arXiv:2302.05549v2 [stat.ME] 3 Aug 2023

大规模因果推断的平衡方法

Sicheng Lin*
Snap Inc. 圣莫尼卡, 美国
slin@snap.com

Meng Xu*
Snap Inc. 圣莫尼卡, 美国
mxu@snap.com

Xi Zhang*Snap Inc. 圣莫尼卡，美国
xizhang@snap.com

Shih-Kang Chao 密苏里大学 哥伦比亚，密苏里州，美国
chaosh@missouri.edu

Ying-Kai Huang†
Realtor.com
圣克拉拉 , CA, USA
ying-kai.huang@move.com

Xiaolin Shi
Snap Inc. 圣莫尼卡, 美国
xiaolin@snap.com

摘要

借助现代软件和在线平台收集海量数据，当随机实验不可行时，大规模应用因果推断的需求日益增长。在观察性研究中，直接纳入协变量平衡的加权方法近年来越来越受到关注，可用于估计因果效应。这些方法减少了研究人员在倾向评分建模与平衡性检验之间反复迭代、直至获得满意协变量平衡结果所需的人工工作。然而，用于确定权重的传统求解器缺乏可扩展性，无法将此类方法应用于 Snap Inc. 等公司的大规模数据集。为应对这些局限并提高计算效率，本文针对两种平衡方法——熵平衡 [14] 和 MicroSynth [33]——提出了可扩展算法 DistEB 和 DistMS。这些求解器具有线性时间复杂度，并可便捷地在 Spark、Hive 等分布式计算框架中实现。我们研究了不同规模下平衡方法的性质，最大规模达到 100 万个处理组单位和 487 个协变量。我们发现，随着样本量增大，因果效应估计中的偏差和方差均显著降低。结果强调了在大规模数据集上应用平衡方法的重要性。我们将平衡方法与合成控制框架相结合，并在 Snap Inc. 部署了一个用于因果影响估计的端到端系统。

CCS 概念

• **计算方法** → **MapReduce 算法**； • **通用与参考领域** → **性能**；**估计**。

关键词

因果推断；观察性研究；大数据；协变量平衡；二次规划

*三位作者对本研究的贡献相同。
†作者在 Snap Inc. 工作期间参与了该论文的撰写。

允许免费为个人或课堂用途制作本作品全部或部分内容的数字或纸质副本，前提是这些副本不得用于营利或商业利益，也不得为此进行分发，并且副本须附带本声明及首页上的完整引文。对于本作品中由作者以外的其他人拥有的组件，必须尊重其版权。注明出处后可以摘录。若要以其他方式复制、再版、发布到服务器或分发至列表，须事先获得明确许可和/或支付费用。请向 permissions@acm.org 申请许可。*KDD '23，2023*年*8*月*6*日至*10*日，美国加利福尼亚州长滩© 2023 版权所有归所有者/作者所有。出版权已授权给 ACM。ACM ISBN 979-8-4007-0103-0/23/08...$15.00 https://doi.org/10.1145/3580305.3599778

ACM 引用格式：
Sicheng Lin、Meng Xu、Xi Zhang、Shih-Kang Chao、Ying-Kai Huang 和 Xi- aolin Shi。2023 年。《大规模因果推断的平衡方法》。载于《第*29*届*ACM SIGKDD*知识发现与数据挖掘大会（*KDD'23*）论文集》，*2023*年*8*月*6*日至*10*日，美国加利福尼亚州长滩。ACM，纽约，美国，12页。
https://doi.org/10.1145/3580305.3599778

1 引言

在大数据蓬勃发展的时代，我们正面临着可用信息和高维数据集爆炸式增长的局面。像 Snap Inc. 这样的公司会开展许多准实验研究，涉及数百万个观测值，旨在证明某项干预与数十项关键业务指标之间的因果关系。这些因果研究推动了决策流程的数据驱动趋势，并有望影响未来的业务战略决策。它们对公司至关重要，但由于在这些情况下我们无法对用户进行干预，因此无法通过 A/B 测试来验证。

在观察性研究中，由于处理条件并非随机分配给个体，处理组与对照组之间的部分协变量出现不平衡是很常见的。当协变量同时与处理分配和结果相关时，因这些协变量的不平衡可能会在处理效应估计中引入选择偏差，因果分析就会变得困难。为了调整协变量不平衡，倾向得分方面的开创性研究 [36] 指出了一个关键点：在给定倾向得分的条件下，处理分配与观测到的协变量相互独立。基于这一性质，人们提出了许多使用倾向得分的统计方法，包括倾向得分匹配 [36, 38],、倾向得分分层 [37], 、逆概率加权[18, 35], 等。在这些方法中，加权具有最吸引人的统计性质。首先，如果倾向得分模型设定正确，加权估计量就是相合的 [27]。其次，非参数倾向得分加权估计量是有效的[19]。相比之下，匹配和分层估计量不具备这些性质 [3, 27]。此外，加权方法不需要明确地建立结果模型，权重仅根据用户在实验设计阶段提供的协变量计算得出。一组权重可用于估计多个结果的因果效应。在这一点上，加权方法与 A/B 测试 [41] 相似，因为它们被视为更偏向实验设计的方法，而非实验分析方法。

