

## 7 APPENDIX

### 7.1 Proof of Lemma 1

Suppose the distribution function of  $X_j$  is  $F_j(x)$ ,  $j = 1, \dots, k$ . And let  $F(x)$  denote the standard normal distribution function. From the weak convergence theorem (Hu, 2009), we get

$$\int e^{itx^2} dF_j(x) \rightarrow \int e^{itx^2} dF(x), \quad j = 1, \dots, k,$$

namely

$$E(e^{itx_j^2}) \rightarrow E(e^{itx^2}), \quad j = 1, \dots, k.$$

By the continuity theorem (Wu et al. 1979), we obtain

$$X_j^{2asy} \sim \chi^2(1), \quad j = 1, \dots, k.$$

Since  $X_1, \dots, X_k$  are mutually independent to each other, we have by Zhong (2010)

$$\sum_{j=1}^k X_j^{2asy} \sim \chi^2(k).$$

Therefore, the proof of Lemma 1 is completed.

### 7.2 Parameter Settings of Monte Carlo Simulation

In Monte Carlo simulations, it is beneficial to explore a wide range of parameter settings, such as various sample sizes and combinations of variances. The parameter settings of Monte Carlo Simulation are set as follows.

As for **Simulated Type 1 Error Probability of Algorithm 1-2**, firstly, let the nominal significance level be 5%, the number of loops are 5,000, and  $\mu = 0$ .

Secondly, as for 9 populations, we set bootstrap sample size as  $(N_1, N_2, N_3, N_4) = (30, 30, 40, 50, 50, 60, 70, 70), (40, 40, 50, 70, 70, 90, 100, 110, 120), (60, 60, 70, 80, 80, 90, 90, 100, 120), (80, 80, 80, 80, 90, 90, 90, 100, 100), (\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2, \sigma_7^2, \sigma_8^2, \sigma_9^2) = (1, 1, 1, 1, 1, 1, 1, 1, 1), (0.1, 0.3, 0.3, 0.5, 0.5, 0.7, 0.7, 0.9, 1.1, 1.1), (0.1, 0.3, 0.3, 0.5, 0.5, 0.7, 0.7, 0.9, 1.1, 1.1), (0.3, 0.3, 0.5, 0.5, 0.5, 0.7, 0.7, 0.8, 0.9, 1.1), (0.7, 0.7, 0.8, 0.8, 0.9, 0.9, 1.1, 1.1, 1.2), (0.5, 0.5, 0.6, 0.7, 0.7, 0.8, 0.9, 1.1), (0.7, 0.7, 0.8, 0.8, 0.9, 0.9, 1.1, 1.1, 1.1), (0.6, 0.6, 0.7, 0.8, 0.9, 1.2, 1.3, 1.4), (0.8, 0.8, 0.9, 0.9, 1.1, 1.1, 1.2, 1.2).$

For 10 populations, we set  $(N_1, N_2, N_3, N_4) = (40, 40, 40, 50, 50, 60, 60, 80, 80, 80), (40, 60, 60, 60, 60, 80, 80, 80, 90, 90), (60, 60, 70, 80, 90, 100, 110, 120, 120), (80, 80, 80, 80, 90, 90, 90, 100, 100), (\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2, \sigma_7^2, \sigma_8^2, \sigma_9^2) = (1, 1, 1, 1, 1, 1, 1, 1, 1), (0.1, 0.3, 0.3, 0.5, 0.5, 0.7, 0.7, 0.9, 1.1, 1.1, 1.3), (0.2, 0.4, 0.4, 0.6, 0.8, 0.8, 1.1, 1.2, 1.4), (0.3, 0.3, 0.5, 0.6, 0.7, 0.7, 0.9, 1.3, 1.5, 1.7), (0.4, 0.5, 0.6, 0.6, 0.8, 0.8, 1.0, 1.2, 1.4, 1.6), (0.5, 0.6, 0.6, 0.7, 0.8, 0.8, 1.3, 1.5, 1.9), (0.6, 0.7, 0.7, 0.9, 0.9, 1.2, 1.4, 1.6, 1.8), (0.7, 0.9, 0.9, 1.0, 1.2, 1.5, 1.5, 1.7, 1.7, 1.9), (0.8, 0.8, 0.8, 1.2, 1.2, 1.4, 1.4, 1.8, 1.8, 2).$

