FAST RERANDOMIZATION FOR BALANCING COVARIATE IN RANDOMIZED EXPERIMENTS: A METROPOLIS–HASTINGS FRAMEWORK

Anonymous authors
Paper under double-blind review

ABSTRACT

Balancing covariates is critical for credible and efficient randomized experiments. Rerandomization addresses this by repeatedly generating treatment assignments until covariate balance meets a prespecified threshold. By shrinking this threshold, it can achieve arbitrarily strong balance, with established results guaranteeing optimal estimation and valid inference in both finite-sample and asymptotic settings across diverse complex experimental settings. Despite its rigorous theoretical foundations, practical use is limited by the extreme inefficiency of rejection sampling, which becomes prohibitively slow under small thresholds and often forces practitioners to adopt suboptimal settings, leading to degraded performance. Existing work focusing on acceleration typically fail to maintain the uniformity over the acceptable assignment space, thus losing the theoretical grounds of classical rerandomization. Building upon a Metropolis-Hastings framework, we address this challenge by introducing an additional sampling-importance resampling step, which restores uniformity and preserves statistical guarantees. Our proposed algorithm, PSRSRR, achieves speedups ranging from 10 to 10,000 times while maintaining exact and asymptotic validity, as demonstrated by simulations and two real-data applications.

1 INTRODUCTION

Randomized experiments are the gold standard for credible causal inference as random assignment balances both observed and unobserved confounders in expectation. However, in practice, even under complete randomization, there remains a nontrivial risk of covariate imbalance (Rosenberger & Sverdlov, 2008), which grows as the number of covariates increases (Krieger et al., 2019; Morgan & Rubin, 2012). Such imbalance can reduce credibility due to accidental bias. While deterministic allocation can enforce near-exact covariate balance, it introduces its own problems, including selection bias, a loss of robustness, and the inability to use randomization-based inference (Harshaw et al., 2024).

An intuitive and practical approach to achieve what Kapelner et al. (2021) describe as "a harmony of optimal deterministic design and completely randomized design" is to randomize repeatedly until an assignment with appropriate and satisfactory covariate balance is achieved, a procedure known as rerandomization. Despite its long history and widespread use (Student, 1938; Cox, 1982; Bailey & Rowley, 1987; Maclure et al., 2006; Imai et al., 2008; Bruhn & McKenzie, 2009), the theoretical implications of rerandomization were first formally studied by Morgan & Rubin (2012) using the Mahalanobis distance. Since then, rerandomization has attracted growing interest, and its theoretical foundations have been established across various scenarios (Morgan & Rubin, 2015; Li et al., 2018; Zhou et al., 2018; Li et al., 2020; Shi et al., 2024; Wang et al., 2023; Lu et al., 2023).

Although rerandomization can, in theory, achieve asymptotically optimal precision by shrinking the balance threshold as the sample size grows (Wang & Li, 2022), it is often regarded as *computationally infeasible* and therefore implemented with suboptimal thresholds when compared with alternative methods (Yang et al., 2023; Harshaw et al., 2024). Because the statistical properties of rerandomization critically depend on the choice of threshold, this computational issue leads to suboptimal statistical performance and results in unfair comparisons with existing methods. The

computational limitation of rerandomization stems from its reliance on naive rejection-sampling, which typically yields a single accepted assignment only after evaluating thousands of candidate allocations based on their Mahalanobis distance. In practice, as the number of covariates increases, the optimal acceptance probability can become astronomically small (e.g., $< 10^{-15}$ with 20 covariates), making naive rejection–sampling implementations of rerandomization practically infeasible.

This computational challenge is amplified when applying Fisher randomization tests (FRT) under rerandomization. Constructing the null distribution requires generating hundreds or thousands of acceptable assignments, compounding an already slow and costly process. Yet, FRT is particularly vital in small-sample settings: Johansson et al. (2021) shows that the asymptotic confidence–interval results of Li et al. (2018) fail to control Type I error when n is limited. Consequently, FRT has been widely advocated as a robust alternative, supported by extensive empirical evidence (Bind & Rubin, 2020; Keele, 2015; Proschan & Dodd, 2019; Young, 2019) and theoretical analyses (Branson, 2021; Cohen & Fogarty, 2022; Caughey et al., 2023; Luo et al., 2021; Wu & Ding, 2021; Zhao & Ding, 2021).

Several methods have been proposed to address the pressing need for accelerating rerandomization, including incorporating heuristic rules to search for satisfactory assignments (Krieger et al., 2019; Zhu & Liu, 2022), directly tackling the tradeoff between robustness and covariate balance by a Gram-Schmidt design (Harshaw et al., 2024), and engineering techniques such as via key-based storage and GPU/TPU backends (Goldstein et al., 2025). Among these approaches, the most relevant to our study is Zhu & Liu (2022), which uses pair-switching to search for a well-balanced allocation. This procedure substantially reduces computational cost and improves practical feasibility. However, due to the nature of the algorithm, it does not guarantee to sample uniformly from the set of acceptable assignments. Consequently, the uniformity condition underpinning asymptotic randomization-based inference is not assured, and those established theoretical results for classical rerandomization in Li et al. (2018) and Wang & Li (2022) do not directly apply.

Our contributions are twofold—theoretical and practical. On the theoretical side, we build on the Metropolis–Hastings framework to construct a Markov chain over the space of treatment assignments via pair switching. We derive the stationary distribution of this Markov chain, thus establishing the distribution of the acceptable assignment space it generates. Starting from this stationary distribution, we then apply rejection sampling to restore uniformity over this acceptable space. These results provide formal guarantees for the uniformity of the generated assignments, therefore validating theoretical results derived for classical rerandomization. Because the guarantees are asymptotic (valid as the number of iterations increases), we introduce an efficient stopping rule and translate our framework into a practical algorithm, PSRSRR. We show on both simulated and real data that PSRSRR yields assignment sets that are approximately uniformly distributed while substantially reducing sampling time compared to classical rerandomization. Our method bridges theory and practice, delivering a fast and reliable rerandomization procedure with desired theoretical guarantees.

The remainder of this paper is organized as follows. In Section 2, we introduce some preliminary knowledge of classical rerandomization. In Section 3, we develop our methodology in three steps: We begin with the formulation of the Markov chain given by pair-switching assignments, and derive the theoretical results on the generated distribution. We then incorporate rejection sampling to restore uniformity with theoretical foundations. Finally, we turn this theoretical result into a practical algorithm with a stopping rule that accelerates the rerandomization process while maintaining good enough uniformity. In Section 4, we present comprehensive experiment results on simulated and real data to illustrate the superior performance of our proposed algorithm in both estimation efficiency and sampling speed. We summarize and discuss our results in Section 5. Technical proofs and additional experimental results are in Appendix A and B.

2 Preliminaries

2.1 The Neyman-Rubin Potential Outcome Framework

This study adopts the Neyman-Rubin potential outcome framework (Neyman, 1923; Rubin, 1974). We consider an experiment with n units randomly drawn from a population, among which n_t units are treated and $n_c = n - n_t$ units are controlled. We denote $\mathbf{W} = (W_1, \ldots, W_n)^{\mathbf{T}}$ as the vector of

treatment assignment indicators, where $W_i = 1$ if unit i receives treatment and $W_i = 0$ otherwise. For each unit i, we consider the existence of two potential outcomes, $(Y_i(1), Y_i(0))$, and assume the stable unit treatment value assumption(SUTVA) (Rubin, 1980), i.e., $Y_i = W_i Y_i(1) + (1 - W_i) Y_i(0)$. The unit-level treatment effect for unit i is defined as $\tau_i = Y_i(1) - Y_i(0)$, and the average treatment effect is defined as $\tau = \frac{1}{n} \sum_{i=1}^{n} (Y_i(1) - Y_i(0))$, which could be estimated using the difference-in-means estimator $\widehat{\tau}(\mathbf{W}) = \frac{1}{n_t} \sum_{i:W_i=1} Y_i(1) - \frac{1}{n_c} \sum_{i:W_i=0} Y_i(0)$. For each unit, we also observe p baseline covariates, denoted by $\mathbf{X_i} = (X_{i1}, \ldots, X_{ip})^{\mathbf{T}}$. The covariates of all units are gathered in a matrix $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_n)^{\mathbf{T}}$, and the corresponding covariance matrix is denoted by $\mathbf{S}_{XX} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{X}_i - \overline{\mathbf{X}})(\mathbf{X}_i - \overline{\mathbf{X}})^{\mathbf{T}}$, where $\overline{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i$.

2.2 Classical Rerandomization Using the Mahalanobis Distance

Mahalanobis distance can be used to measure the covariate balance between the treatment and control groups. For a given assignment $\mathbf{W}$, the Mahalanobis distance is defined as $M(\mathbf{W}) := \left(\overline{\mathbf{X}}_t - \overline{\mathbf{X}}_c\right)^{\mathbf{T}} \left[\mathrm{Cov}\left(\overline{\mathbf{X}}_t - \overline{\mathbf{X}}_c\right)\right]^{-1} \left(\overline{\mathbf{X}}_t - \overline{\mathbf{X}}_c\right)$. Morgan & Rubin (2012) suggested performing randomization by sampling a treatment assignment $\mathbf{W}$ from the set $\mathcal{W} = \{W \in \mathbb{R}^n : \sum_{i=1}^{n} W_i = n_t, W_i \in \{0, 1\}\}$. For the sampled assignment $\mathbf{W}$, its Mahalanobis distance $M(\mathbf{W})$ is compared against a pre-specified threshold a that controls the acceptance level of the covariate imbalance. If $M(\mathbf{W}) \leq a$, the assignment is accepted; otherwise, the sampling process is repeated until a satisfactory assignment is found. We refer to rerandomization based on this acceptance-rejection sampling strategy as RR, and denote the set formed by all acceptable assignments as $\mathcal{W}_a = \{\mathbf{W} \in \mathcal{W} : M(\mathbf{W}) \leq a\}$.

3 Method

3.1 Stationary Distribution of Pair-switching Markov Chain

Classical rerandomization inefficiently searches for balanced assignments via rejection sampling. To improve this process, we propose a constructive approach based on a Metropolis-Hastings algorithm. Our method starts with a single random assignment and iteratively refines it. In each step, a candidate assignment is proposed by swapping a randomly selected treatment-control pair. This candidate is then accepted or rejected based on a probability determined by the change in the Mahalanobis distance, $M(\mathbf{W})$. The temperature, T, is a tuning parameter that controls the likelihood of accepting a candidate with a worse balance (i.e., a higher $M(\mathbf{W})$), allowing the search to escape local minima. This process is repeated for a fixed number of iterations, N. The complete procedure is detailed in Algorithm 1, which will serve as a crucial building block for our final proposed algorithm PSRSRR.