尽管倾向得分加权方法具有坚实的理论基础，但在实际应用中，这种方法的性能对模型设定错误十分敏感。应用研究人员使用倾向得分加权方法时，往往会交替调整倾向得分模型并进行平衡性检验，直至得到满意的平衡结果 [23]。近年来，人们提出了熵平衡（以下简称 EB）[14], 、MicroSynth [33] 及其他几种方法 [24, 48, 50] ，通过直接平衡协变量而非建立倾向得分模型来解决这一问题。本文将这类方法称为**平衡方法**。总体而言，平衡方法通过实现更好的协变量平衡，效果优于倾向得分建模方法。研究还表明，在采用线性模型设定时，EB 具有双重稳健性 [49]。

我们依赖平衡方法来解决因果推断问题，因为一组权重可以重复用于多个业务指标，而且平衡方法通常比倾向得分建模方法具有更小的偏差。然而，我们发现，当前的平衡方法缺乏处理此类大数据所需的计算效率。在我们的应用中，个体数量和维度数分别可高达 1 亿和 1000 多个，整体数据规模可达数 TB。大多数现有因果推断软件包都是为数据集能够装入一台单机的问题而设计的。它们主要使用 Python 或 R 等对数据科学友好的语言实现，很少采用适用于大规模数据的分布式框架。Stata 软件包 Ebalance [16] 和 R 软件包 ebal [15] 是最早用于求解熵平衡权重的软件。以 R 实现的 microsynth [32] 软件包通过求解二次规划问题来计算 MicroSynth 权重。之后，谷歌开发的 Python 库经验校准（以下简称 EC）_ [44], 既可求解熵目标函数，也可求解二次目标函数，其计算速度比 ebal 快 22 倍。除了平衡方法之外，其他现代因果推断软件（例如微软的 EconML [4]和 DoWhy [42]、Uber 的 CausalML [10]、谷歌的 CausalImpact [7]）提供了丰富的因果分析工具，但受限于计算机内存。当前缺乏专为大规模因果推断任务打造的成熟软件 [46]。

为了扩展因果推断方法，Netflix 通过从原始数据集中提取汇总统计量，投入研发数据压缩策略 [45] 。他们发现，该策略能够对一大类线性模型 [45]的参数进行无损估计。另有一些研究（例如 [5, 30]）尝试利用并行计算扩展线性模型，包括普通最小二乘法、逻辑回归等。目前尚无研究致力于扩展平衡方法等更先进的因果推断方法。我们设计了一些算法，直接利用分布式计算框架来求解平衡方法中的优化问题。据我们所知，这是首项将平衡方法扩展到大型数据集的研究。

本文的贡献有两方面。首先，我们提出了两种分布式计算算法，分别命名为 DistEB 和 DistMS，用于求解两种著名的平衡方法：熵平衡和 MicroSynth。我们利用这两种方法中优化问题的对偶函数，使用 MapReduce 框架计算拉格朗日乘子的梯度，并应用迭代梯度求解器。我们的算法能够随观测值数量和协变量数量线性扩展。它可以估计多达 1 亿个个体、超过 1000 维数据的因果效应。其次，我们应用这些方法后发现，更大规模的数据集能显著降低因果效应估计中的偏差和方差，凸显了因果推断方法可扩展性的重要性。

本文其余部分的结构如下。第2节介绍因果推断的假设以及平衡方法中的优化问题。第3节详细介绍我们的算法 DistEB 和 DistMS。第4节评估熵平衡和 MicroSynth 的性能，以及我们算法的可扩展性。第5节讨论我们的算法在 Snap Inc. 的应用，并在第6节总结全文。

2 预备知识

2.1 基于观测数据的因果推断：设定与假设

考虑观测性研究中的以下因果推断问题。在从总体中抽取的一个大小为 N的随机样本中，每个个体 i接受二元处理分配 T_i。若个体 i被分配到处理组，则 $T_i = 1$；若个体 i属于对照组，则 $T_i = 0$。对照组和处理组中的个体数量分别为 N_0 和 N_1。对于每个个体 i，我们观察到一个结果Y_i以及一组协变量 $X_i = [X_{i1}, X_{i2}, \cdots \ X_{id}]^\top$，其中 d为协变量的维度。假设满足稳定单元处理值假设（SUTVA） [40], ，潜在结果$\{Y_{(0)\,i}, Y_{(1)\,i}\}$不依赖于其他个体的处理分配，即不存在网络效应。那么，观测到的结果可表示为 $Y_i = T_i Y_{(1)\,i} \ \ + (1 - T_i) Y_{(0)\,i}$。