For 15 populations, we set  $(N_1, N_2, N_3, N_4) = (40, 40, 40, 50, 50, 50, 60, 60, 70, 70, 70, 80, 80, 80), (50, 50, 50, 60, 60, 60, 80, 80, 80, 90, 90, 90, 90, 90), (60, 60, 60, 70, 80, 80, 90, 90, 90, 100, 100, 100, 100, 100), (70, 70, 70, 70, 80, 80, 80, 80, 100, 100, 100, 120, 120, 120, 120), (\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2, \sigma_5^2, \sigma_6^2, \sigma_7^2, \sigma_8^2, \sigma_9^2) = (1, 1, 1, 1, 1, 1, 1, 1, 1), (0.1, 0.3, 0.3, 0.3, 0.5, 0.5, 0.5, 0.7, 0.7, 0.9, 1.1, 1.1, 1.3, 1.5, 1.5), (0.3, 0.3, 0.4, 0.4, 0.5, 0.5, 0.6, 0.7, 0.8, 0.8, 1.1, 1.2, 1.4, 1.4), (0.4, 0.4, 0.4, 0.6, 0.6, 0.6, 0.8, 0.8, 0.8, 0.9, 0.9, 1.1, 1.2, 1.4, 1.6), (0.5, 0.5, 0.5, 0.6, 0.6, 0.6, 0.8, 0.8, 0.8, 0.9, 0.9, 1.0, 1.2, 1.4, 1.6), (0.7, 0.7, 0.7, 0.8, 0.8, 0.8, 0.9, 0.9, 0.9, 0.9, 1.1, 1.2, 1.5, 1.8), (0.8, 0.8, 0.9, 0.9, 1.1, 1.2, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.2), (0.8, 0.9, 0.9, 0.9, 1.1, 1.2, 1.3, 1.4, 1.5, 1.5, 1.6, 1.7, 1.9), (1.1, 1.1, 1.2, 1.2, 1.3, 1.3, 1.4, 1.4, 1.5, 1.5, 1.5, 1.6, 1.7, 1.8, 2).$

As for **Simulated Power of Algorithm 1-2**. For 9 populations,

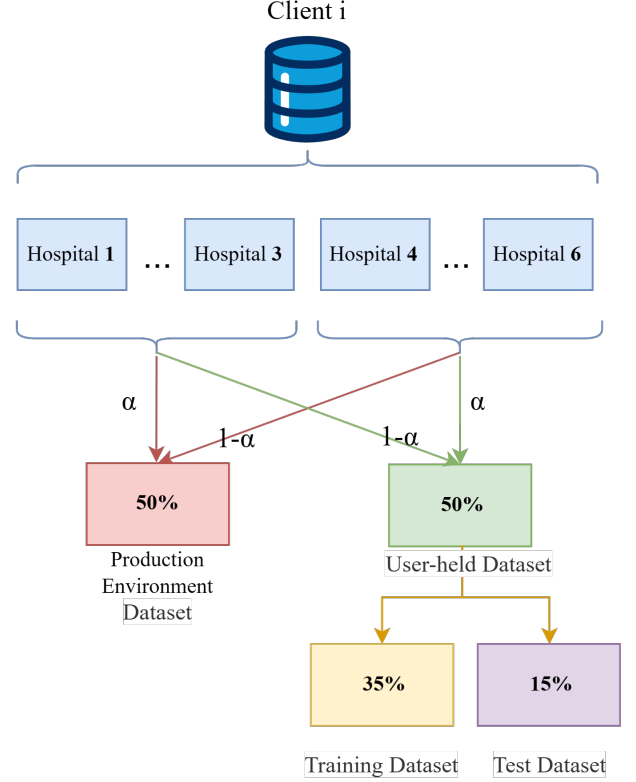


Figure 7: The framework of compare experiments

we set  $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7) = (0, 0, 0, 0, 0, 0, 0.1, 0.1, 0.1), (0, 0, 0, 0, 0, 0, 0.1, 0.2, 0.2), (0, 0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2), (0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3), (0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.3), (0, 0, 0, 0.1, 0.2, 0.2, 0.3, 0.4, 0.4), (V_1, V_2, V_3) = (0, 0.2, 0.2, 0.4, 0.6, 0.6, 0.8, 1, 1), (0.2, 0.2, 0.4, 0.4, 0.4, 0.6, 0.6, 0.7, 0.8), (0.4, 0.5, 0.6, 0.7, 0.8, 0.8, 0.9, 1, 1.1).$

For 10 populations, we set  $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7) = (0, 0, 0, 0, 0, 0, 0.1, 0.1, 0.1), (0, 0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2), (0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3), (0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.3), (0, 0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.4, 0.4, 0.5), (0, 0, 0.1, 0.2, 0.2, 0.3, 0.4, 0.5, 0.5), (V_1, V_2, V_3) = (0.1, 0.3, 0.3, 0.5, 0.7, 0.7, 0.9, 0.9, 1.1, 1.3), (0.2, 0.2, 0.4, 0.5, 0.6, 0.6, 0.8, 1.2, 1.4, 1.6), (0.4, 0.5, 0.5, 0.6, 0.7, 0.7, 0.9, 1.2, 1.4, 1.9).$