Algorithm 1: Truncated Pair-Switching

Input: Covariates data $\mathbf{X}$, temperature T, max iteration number N.
Set $t = 0$;
Set $\mathbf{W}^{(0)}$ as n_t elements equal to 1 and n_c elements equal to 0 with random positions;
Set $M^{(0)} = M\left(\mathbf{W}^{(0)}\right)$;
while $t < N$ **do**
  Randomly switch the positions of one of the 1's and one of the 0's in $\mathbf{W}^{(t)}$ and obtain $\mathbf{W}^*$;
  Set $M^* = M\left(\mathbf{W}^*\right)$;
  Sample J from a Bernoulli distribution with probability $\min\{\left(M^{(t)}/M^*\right)^{1/T}, 1\}$;
  if $J = 1$ **then**
    Set $\mathbf{W}^{(t+1)} = \mathbf{W}^*$;
  end
  else
    Set $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)}$;
  end
  Set $t = t + 1$;
end
Output: $\mathbf{W} = \mathbf{W}^{(N)}$.

By the definition of Markov chain (see for example, Givens & Hoeting (2012)), the sequence $\{\mathbf{W}^{(t)}\}_{t\geq 0}$ in Algorithm 1 forms a Markov chain over the space of all valid assignments, $\mathcal{W}$. As a result, as the number of iterations $N \to \infty$, the distribution of the generated assignments converges to a stationary distribution. This stationary distribution is characterized by the following theorem.

Theorem 1 *The limiting distribution of the Markov chain $\{\mathbf{W}^{(t)}\}_{t\geq 0}$ with temperature T is $\pi(\mathbf{W}) = \frac{M(\mathbf{W})^{-1/T}}{\sum_{\mathbf{W}^*\in\mathcal{W}} M(\mathbf{W}^*)^{-1/T}}$ for any $\mathbf{W} \in \mathcal{W}$. π is also the stationary distribution.*

This stationary distribution is not uniform; the probability of sampling an assignment, $\pi(\mathbf{W})$, is inversely proportional to its Mahalanobis distance raised to a positive exponent ($\pi(W) \propto M(\mathbf{W})^{-1/T}$). We will leverage this non-uniform distribution in the next section to generate assignments that are uniform over the acceptable set, $\mathcal{W}_a$.

3.2 Rejection Sampling

Although Algorithm 1 introduces a probabilistic mechanism for assignment generation and possesses a valuable theoretical guarantee, it can not be used directly for rerandomization because of two reasons. First, its final output is not guaranteed to be an acceptable assignment (i.e., it may have a Mahalanobis distance greater than the threshold). Second, its final output is not uniformly distributed. This departure from uniformity invalidates the theoretical guarantees that underpin classical rerandomization and can compromise the statistical efficiency of the resulting treatment effect estimates. Prior work like the PSRR method (Zhu & Liu, 2022) solves the first problem by letting the algorithm stop at the first accpetable assignment, but fails to solve the second.

To solve these challenges, we introduce a second step based on the principle of rejection sampling. The key insight is to treat the non-uniform stationary distribution π (from Theorem 1) as a proposal distribution and then apply a corrective filter to obtain our target uniform distribution over the acceptable set $\mathcal{W}_a$. We achieve this based on a carefully designed formula of the acceptance probability.

The acceptance rule has two components. First, to ensure acceptability, any proposed assignment $\mathbf{W}$ that does not meet the balance criterion ($M(\mathbf{W}) > a$) is automatically rejected by setting its acceptance probability to zero. Second, to ensure uniformity among the remaining candidates, we apply the "inverse back" strategy. From Theorem 1, we know the probability of proposing an assignment is *inversely proportional to* $M(\mathbf{W})^{1/T}$. To cancel this known bias, the acceptance probability for a valid candidate is made *directly proportional to* $M(\mathbf{W})^{1/T}$. The initial sampling bias and the corrective acceptance probability thereby cancel each other out, making the final probability constant for all assignments in $\mathcal{W}_a$. This two-part rule, formalized in Algorithm 2, results in a uniform sample from the acceptable set.

Algorithm 2: Rejection Sampling of Truncated Pair-Switching

Input: Covariates data $\mathbf{X}$, temperature T, max iteration number N, threshold a.
Set $Acc = \text{False}$;
while $Acc = \text{False}$ **do**
  Run Algorithm 1 with inputs$(\mathbf{X}, T, N)$ to generate assignment $\mathbf{W}$;
  Determine the acceptance probability

$$p(\mathbf{W}) = \begin{cases} (M(\mathbf{W})/a)^{1/T} & M(\mathbf{W}) \leq a \\ 0 & M(\mathbf{W}) > a \end{cases} \tag{1}$$

  Sample J from Bernoulli variable with probability $p(\mathbf{W})$;
  if *J = 1* **then**
    Set $Acc = \text{True}$;
  end
end
Output: $\mathbf{W}$.

Theorem 2 *The assignment* $\mathbf{W}$ *generated by PSRSRR follows a uniform distribution on* $\mathcal{W}_a$*; that is, each* $\mathbf{W} \in \mathcal{W}_a$ *is selected with equal probability.*

We defer the proof of Theorem 2 to Appendix A. Since the assignments generated by PSRSRR follow the uniform distribution over $\mathcal{W}_a$, which is a fundamental assumption in Li et al. (2018), we can verify the unbiasedness of the resulting treatment effect estimator immediately and build upon their theoretical guarantees to construct asymptotic confidence intervals.

Corollary 3 *Let* $\chi^2_{p,a} \sim \chi^2_p \mid (\chi^2_p \leq a)$ *be a truncated* χ^2 *random variable,* U_p *be the first coordinate of the uniform random vector over the* $(p-1)$*-dimensional unit sphere. Let* $\nu_\xi\left(R^2, p_a, p\right)$ *be the* ξ *th quantile of* $\sqrt{1-R^2}\cdot\varepsilon_0 + \sqrt{R^2}\cdot\chi_{p,a}U_p$ *where* $\varepsilon_0 \sim \mathcal{N}(0,1)$*. Denote by* $\boldsymbol{s}_{Y(i),\boldsymbol{X}}$ *the sample covariance between potential outcomes and covariates,* $s^2_{Y(i)}$ *the sample variance of potential outcomes, and* $s^2_{Y(i)|X}$ *the projection of potential outcomes on covariates (with intercept term). Let* $s^2_{\tau|\boldsymbol{X}} = \left(s_{Y(1),\boldsymbol{X}} - s_{Y(0),\boldsymbol{X}}\right)\left(\boldsymbol{S}^2_X\right)^{-1}\left(s_{\boldsymbol{X},Y(1)} - s_{\boldsymbol{X},Y(0)}\right)$*. Let* $\hat{V}_{\tau\tau} = n/n_t \cdot s^2_{Y(1)} + n/n_c \cdot s^2_{Y(0)} - s^2_{\tau|\boldsymbol{X}}$*. Let* $\hat{R}^2 = \hat{V}^{-1}_{\tau\tau}\left\{n/n_t \cdot s^2_{Y(1)|\boldsymbol{X}} + n/n_c \cdot s^2_{Y(0)|\boldsymbol{X}} - s^2_{\tau|\boldsymbol{X}}\right\}$*. An asymptotic* $(1-\alpha)\times 100\%$ *confidence interval of the difference-in-means estimator is given by* $\tau \in \left[\hat{\tau} - \nu_{\alpha/2}\left(\hat{R}^2, p_a, p\right)\sqrt{\hat{V}_{\tau\tau}/n},\quad \hat{\tau} - \nu_{1-\alpha/2}\left(\hat{R}^2, p_a, p\right)\sqrt{\hat{V}_{\tau\tau}/n}\right]$.

3.3 Practical Implementation

The procedure in Algorithm 2 is theoretically perfect: it is guaranteed to produce a uniform sample from the acceptable set $\mathcal{W}_a$. However, it can be computationally slow, as it requires running a full Markov chain (Algorithm 1) for many iterations just to generate a single candidate, which might then be rejected. This process is repeated until a candidate is finally accepted.

To bridge the gap between theoretical purity and practical speed, we introduce our main algorithm, Pair-Switching Rejection Sampling Rerandomization (PSRSRR). This algorithm fuses the Markov chain search and the rejection sampling check into a single, efficient procedure. Instead of running a full chain to draw a single candidate from the stationary distribution, we run a single chain and perform the acceptance check on-the-fly.

As the chain evolves from state $\mathbf{W}^{(t)}$ to $\mathbf{W}^{(t+1)}$, we check if the new state is acceptable (i.e., if $M(\mathbf{W}^{(t+1)}) < a$). If it is, we immediately apply the "inverse back" rejection sampling step. The chain terminates the very first time a candidate passes this second check. While this "early-stopping" heuristic no longer provides a formal guarantee of perfect uniformity, our experiments show that it produces a distribution that is nearly uniform in practice, with a substantial reduction in computational cost. This practical procedure is formalized in Algorithm 3.

4 Experiments

4.1 Simulation Studies

Objective Our simulation studies have three primary objectives. First, we empirically test whether our practical algorithm generates a nearly uniform distribution over the set of acceptable assignments. Second, we compare our method against competing methods on key statistical metrics, including mean squared error (MSE), confidence interval coverage, and statistical power. Third, we compare the computational time of our method with existing methods to demonstrate the computational efficiency of our method.

