17
1280	1.00	1.07	1.18	2.07	2.05	2.10	1.98
2280	1.00	1.14	1.20	2.00	2.02	2.06	2.15

Table 9: Variance $\text{var}(\hat{\tau}_{1,2} - \tau_{1,2}) = \text{var}(\hat{\tau}_{1,2} - \tau^*_{1,2})$ ratios relative to GKK+ for the LaLonde dataset with $m = 2$ and $p = 4$, where all variances are scaled so that GKK+ column always has 1.00. GS: GreedySwitch.