For 15 populations, we set  $(\delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \delta_7) = (0, 0, 0, 0, 0, 0, 0.1, 0.1, 0.1), (0, 0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2), (0, 0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3), (0, 0, 0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.3), (0, 0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.4, 0.4, 0.5), (0, 0.1, 0.1, 0.2, 0.2, 0.3, 0.4, 0.5, 0.5), (V_1, V_2, V_3) = (0.1, 0.3, 0.3, 0.5, 0.7, 0.7, 0.9, 0.9, 1.1, 1.3), (0.2, 0.2, 0.4, 0.5, 0.6, 0.6, 0.8, 1.2, 1.4, 1.6), (0.4, 0.5, 0.5, 0.6, 0.7, 0.7, 0.9, 1.2, 1.4, 1.9).$

$(V_1, V_2, V_3) = (0, 0.2, 0.2, 0.4, 0.4, 0.4, 0.6, 0.6, 0.8, 1.0, 1.0, 1.2, 1.4, 1.4), (0.3, 0.3, 0.3, 0.5, 0.5, 0.5, 0.7, 0.7, 0.7, 0.8, 0.8, 1.0, 1.1, 1.3, 1.5), (0.6, 0.6, 0.6, 0.7, 0.7, 0.7, 0.8, 0.8, 0.8, 0.8, 1.1, 1.1, 1.4, 1.7).$

Here,  $\delta$  represents the difference between the means of each client. As the performance differences of different clients' models

increase, the power value quickly approaches 1, indicating that our proposed method can promptly detect unfair phenomena.

### 7.3 Details of Compare experiments

**Data Bias:** To demonstrate that our approach yields more stable results than traditional methods in the face of uncertainty in the distribution of production environment datasets, we partitioned 50% of the data from each client's dataset to simulate the production environment dataset. The remaining 50% was further randomly divided into 35% and 15%, used to simulate the training and testing sets held by users. Figure 7 illustrates the dataset partitioning method we employed, capable of mimicking the distribution shift between production environment datasets and user-held datasets.

This partition method is based on an  $\alpha$  parameter, ranging from 0 to 0.5. Simulated datasets for each client are created by sampling  $\alpha$  proportion of data from the first three hospitals and  $(1 - \alpha)$  proportion from the last three. When  $\alpha$  is 0.5, the simulated and test datasets are completely i.i.d., and at  $\alpha = 0$ , they are fully non-i.i.d. Values between 0 and 0.5 indicate varying levels of distribution skew.

In Figures 8-10, detailed result charts are presented from comparative experiments conducted under the above seven parameter settings.

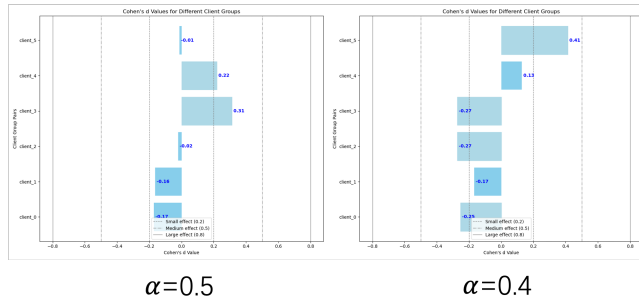


Figure 8:  $\alpha=0.4$  and  $0.5$

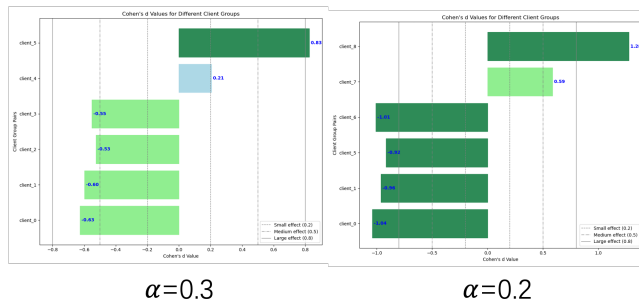


Figure 9:  $\alpha=0.2$  and  $0.3$

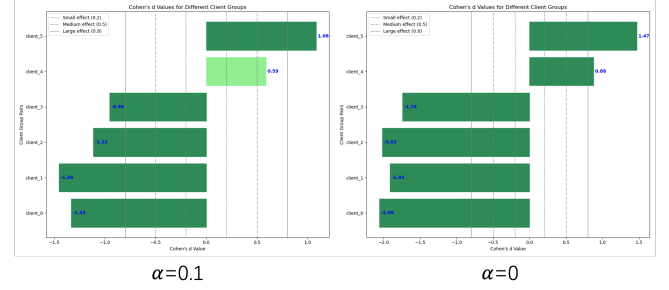


Figure 10:  $\alpha=0$  and  $0.1$