Simulation setup Covariates are drawn from the standard normal distribution identically and independently: $X_{ij} \overset{i.i.d.}{\sim} N(0,1)$, $i = 1,\ldots,n$, $j = 1,\ldots,p$. The potential outcomes of the control group are generated independently from a linear model, $Y_i(0) = \sum_{j=1}^p X_{ij} + \epsilon_i$, where $\epsilon_i \sim N(0,\sigma^2)$. σ^2 is selected such that $R^2 = \mathbb{V}(\sum_{j=1}^p X_{ij})/\mathbb{V}(\sum_{j=1}^p X_{ij} + \epsilon_i) = 0.2$, 0.5 or 0.8. We conduct simulations under both the null hypothesis and the alternative hypothesis, where we set $Y_i(1) = Y_i(0)$ and $Y_i(1) = Y_i(0) + 0.3\sqrt{\mathbb{V}[Y_i(0)]}$, respectively. The sizes of the treatment group

Algorithm 3: Pair-Switching Rejection Sampling Rerandomization (PSRSRR)

Input: Covariates data $\mathbf{X}$, threshold a, temperature T.
Set $Acc = \text{False}$;
Set $t = 0$;
Set $\mathbf{W}^{(0)}$ as n_t elements equal to 1 and n_c elements equal to 0 with random positions;
Set $M^{(0)} = M\left(\mathbf{W}^{(0)}\right)$;
while $Acc = \text{False}$ **do**
  Randomly switch the positions of one of the 1's and one of the 0 's in $\mathbf{W}^{(t)}$ and obtain $\mathbf{W}^*$;
  Set $M^* = M\left(\mathbf{W}^*\right)$;
  Sample J from a Bernoulli distribution with probability $\min\left\{\left(M^{(t)}/M^*\right)^{1/T}, 1\right\}$;
  if $J = 1$ **then**
    Set $\mathbf{W}^{(t+1)} = \mathbf{W}^*$;
    Set $M^{(t+1)} = M^*$;
    if $M^{(t+1)} < a$ **then**
      Sample $\tilde{J}$ from a Bernoulli distribution with probability $\left(M^{(t+1)}/a\right)^{1/T}$;
    end
    if $\tilde{J} = 1$ **then**
      Set $Acc = \text{True}$;
    end
  end
  else
    Set $\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)}$;
  end
  Set $t = t + 1$;
end
Output: $\mathbf{W} = \mathbf{W}^{(t)}$.

and the control group are set as equal. More detailed settings regarding sample sizes n and their corresponding sets of number of covariates p are deferred to Appendix B.1.

We use two strategies for setting the acceptance threshold a. The first is the conventional approach, which sets a to a small quantile of the χ^2_p distribution, (e.g., $p_a = \mathbb{P}\left(\chi^2_p \leq a\right) = 10^{-3}$ or 10^{-5}). The second strategy chooses a based on the desired asymptotic variance reduction. Specifically, we set $\nu_{p,a} = \mathbb{P}\left(\chi^2_{p+2} \leq a\right) / \mathbb{P}\left(\chi^2_p \leq a\right) = 0.01$ as recommended by Wang & Li (2022).

We set the temperature hyperparameter T using the empirical rule $T = 1.8/p$. The intuition is that as the number of covariates p increases, the Mahalanobis distance landscape becomes smoother, meaning a single pair-switch yields a smaller change in $M(\mathbf{W})$. A lower temperature is therefore appropriate, as large, unfavorable jumps become less necessary to explore the assignment space effectively.

Competing methods We benchmark our method against four competing methods, using hyperparameters as recommended in their respective papers: (1) PSRR with temperature as 0.1 and $p_a = \mathbb{P}\left(\chi^2_p \leq a\right) = 10^{-5}$ (Zhu & Liu, 2022); (2) GSW with $\phi = 0.1$ (Harshaw et al., 2024). (3) Classical rerandomization (RR) with acceptance threshold $M(\mathbf{W}) < a$, where a satisfies $p_a = \mathbb{P}\left(\chi^2_p \leq a\right) = 10^{-3}$; (4) Complete randomization (CR) that randomly draws assignments from all possible ones.

Uniformity We provide strong indirect evidence for the near-uniformity of our practical algorithm throughout our main results, as the validity of the statistical inference we construct relies on the fundamental assumption of uniformity. We defer a direct statistical evaluation using the Kolmogorov-Smirnov test to Appendix B.2.

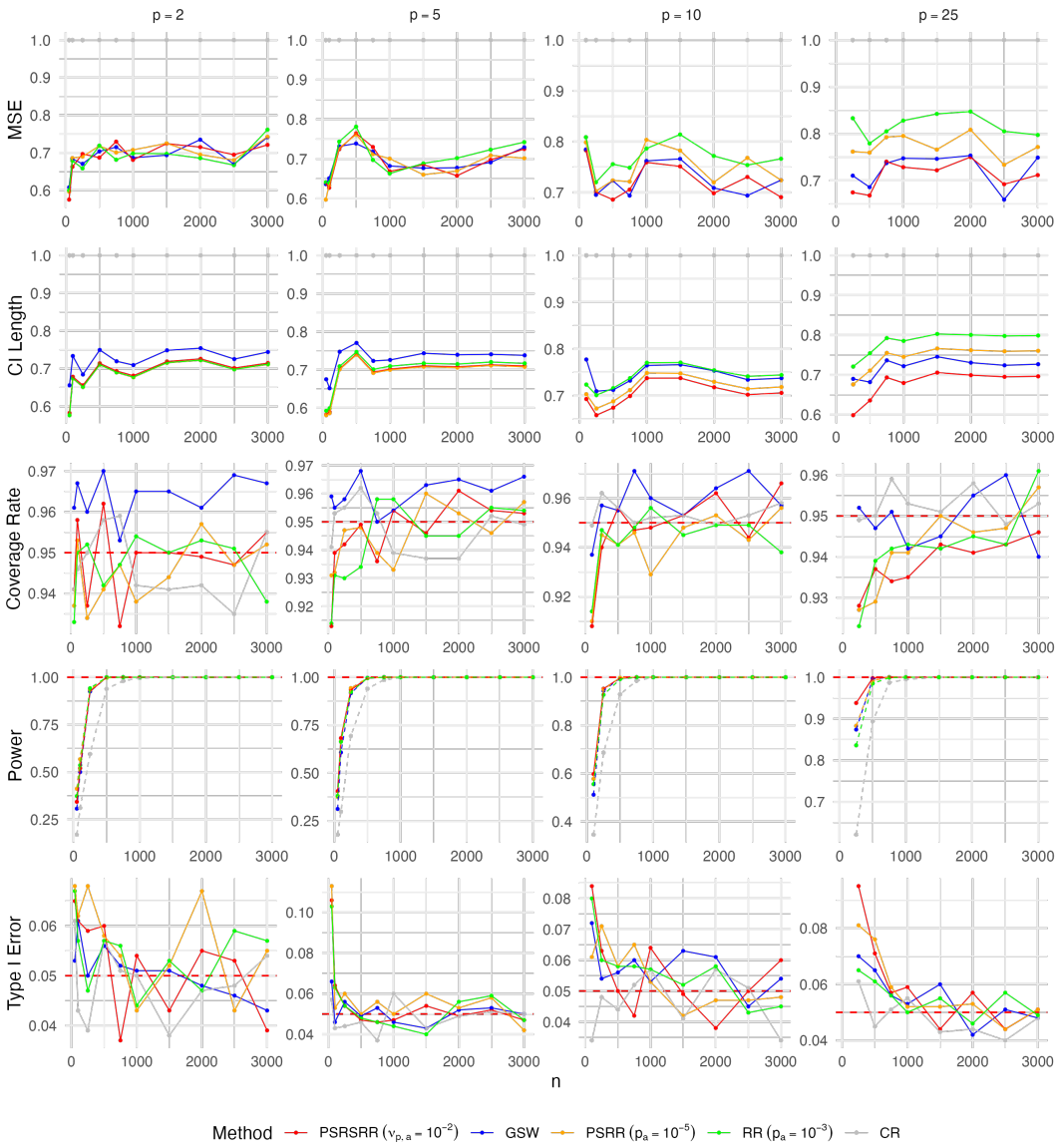


Figure 1: Comparison plots of, from top to bottom, MSE (relative to CR), CI length (relative to CR), coverage rate, power and type I error, by sample size n and number of covariates p. The red dashed lines represent the nominal level of 95% CI coverage rate and 0.05 type I error. Better performance is indicated by lower MSE and CI Length, a coverage rate at or above 95%, higher power, and a type I error at or below 0.05.

Evaluation metrics for estimation and inference For each method, we evaluate the performance of its resulting treatment effect estimator using five key statistical metrics. To assess estimation efficiency, we measure the estimator's mean squared error (MSE) and the average length of its corresponding confidence interval (CI). Both are reported as ratios relative to CR. To assess the validity of statistical inference, we evaluate the CI Coverage Rate (which should exceed the nominal 95% level), the Type I Error rate under the null hypothesis, and the statistical power under the alternative hypothesis.

Comparison of estimation and inference results For each method, we sample 1000 assignments to evaluate their estimation and inference performance. The results for $R^2 = 0.5$ are presented below; additional reults can be found in Appendix B.3. As demonstrated by Figure 1, PSRSRR has the best and comparable performance as GSW in terms of the relative MSE, and the best performance in the relative confidence interval length. More detailed simulation results show that PSRSRR could

have relative MSE ratio compared with CR as $58\% \sim 78\%$, $81\% \sim 107\%$ with RR, and $88\% \sim 107\%$ with PSRR. And when compared with PSRR, the improvement in MSE appears to be more significant when p is larger, for example, when $p = 25$, the relative MSE ratio compared with PSRR is $88\% \sim 94\%$. And we can obtain similar statistics for the relative confidence interval length ratio, which is $58\% \sim 74\%$ compared with CR, $83\% \sim 101\%$ with RR, $89\% \sim 101\%$ with PSRR, and $86\% \sim 97\%$ with GSW. As for CI coverage rate, power, and Type I error, PSRSRR achieves satisfactory results and has comparable or better performance than the other competing methods.

Comparison of sampling time We compare the computational efficiency of each method by measuring the average time required to generate 100 assignments. As shown in Figure 2, PSRSRR is highly efficient. Its sampling time is comparable to the fast but non-uniform PSRR method and is faster than GSW and RR. Notably, PSRSRR achieves a dramatic speedup over these competitors, which is, on median, 4 times faster than GSW and over 1,800 times faster than RR.

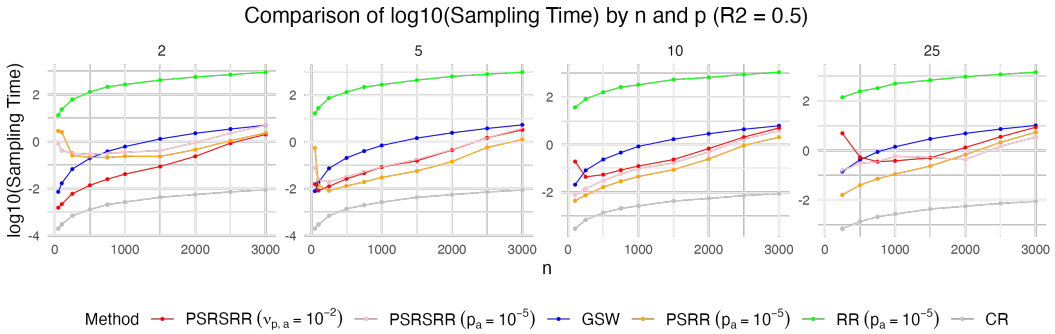


Figure 2: Comparison of sampling time (on a $\log_{10}$ scale) by sample size n and number of covariates p. CR is plotted to provide a baseline for the computational cost of a single random draw, rather than as a benchmark to outperform.

4.2 Application to Real Datasets

We apply PSRSRR to two real-world experimental datasets, one is the reserpine data (Jones, 2017) with 30 participants, and the other is the data from the Student Achievement and Retention (STAR) Project (Angrist et al., 2009) with nearly 1000 participants. The difference in these two datasets helps us to examine the performance of algorithms in both small-sample and large-sample circumstances, thus providing a more overall illustration. We provide a description for the analysis of STAR data and defer that of reserpine data and other details to Appendix B.4.

Table 1: Sampling Time Comparison in Reserpine and STAR Data.

Data	**PSRSRR** ($\nu_{p,a}$)	**PSRSRR** (p_a)	**PSRR**	**RR**	**CR**
Reserpine	–	0.47s	0.30s	22.70s	0.03s
STAR	2.95s	3.63s	1.28s	193.65s	0.21s

STAR data Similar to the pre-processing procedures in Li et al. (2018) and Wang & Li (2022), we drop the students with missingness in some important variables, resulting in the treatment group of size $n_1 = 118$ and control group $n_0 = 856$. And we include the following variables to balance: high-school GPA, age, gender and indicators for whether lives at home and whether rarely puts off studying for tests. We exclude GSW due to its incompatibility in dealing with exact imbalanced designs, and include PSRSRR using both $p_a = 10^{-3}$ and $\nu_{p,a} = 0.01$ for threshold selection, while setting $p_a = 10^{-3}$ for other methods.

We generate $10,000$ assignments, and obtain the estimation performance result in Figure 3 and sampling time performance in Table 1. The results show that both selection strategies of PSRSRR have achieved much faster sampling speed than RR and much improved variance reduction compared

with CR. Regarding the different strategies used for threshold selection, we can conclude from the results that PSRSRR ($\nu_{p,a}$) is able to reduce most of the variance among all methods, and has an even shorter sampling time compared with PSRSRR (p_a). This illustrates the strength of this threshold selection strategy in asymptotic scenarios.

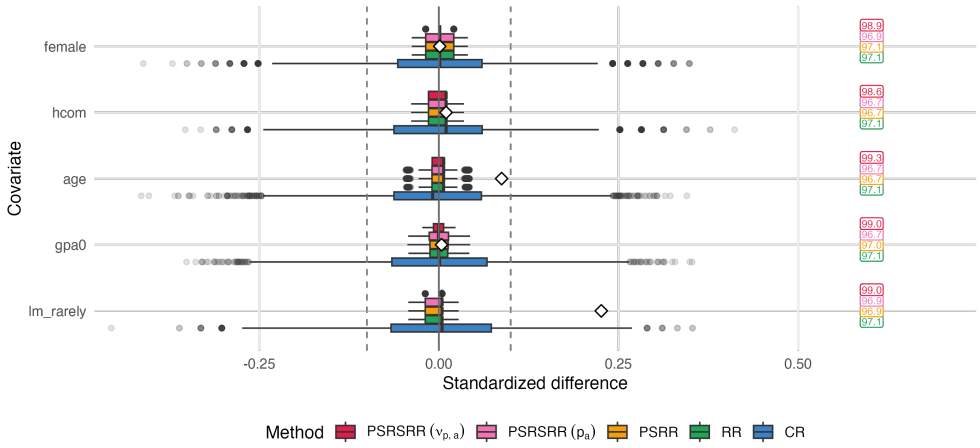


Figure 3: Box-plot of standardized differences in covariate means for STAR data. The diamonds indicates the standardized difference for the actual assignment in the experiment. The values on the right side are PRIVs, the empirical percent reductions in variance compared with CR of each method.

5 Conclusion and Discussion

Our study demonstrates that rerandomization, long considered computationally impractical under stringent thresholds, can be made both fast and theoretically sound through a Metropolis–Hastings framework with an importance resampling correction. The key insight is that pair–switching updates naturally form a Markov chain whose stationary distribution favors balanced allocations, and by layering rejection sampling we recover exact uniformity over the accepted set. In practice, a simple early–stopping rule yields nearly uniform assignments while accelerating computation by orders of magnitude. Extensive simulations and real–data applications show that this approach preserves the inferential validity of classical rerandomization, narrows confidence intervals, and reduces mean squared error, all while drastically cutting down runtime.

Still, our framework opens several new directions. First, more advanced MCMC kernels may further improve mixing and exploration of the assignment space. Techniques such as adaptive tempering or hybrid proposals (Liang et al., 2011) could be incorporated, potentially achieving better balance or faster convergence. Second, complex experiments are increasingly central to causal inference (Cinelli et al., 2025). Extending our method to factorial, clustered, stratified, or sequential designs will be crucial for ensuring that rerandomization remains feasible and theoretically justified in these contexts. Third, while we improved both the algorithmic efficiency and the implementation via Rcpp, complementary system-level advances are emerging. For example, Goldstein et al. (2025) leverage hardware–accelerated tools for rerandomization and randomization testing. Combining their acceleration with our sampling framework could make strict thresholds practical at scale.

Taken together, our results show that rerandomization need not force a tradeoff between balance and computational feasibility. By unifying fast sampling with rigorous inference guarantees, our approach makes rerandomization a practical tool for modern experimental research, and lays the groundwork for further advances in both methodology and applications.

Reproducibility Statement

The code for reproducing our experiments will be released publicly following the double-blind review process. The technical proofs and details regarding data preprocessing can be found in Appendix A and B, respectively.

References

Joshua D. Angrist, Daniel Lang, and Philip Oreopoulos. Incentives and services for college achievement: Evidence from a randomized trial. *American Economic Journal: Applied Economics*, 1(1): 136–163, 2009. doi: 10.1257/app.1.1.136.

Peter C. Austin. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, 28(25):3083–3107, 2009. doi: 10.1002/sim.3697.

RA Bailey and CA Rowley. Valid randomization. *Proceedings of the royal society of London. A. Mathematical and Physical Sciences*, 410(1838):105–124, 1987.

Marie-Abele C. Bind and Donald B. Rubin. When possible, report a fisher-exact p value and display its underlying null randomization distribution. *Proceedings of the National Academy of Sciences*, 117(32):19151–19158, 2020.

Zach Branson. Randomization tests to assess covariate balance when designing and analyzing matched datasets. *Observational Studies*, 7(2):1–36, 2021.

Miriam Bruhn and David McKenzie. In pursuit of balance: Randomization in practice in development field experiments. *American economic journal: applied economics*, 1(4):200–232, 2009.

Devin Caughey, Allan Dafoe, Xinran Li, and Luke Miratrix. Randomisation inference beyond the sharp null: bounded null hypotheses and quantiles of individual treatment effects. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 85(5):1471–1491, November 2023. doi: 10.1093/jrsssb/qkad080. URL https://doi.org/10.1093/jrsssb/qkad080.

Carlos Cinelli, Avi Feller, Guido Imbens, Edward Kennedy, Sara Magliacane, and Jose Zubizarreta. Challenges in statistics: A dozen challenges in causality and causal inference. *arXiv preprint arXiv:2508.17099*, 2025.

Paul L. Cohen and Colin B. Fogarty. Gaussian prepivoting for finite population causal inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(2):295–320, 2022.

DR Cox. Randomization and concomitant variables in the design of experiments. statistics and probability.(eds. g kallianpur, p krishnaiah, j ghosh) pp. 197-202, 1982.

Geof H. Givens and Jennifer A. Hoeting. *Computational Statistics*. John Wiley & Sons, Hoboken, NJ, 2012. ISBN 9780470533314. doi: 10.1002/9781118555552. URL https://doi.org/10.1002/9781118555552. First published: 22 October 2012.

Rebecca Goldstein, Connor T Jerzak, Aniket Kamat, and Fucheng Warren Zhu. fastrerandomize: An r package for fast rerandomization using accelerated computing. *arXiv preprint arXiv:2501.07642*, 2025.

Mor Harchol-Balter. *Introduction to Probability for Computing*. Cambridge University Press, Cambridge, UK, 2024.

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*, pp. 1–13, 2024.

Kosuke Imai, Gary King, and Elizabeth A Stuart. Misunderstandings between experimentalists and observationalists about causal inference. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 171(2):481–502, 2008.

Per Johansson, Donald B. Rubin, and Martin Schultzberg. On optimal rerandomization designs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 83(2):395–403, 2021.

R. Jones. A phase 1 parallel-group, double-blind, placebocontrolled cardiovascular and behavioral study assessing interactions between single doses of oral reserpine and intravenous methamphetamine. *National Institute on Drug Abuse (NIDA) Data Share web site*, 2017.

Adam Kapelner, Abba M Krieger, Michael Sklar, Uri Shalit, and David Azriel. Harmonizing optimized designs with classic randomization in experiments. *The American Statistician*, 75(2): 195–206, 2021.

Luke Keele. The statistics of causal inference: A view from political methodology. *Political Analysis*, 23(3):313–335, 2015.

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

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.

Xinran Li, Peng Ding, and Donald B Rubin. Rerandomization in 2k factorial experiments. *Annals of Statistics*, 48(1):43–63, 2020.

Faming Liang, Chuanhai Liu, and Raymond Carroll. *Advanced Markov chain Monte Carlo methods: learning from past samples*. John Wiley & Sons, 2011.

Xin Lu, Tianle Liu, Hanzhong Liu, and Peng Ding. Design-based theory for cluster rerandomization. *Biometrika*, 110(2):467–483, 2023.

Xiaokang Luo, Tirthankar Dasgupta, Minge Xie, and Regina Y Liu. Leveraging the fisher randomization test using confidence distributions: Inference, combination and fusion learning. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(4):777–797, 2021.

Malcolm Maclure, Anne Nguyen, Greg Carney, Colin Dormuth, Hendrik Roelants, Kendall Ho, and Sebastian Schneeweiss. Measuring prescribing improvements in pragmatic trials of educational tools for general practitioners. *Basic & clinical pharmacology & toxicology*, 98(3):243–252, 2006.

Kari Lock Morgan and Donald B Rubin. Rerandomization to improve covariate balance in experiments. *The Annals of Statistics*, pp. 1263–1282, 2012.

Kari Lock Morgan and Donald B Rubin. Rerandomization to balance tiers of covariates. *Journal of the American Statistical Association*, 110(512):1412–1421, 2015.

Jerzy Neyman. On the application of probability theory to agricultural experiments. essay on principles. *Ann. Agricultural Sciences*, pp. 1–51, 1923.

Michael A. Proschan and Lori E. Dodd. Re-randomization tests in clinical trials. *Statistics in Medicine*, 38(12):2292–2302, 2019.

W. F. Rosenberger and O. Sverdlov. Handling covariates in the design of clinical trials. *Statistical Science*, 23(3):404–419, 2008.

Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

Donald B Rubin. Randomization analysis of experimental data: The fisher randomization test comment. *Journal of the American statistical association*, 75(371):591–593, 1980.

W. Shi, A. Zhao, and H. Liu. Rerandomization and covariate adjustment in split-plot designs. *Journal of Business & Economic Statistics*, pp. 1–22, 2024. doi: 10.1080/07350015.2024.2429464. URL `https://doi.org/10.1080/07350015.2024.2429464`. Advance online publication.

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

Xinhe Wang, Tingyu Wang, and Hanzhong Liu. Rerandomization in stratified randomized experiments. *Journal of the American Statistical Association*, 118(542):1295–1304, 2023.

Yuhao Wang and Xinran Li. Rerandomization with diminishing covariate imbalance and diverging number of covariates. *The Annals of Statistics*, 50(6):3439–3465, 2022.

Jingqin Wu and Peng Ding. Randomization tests for weak null hypotheses in randomized experiments. *Journal of the American Statistical Association*, 116(536):1898–1913, 2021.

Haoyu Yang, Yichen Qin, Fan Wang, Yang Li, and Feifang Hu. Balancing covariates in multi-arm trials via adaptive randomization. *Computational Statistics & Data Analysis*, 179:107642, 2023.

Alwyn Young. Channeling fisher: Randomization tests and the statistical insignificance of seemingly significant experimental results. *The Quarterly Journal of Economics*, 134(2):557–598, 2019.

Anqi Zhao and Peng Ding. Covariate-adjusted fisher randomization tests for the average treatment effect. *Journal of Econometrics*, 225(2):278–294, 2021.

Quan Zhou, Philip A Ernst, Kari Lock Morgan, Donald B Rubin, and Anru Zhang. Sequential rerandomization. *Biometrika*, 105(3):745–752, 2018.

Ke Zhu and Hanzhong Liu. Pair-switching rerandomization. *Biometrics*, 2022.

A Technical Proofs

A.1 Proof of Theorem 1

We first give a rigorous mathematical definition of the pair-switching Markov chain constructed in Algorithm 1. For any pair of assignments $\mathbf{W}_i, \mathbf{W}_j \in \mathcal{W}$ and temperature T, we use $\mathbb{Q}_T(\mathbf{W}_j|\mathbf{W}_i)$ to denote the transition probability from $\mathbf{W}_i$ to $\mathbf{W}_j$, i.e., the probability that $\mathbf{W}_i$ is updated to $\mathbf{W}_j$ within one step of pair-switching. The update rule indicated by Algorithm 1 could therefore be formulated as follows,

- if $\mathbf{W}_i$ and $\mathbf{W}_j$ are neighbors, then
$$\mathbb{Q}_T(\mathbf{W}_j|\mathbf{W}_i) = \frac{1}{n_t n_c} \min\{1, (M(\mathbf{W}_i)/M(\mathbf{W}_j))^{1/T}\}, \tag{2}$$
- if $\mathbf{W}_i$ and $\mathbf{W}_j$ are not neighbors and $\mathbf{W}_i \neq \mathbf{W}_j$, then
$$\mathbb{Q}_T(\mathbf{W}_j|\mathbf{W}_i) = 0, \tag{3}$$
- if $\mathbf{W}_i = \mathbf{W}_j$, then
$$\mathbb{Q}_T(\mathbf{W}_j|\mathbf{W}_i) = 1 - \sum_{k \neq i} \mathbb{Q}_T(\mathbf{W}_k|\mathbf{W}_i), \tag{4}$$

where two assignments are called neighbors if and only if one assignment can be obtained by switching one 0-1 pair in the other assignment. By definition, it is straightforward to identify the sequence $\{\mathbf{W}^{(t)}\}, t = 0, 1, \ldots,$ as a Markov chain (Givens & Hoeting, 2012, Equation (1.41)). The transition matrix of this Markov chain could be formulated as $\mathbf{Q_T} = (q_{ij})_{m\times m}$, $q_{ij} = \mathbb{Q}_T(\mathbf{W}_j|\mathbf{W}_i)$, where $m = \binom{n}{n_t}$ is the total number of all possible assignments, and $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_m\} = \mathcal{W}$ is the assignment space. From the perspective of Markov chain, Algorithm 1 can be interpreted as starting from an all-zeros distribution vector of length m except for the k-th entry being 1, i.e., the initial assignment $\mathbf{W}^{(0)} = \mathbf{W}_k$. We denote this distribution vector as $\pi^{(0)}$. Followed by N successive left-multiplications of the transition matrix $\mathbf{Q_T}$, i.e., after N iterations in Algorithm 1, the final output could be regarded as a randomly selected assignment according to the distribution $\pi^{(N)}$, where $\pi^{(N)} = (\mathbf{Q_T})^N \pi^{(0)}$.

With the above formulation and the perspective of Markov chain, we present the proof for Theorem 1, which relies on the following lemma.

Lemma 4 *The limiting distribution of the Markov chain* $\{\mathbf{W}^{(t)}\}, t = 0, 1, 2, \ldots,$ *exists and is equal to its stationary distribution.*

With this lemma, we only need to verify that the distribution of $\pi = \lim_{N\to\infty} \pi^{(N)}$ given in Theorem 1 is indeed the stationary distribution. Since $\forall i, j = 1, 2, \ldots, m$, we have $q_{ij}\pi_i = q_{ji}\pi_j$ given the formulation in Equation (2) (3) (4). This indicates that the distribution π satisfies the detailed balance conditions, and therefore is the stationary distribution of the Markov chain (Givens & Hoeting, 2012, Equation (1.43)). This completes the proof of Theorem 1.

Hence, it suffices to prove Lemma 4. We use the following result from Harchol-Balter (2024).

Lemma 5 *(Summary theorem for ergodic, finite-state DTMCs, Theorem 25.19 in Harchol-Balter (2024)) In a finite-state DTMC, the word ergodic refers to two properties: aperiodic and irreducible. Given an ergodic finite-state chain, the following results hold:*

- *The limiting distribution exists and has all-positive components.*
- $\pi_j^{\textit{limiting}} = \frac{1}{m_{jj}}$.
- *The stationary distribution is unique and is equal to the limiting distribution.*
- *Time-average* $p_j = \frac{1}{m_{jj}}$*, w.p.1.*
- *Putting it all together, we have that:*

$$0 < \frac{1}{m_{jj}} = \pi_j^{\textit{limiting}} = \pi_j^{\textit{stationary}} = p_j, \textit{ w.p.1.}$$

According to Lemma 5, we only need to verify the irreducibility and aperiodicity of the Markov chain $\{\mathbf{W}^{(t)}\}$, since it has finite states.

To show its irreducibility, we denote any two assignments as $\mathbf{W}_i, \mathbf{W}_j \in \mathcal{W}$, and their treatment and control indices are $\mathcal{I}_t, \mathcal{I}_c$ and $\mathcal{J}_t, \mathcal{J}_c$, respectively. The shared indices in the treatment and control groups are denoted as sets $\mathcal{D}_t, \mathcal{D}_c$, respectively. Apparently, we have

$$\mathcal{I}_t \cup \mathcal{I}_c = \mathcal{J}_t \cup \mathcal{J}_c = \{1, 2, \ldots, m\},$$

and further

$$|\mathcal{I}_t \setminus \mathcal{D}_t| = |\mathcal{J}_t \setminus \mathcal{D}_t|, \quad |\mathcal{I}_c \setminus \mathcal{D}_c| = |\mathcal{J}_c \setminus \mathcal{D}_c|.$$

For the treatment indices of $\mathbf{W}_i$, if they are not shared by treatment indices of $\mathbf{W}_j$, then they will be included in the control indices of $\mathbf{W}_j$ not shared by control indices of $\mathbf{W}_i$, since treatment and control groups have no overlap, i.e., $\mathcal{I}_t \setminus \mathcal{D}_t \subseteq \mathcal{J}_c \setminus \mathcal{D}_c$. We can similarly obtain the conclusion that $\mathcal{I}_c \setminus \mathcal{D}_c \subseteq \mathcal{J}_t \setminus \mathcal{D}_t$. Therefore, we have

$$|\mathcal{I}_t \setminus \mathcal{D}_t| = |\mathcal{J}_t \setminus \mathcal{D}_t| = |\mathcal{I}_c \setminus \mathcal{D}_c| = |\mathcal{J}_c \setminus \mathcal{D}_c|,$$

indicating that pairs could be formed between $\mathcal{I}_t \setminus \mathcal{D}_t$ and $\mathcal{I}_c \setminus \mathcal{D}_c$. So after switching each pair, whose probability is positive as shown in the formulation of the transition matrix, the generated $\mathbf{W}_{i'}$ would be equal to $\mathbf{W}_j$, which verifies that any two states in this Markov chain could communicate with each other.

To establish its aperiodicity, we need to show that ${\mathbf{Q}_\mathbf{T}}^2$ and ${\mathbf{Q}_\mathbf{T}}^3$ both have positive diagonal elements, implying that the period is given by $\gcd(2, 3) = 1$, where $\gcd()$ denotes the greatest common divisor function. The property of ${\mathbf{Q}_\mathbf{T}}^2$ could be readily verified: as for any $\mathbf{W}_i \in \mathcal{W}$, the assignment can return to itself by switching the 0-1 pair to become its neighbor and then reversing that switch, with transition probabilities both greater than zero. As for ${\mathbf{Q}_\mathbf{T}}^3$, for any $\mathbf{W}_i \in \mathcal{W}$, denote p, s as the indices in the control group and q the index in the treatment group. Since transition probability between any neighbor is positive, we can easily verify the conclusion by the following derivation. First, the switch takes place between 0-1 pair (p, q). Then the second switch is performed between (p, s), and the third between (q, s), returning to the initial $\mathbf{W}_i$. Therefore, we confirm the aperiodicity of the chain. □

A.2 Proof of Theorem 2

We follow the steps given in Section 6.2.3 in Givens & Hoeting (2012) to give this proof.

Our target distribution is a uniform distribution on $\mathcal{W}_a$, i.e., the probability mass function would be $f(\mathbf{W}) = \frac{\mathbb{I}\{\mathbf{W}\in\mathcal{W}_a\}}{|\mathcal{W}_a|}$, and $\pi(\mathbf{W})$ denote the stationary distribution sampled from Algorithm 1. Let $e(\mathbf{W})$ denote an envelope function, and in this case, we have

$$e(\mathbf{W}) = \frac{\pi(\mathbf{W})}{\alpha} = \frac{M(\mathbf{W})^{-1/T}}{\alpha \cdot \sum_{w\in\mathcal{W}} M(\mathbf{W})^{-1/T}} \geq f(\mathbf{W}) = \frac{\mathbb{I}\{\mathbf{W}\in\mathcal{W}_a\}}{|\mathcal{W}_a|}$$

where α is a scaling parameter, and we choose

$$\alpha = \alpha_{\min} = \frac{a^{-1/T}|\mathcal{W}_a|}{\sum_{w\in\mathcal{W}} M(\mathbf{W})^{-1/T}}.$$

We go through the following sampling procedure.

1. Sample $\mathbf{Y} \sim g$;
2. Sample $J \sim \mathcal{B}er\left(\frac{f(\mathbf{Y})}{e(\mathbf{Y})}\right)$, where
$$\begin{aligned}\frac{f(\mathbf{Y})}{e(\mathbf{Y})} =& \frac{\mathbb{I}\{\mathbf{W}\in\mathcal{W}_a\}}{|\mathcal{W}_a|} \frac{\sum_{w\in\mathcal{W}} M(\mathbf{W})^{-1/T}}{M(\mathbf{W})^{-1/T}} \frac{a^{-1/T}|\mathcal{W}_a|}{\sum_{w\in\mathcal{W}} M(\mathbf{W})^{-1/T}} \\ =& \frac{M(\mathbf{W})^{1/T}\cdot\mathbb{I}\{\mathbf{W}\in\mathcal{W}_a\}}{a^{1/T}} \\ =& \mathbb{I}\{\mathbf{W}\in\mathcal{W}_a\}\left(\frac{M(\mathbf{W})}{a}\right)^{1/T} \\ =& p(\mathbf{W})\end{aligned}$$
which is the acceptance probability as defined in Algorithm 2;
3. Reject $\mathbf{Y}$ if $J = 0$, and do not record $\mathbf{Y}$ but instead return to step 1; Otherwise, keep the value of $\mathbf{Y}$, set $\mathbf{W} = \mathbf{Y}$.

We verify that the above sampling procedure could indeed generate the targeted distribution,

$$\begin{aligned}\mathbb{P}(\mathbf{W} = y) &= \mathbb{P}(\mathbf{Y} = y \mid J = 1) \\ &= \frac{\mathbb{P}(\mathbf{Y} = y, J = 1)}{\mathbb{P}(J = 1)} \\ &= \frac{\pi(y)\cdot\frac{f(y)}{e(y)}}{\sum\limits_{z\in\mathcal{W}} \pi(z)\cdot\frac{f(z)}{e(z)}} \\ &= \frac{\pi(y)\cdot\alpha\cdot\frac{f(y)}{\pi(y)}}{\sum\limits_{z\in\mathcal{W}} \pi(z)\cdot\alpha\cdot\frac{f(z)}{\pi(z)}} \\ &= \frac{\alpha f(y)}{\alpha\sum\limits_{z\in\mathcal{W}} f(z)} \\ &= f(y).\end{aligned}$$

Therefore, we conclude the proof. □

B Additional Experiment Details and Results

B.1 Detailed Simulation Settings

When conducting simulation studies to examine the performance of estimation results as well as sampling speeds, we use different sample sizes n and their corresponding sets of number of covari-

ates p as shown in Table 2 below. The range of sample size n could help illustrate the performance of our proposed method in a larger scale, from small-sample behavior to large-sample behavior. And for each sample size n, we choose different values of p, while not violating the asymptotic rule, $p = O(\log n)$, as demonstrated in Harshaw et al. (2024).

Table 2: Simulation setting regarding sample size and number of covariates.

n	p	n	p	n	p
50	$\{2, 5\}$	500	$\{2, 5, 10, 25\}$	2000	$\{2, 5, 10, 25\}$
100	$\{2, 5, 10\}$	1000	$\{2, 5, 10, 25\}$	2500	$\{2, 5, 10, 25\}$
250	$\{2, 5, 10, 25\}$	1500	$\{2, 5, 10, 25\}$	3000	$\{2, 5, 10, 25\}$

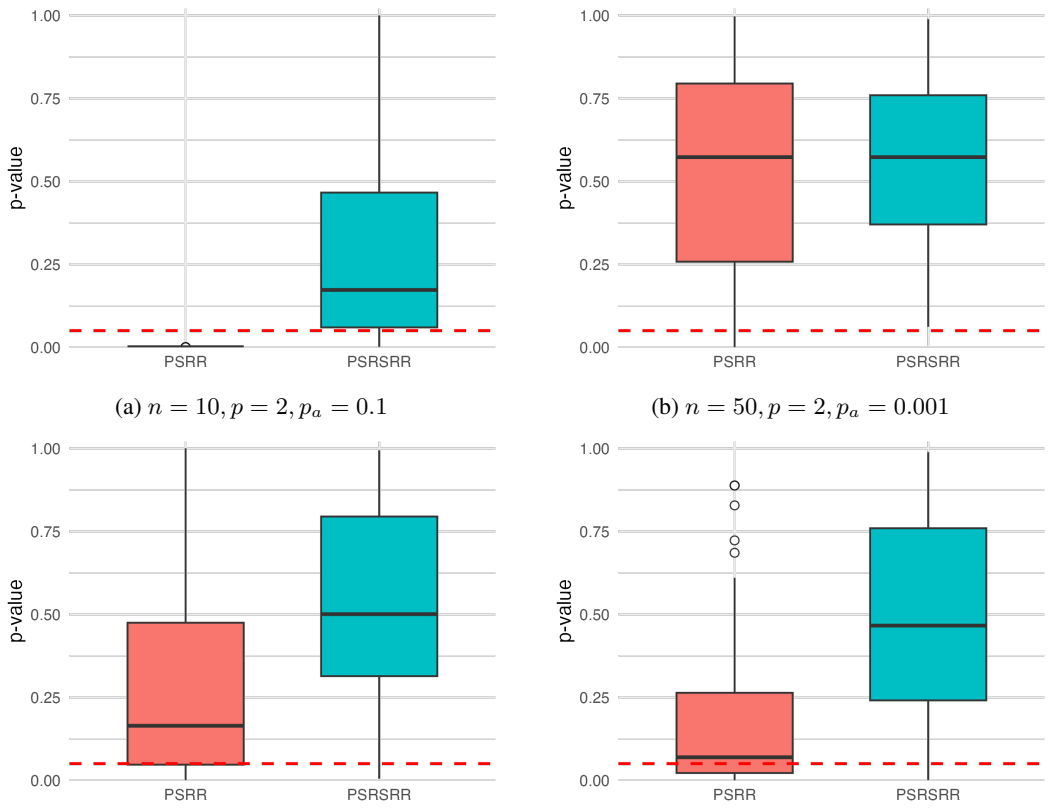
For each (n, p) pair, we have a total of 6 simulation settings, with $R^2 = 0.2, 0.5, 0.8$ and individual causal effect zero or non-zero, as described in the main body of the paper, to study and compare the estimation and inference results. As for the comparison of sampling times, we only keep one setting of each (n, p) pair, i.e., $R^2 = 0.5$ and non-zero effect, as changes in R^2 and the effect would not essentially affect the simulation speed and therefore one setting would already be sufficient in illustrating the acceleration performance of our proposed method PSRSRR.

B.2 Uniformity Verification

Besides the non-direct verification of near-uniformity of the assignment distribution generated by our proposed method PSRSRR, we here verify more directly whether the assignments generated by PSRSRR can be regarded as uniformly distributed and possess better uniformity than PSRR.

The verification procedure is designed as follows. For each setting, we generate $10,000$ assignments using PSRSRR, PSRR and RR, respectively. We calculate the Mahalanobis distance of each assignment and obtain the Mahalanobis distance distribution. We conduct one Kolmogorov-Smirnov test on whether the Mahalanobis distance of assignments generated by PSRSRR and that generated by RR are the same, and another Kolmogorov-Smirnov test on whether the Mahalanobis distance of assignments generated by PSRR and that generated by RR are the same. This way, we obtain one p-value for each hypothesis testing, which has the implication whether the accelerated algorithm could generate assignments that have the distribution to be accepted as the same as the assignment distribution generated by RR in the sense of Mahalanobis distance. We repeat this process for 100 times to get an empirical distribution of p-values, which can be visualized in a boxplot.

Due to the computational constraint of RR, we limit our verification to some small-sample settings, and also the threshold selection strategy to p_a only, since $\nu_{p,a}$-based strategy would likely lead to much stricter criteria that RR could hardly handle. We present the boxplots obtained from the verification procedures described above as follows. In all these settings, PSRSRR shows a great proportion of p-values above the rejection threshold 0.05, indicating a good alignment with RR, verifying the good uniformity of PSRSRR in the sense of Mahalanobis distance. While PSRR, especially in (a) and (d), appears to have many p-values below the 0.05 threshold, meaning that in many of the repetitions, the null hypothesis that the Mahalanobis distance distribution of the assignments generated by PSRR is the same as that generated by RR is rejected. Therefore, we can conclude that PSRSRR can generally be considered as a better approximation of RR than PSRR.



(a) $n = 10, p = 2, p_a = 0.1$

(b) $n = 50, p = 2, p_a = 0.001$

(c) $n = 50, p = 5, p_a = 0.001$

(d) $n = 100, p = 10, p_a = 0.0001$

Figure 4: Boxplots of Kolmogorov-Smirnov test p-value distribution. In each figure, the left boxplot is for H_0: PSRR = RR and the right one is for H_0: PSRSRR = RR. The red line indicates the 0.05 significance threshold of rejecting the null hypothesis.

B.3 Estimation and Inference Results for Additional Settings

In the main body of the paper, we have already presented and discussed results under the simulation settings with $R^2 = 0.5$. Here, we include more simulation results under settings with $R^2 = 0.2$ and $R^2 = 0.8$, in order to provide a more comprehensive overview of the performance of our proposed algorithm PSRSRR.

In the following, we present the comparison plots of MSE for all three R^2 settings, as well as the comparative statistics regarding the MSE improvement of PSRSRR with respect to other competing methods. We can obtain the similar conclusion as given in the main body of the paper, that PSRSRR has the best performance among all methods, with very similar estimation results as GSW. And when the number of covariates p increases, i.e., in more completed simulation settings, the improvement of PSRSRR compared with PSRR and CR is more stable and more significant. And when R^2 increases, i.e., the covariates are more explanatory and there is less noise, the improvement of PSRSRR becomes more significant (except for GSW). Therefore, MSE comparison results show that PSRSRR is superior, especially in large-sample and high-dimensional settings.

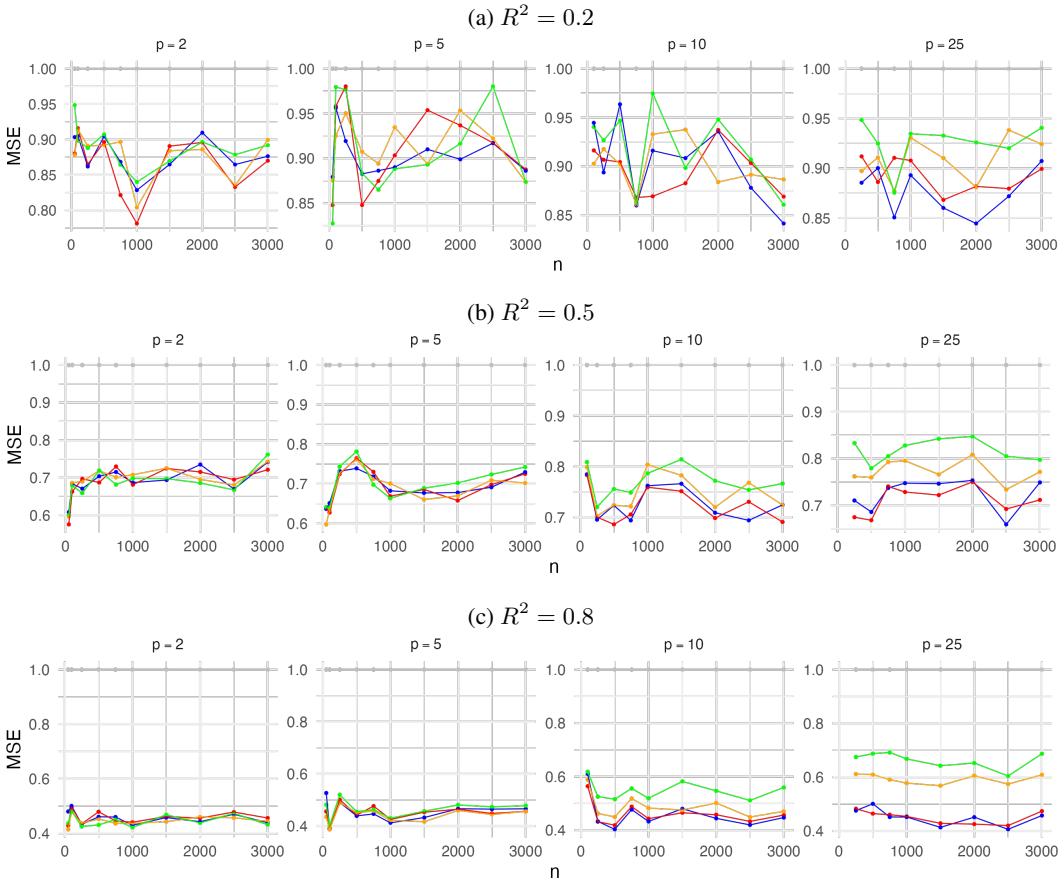


Figure 5: Comparison plots of MSE relative to CR, by sample size n and number of covariates p.

R^2	Statistic	Relative MSE Ratio			
		To CR	To RR	To PSRR	To GSW
0.2	Range	$[78\%, 98\%]$	$[89\%, 107\%]$	$[92\%, 107\%]$	$[94\%, 107\%]$
	Mean	89%	98%	99%	100%
0.5	Range	$[56\%, 78\%]$	$[81\%, 107\%]$	$[88\%, 107\%]$	$[95\%, 105\%]$
	Mean	70%	96%	97%	99%
0.8	Range	$[39\%, 56\%]$	$[65\%, 111\%]$	$[70\%, 109\%]$	$[87\%, 107\%]$
	Mean	46%	89%	95%	100%

Table 3: Relative MSE ratios: range and mean of PSRSRR compared with each competing method under different R^2.

In the following, we present the comparison plots of confidence interval length for all three R^2 settings, as well as the comparative statistics regarding the improvement of PSRSRR with respect to other competing methods. We can obtain similar conclusions that PSRSRR has consistently the best performance among all methods. And its superiority becomes more significant as R^2 increases, n increases and p increases, illustrating its strength in large-sample and high-dimensional settings.

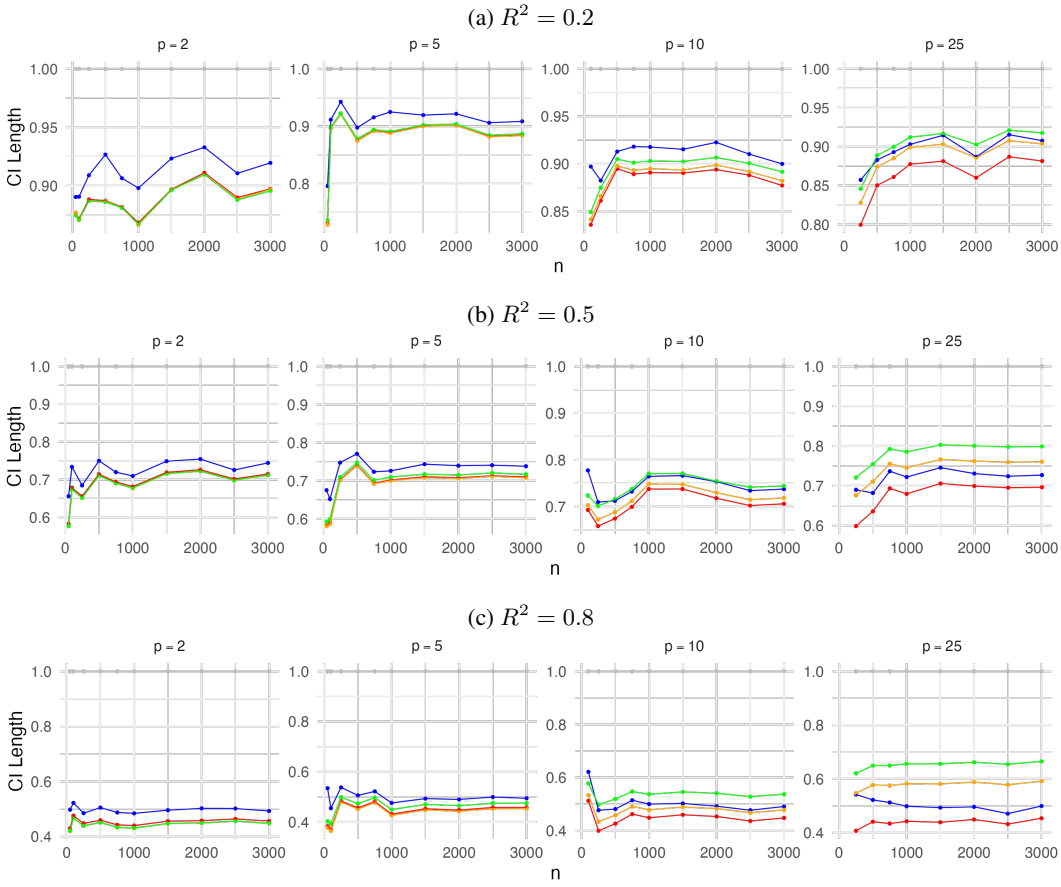


Figure 6: Comparison plots of CI length relative to CR, by sample size n and number of covariates p.

R^2	Statistic	Relative CI Length Ratio			
		To CR	To RR	To PSRR	To GSW
0.2	Range	$[73\%, 92\%]$	$[95\%, 100\%]$	$[97\%, 100\%]$	$[92\%, 98\%]$
	Mean	88%	99%	99%	97%
0.5	Range	$[58\%, 74\%]$	$[83\%, 101\%]$	$[88\%, 101\%]$	$[86\%, 97\%]$
	Mean	69%	96%	98%	94%
0.8	Range	$[37\%, 51\%]$	$[66\%, 102\%]$	$[74\%, 102\%]$	$[72\%, 93\%]$
	Mean	45%	88%	94%	89%

Table 4: Relative CI Length ratios: range and mean of PSRSRR compared with each competing method under different R^2.

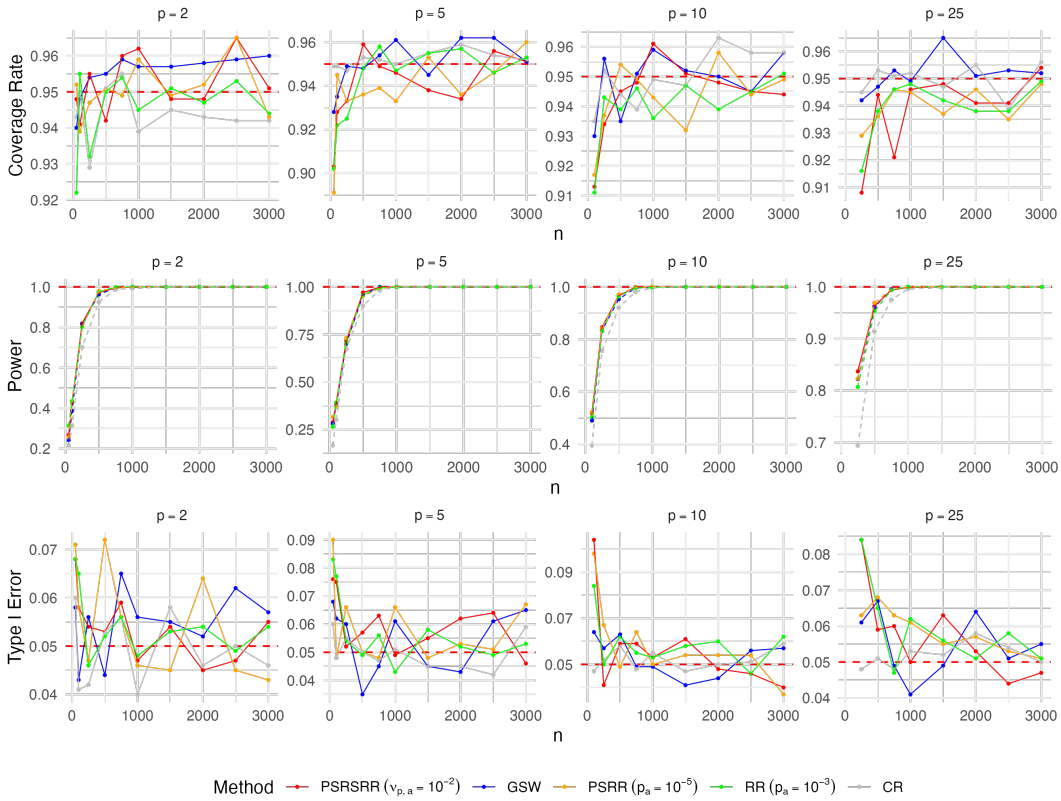


Figure 7: Comparison plots of Coverage Rate, Power and Type I Error under $R^2 = 0.2$.

The above plots are the comparison plots under the settings with $R^2 = 0.2$. We can tell from the Coverage Rate one that PSRSRR can achieve a satisfactory coverage rate, with values closely around the nominal level 95%. And PSRSRR has relatively better power when the number of covairates p is larger. As for the Type I Error, when sample size n increases, PSRSRR has error values slightly above or below the error bound 0.05. These results all show the validity of applying the theoretical results with the ground in uniformity of distribution to PSRSRR, thus non-directly verifying the near-uniformity of the distribution of the assignments generated by our proposed algorithm.

We can obtain similar conclusions from the following plots under the settings with $R^2 = 0.8$. In the Coverage Rate figure, GSW seems to be well above the nominal level 95%, indicating the fact that inference of GSW under settings with $R^2 = 0.8$ may tend to be conservative; while PSRSRR still has values closely around the nominal level, showing good inference performance. In Power, compared with settings under $R^2 = 0.2$ and $R^2 = 0.5$, all methods appear to have higher powers, due to the fact that the covariates can explain more variation in the simulation model. And among all methods, PSRSRR has an obviously higher power when p is relatively large. As for the Type I Error, when sample size n increases, PSRSRR has error values reasonably around 0.05.

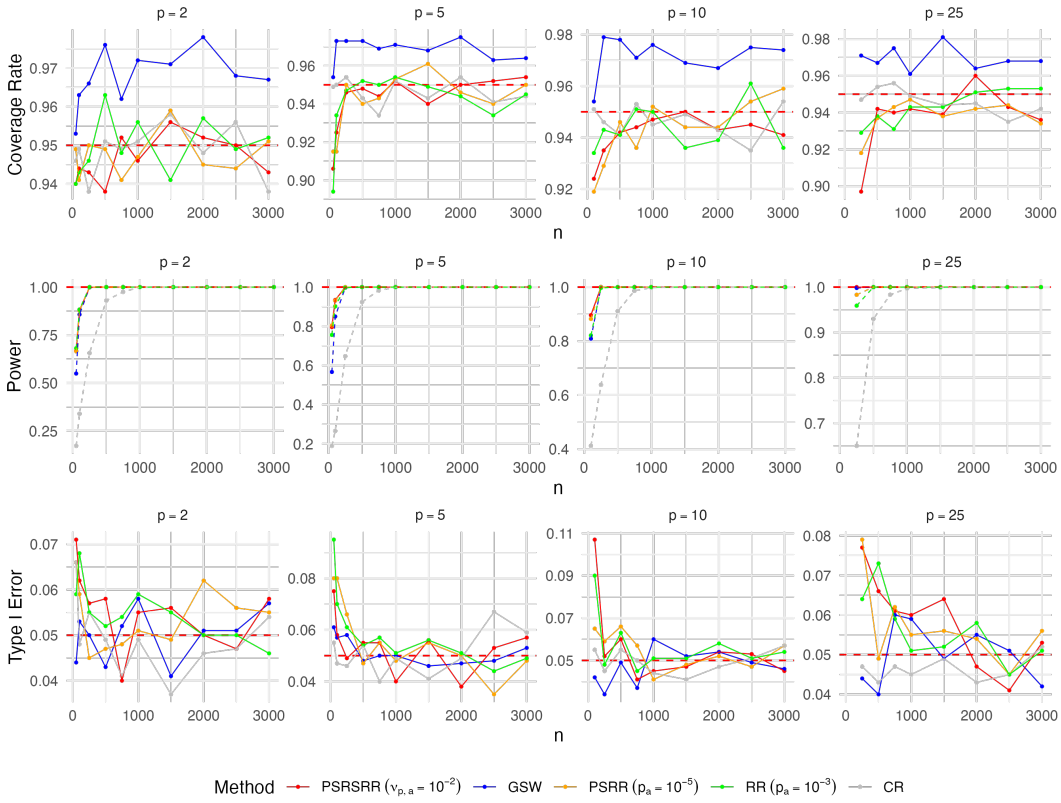


Figure 8: Comparison plots of Coverage Rate, Power and Type I Error under $R^2 = 0.8$.

B.4 More Details of Real Data Applications

Analysis of Reserpine data. In this dataset, the treatment group has 20 participants while the control group has 10. Similar to Zhu & Liu (2022), we include 8 important covariates to balance, and the threshold a is set as the $p_a = 0.001$ quantile of χ^2_8, i.e., $a = 0.86$ for all methods. Here, we exclude comparison with GSW as it could not have an exact treatment-control division as desired. And due to the small sample size, we also do not include the proposed method using $\nu_{p,a}$ for threshold selection, as this would only have correct interpretations in asymptotic scenarios.

We generate $10,000$ assignments, and obtain the following results. In Table 1, we compare the total time of generating these $10,000$ assignments, and in Figure 9, the empirical distributions of the standardized differences in each covariate mean are presented and the empirical percent reductions in variance (PRIVs) relative to CR are calculated. From these results, we can conclude that our proposed algorithm PSRSRR has achieved much faster sampling speed than the CR and much improved variance compared to the randomized design. Besides, the assignments generated using our proposed algorithm have a balance table for each covariate mostly within the recommended univariate balance thresholds $[-0.1, 0.1]$ (Austin, 2009) (illustrated with dashed lines).

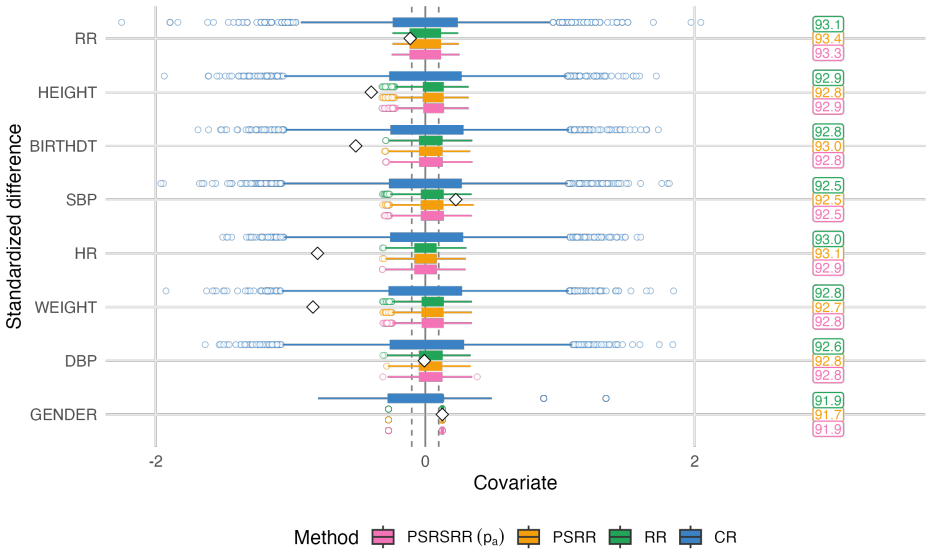


Figure 9: Box-plot of standardized differences in covariate means for Reserpine data. The diamonds indicates the standardized difference for the actual assignment in the experiment.

STAR data in Angrist et al. (2009) and its preprocessing. Student Achievement and Retention (STAR) Project is a randomized experimental evaluation of strategies designed to improve academic performance among college freshmen. Students in this experiment, except for those with a high school grade point average (GPA) in the upper quantile, were randomly assigned to one of three treatment groups or a control group. To keep the situation simple, similar to Li et al. (2018) and Wang & Li (2022), we keep only one treatment group, which received both additional mentoring services as well as incentives in the form of substantial cash awards for meeting a target GPA, to compare against the control group that was only eligible for standard university support services but nothing extra. And following their data preprocessing procedure, we discard students with missing data in the following covariates, or the first year GPA which is used as the observed outcome in their analysis, resulting in a treatment group of $n_1 = 118$ and control group of $n_0 = 856$. Considering the practical implications regarding the number of covariates given in Wang & Li (2022), we keep the first five covariates, i.e., high-school GPA, whether lives at home, gender, age and whether rarely puts off studying for tests, in our design stage.

Table 5: Covariates by tier in STAR data as in Li et al. (2018).

Tier	**Covariates**
Tier 1	High-school GPA
Tier 2	Whether lives at home, gender, age
	Whether rarely puts off studying for tests
Tier 3	Whether mother/father is a college graduate
	Whether mother/father is a high-school graduate
	Whether never puts off studying for tests
	Whether wants more than a bachelor degree
	Whether intends to finish in 4 years
	Whether plans to work while in school
	Whether at the first choice school, mother tongue

C LLM USAGE

We used ChatGPT-5 to improve the clarity of our writing by correcting grammar, refining sentence structure, and ensuring stylistic consistency.