每个个体的处理效应定义为 $\tau_i = Y_{(1 \ \ - \ Y)(0)\,i}$。本文重点关注处理组总体平均处理效应（PATT）^{i} 的估计，

$$\tau_{\text{PATT}} = \mathbb{E}[\tau_i | T_i = 1] = \mathbb{E}[Y_i^{(1)} | T_i = 1] - \mathbb{E}[Y_i^{(0)} | T_i = 1]. \quad (1)$$

第一个期望可以通过处理组中结果的样本均值直接估计。第二项描述了未观测到的反事实结果，需要利用对照组的数据进行估计。在随机对照试验中，治疗组是从总体中随机抽取的，因此我们可以使用对照组中结果的样本均值来估计第二项，因为 $\mathbb{E}[Y_{(0 \ \ | T_i = 1] = \mathbb{E}[Y^{)(0)}{}_i \ \ | T_i = 0]$。然而，在观察性研究中，处理分配并非随机，是否接受处理以及结果都可能与混杂因素有关，从而导致两个期望不相等。为纠正混杂因素造成的选择偏差，我们采用了 [36] 中的强可忽略性和重叠性假设。

假设 1（强可忽略性）：$(Y_{(0)} \ _{Y^{(1)}}) \perp T | X$。这意味着不存在不可观测混杂因素。

假设 2（重叠性）： $0 < P(T = 1|X) < 1$。它指出，每个个体接受每种处理条件的可能性都大于零。

2.2 加权方法：建模方法与平衡方法

加权方法源自逆概率加权（IPW），旨在为每个个体确定权重，以调整观测协变量之间的失衡。对于 PATT估计，只需为对照组中的个体分配权重。加权估计量的计算公式为：

$$\widehat{\tau_{\text{PATT}}} = \sum_{T_i=1} \frac{1}{N_1} Y_i - \sum_{T_i=0} w_i Y_i, \tag{2}$$

其中，w_i 是个体 i 的权重，权重之和满足 $\sum_{T_i=0} w_i = 1$，N_1 是处理组单位的数量。

计算权重通常有两种方法。建模方法通过对接受处理的概率进行建模来估计权重。平衡方法则直接最小化协变量不平衡和权重离散度。

2.2.1 建模方法。
建模方法基于倾向得分 [36]，其定义为在给定观测协变量的条件下，接受处理分配的概率。在观察性研究中，该量未知，需要通过在观测数据上拟合选择模型来估计。

拟合倾向得分模型的方法有很多。最常用的是逻辑回归。倾向得分模型也可以基于非线性或非参数方法，包括正则化逻辑回归、随机森林、提升回归、支持向量机、神经网络、多元自适应回归样条等。

为估计处理组平均处理效应（PATT），逆概率加权法（IPW）中的权重为

$$w_i^{\text{IPW}} = \frac{\hat{e}(X_i)(1-\hat{e}(X_i))^{-1}}{\sum_{T_i=0} \hat{e}(X_i)(1-\hat{e}(X_i))^{-1}}, \tag{3}$$

其中，$\hat{e}(X_i)$ 是接受处理的估计概率。当倾向得分相合估计时，逆概率加权估计量也相合。

提高逆概率加权估计量性能的一种方法是将倾向得分模型与结果模型结合，以实现双重稳健性（DR）。如果两个模型中任意一个设定正确，这种双重稳健程序就能产生相合估计 [34]。双重稳健估计量可以通过多种方式构造 [26]。 [49] 采用的一种带有残差偏差校正的回归估计修正形式为

$$\widehat{\tau_{\text{PATT}}^{\text{IPW,DR}}} = \sum_{T_i=1} \frac{1}{N_1}(Y_i - \hat{g}_0(X_i)) - \sum_{T_i=0} w_i^{\text{IPW}}(Y_i - \hat{g}_0(X_i)), \tag{4}$$

其中，$\hat{g}_0$ 是基于对照组数据建立的结果模型。

这种建模方法存在若干局限。首先，即使倾向得分模型设定正确，逆概率加权权重也只能在期望意义上平衡协变量，而不能在有限样本中实现平衡。在任何特定的有限样本中，协变量都可能难以平衡。其次，建模方法需要进行平衡性检验。如果经过逆概率加权调整后，治疗组之间的协变量仍不平衡，我们就需要修改倾向得分模型，以实现令人满意的平衡。这一调整过程涉及模型选择和超参数调优。对于大数据而言，寻找能够实现最佳平衡的模型可能代价高昂。由于第一项局限，在有限样本中实现最佳平衡的模型不一定是设定正确的模型，也不一定是偏差最小的模型（参见第4.3节的讨论）。这意味着，代价高昂的模型选择和调优过程并不能保证获得令人满意的性能。在我们的模拟数据中，公式4中的双重稳健逆概率加权估计量的性能远优于逆概率加权估计量。然而，当结果变量数量较多时，我们需要分别对它们建模。对于结果变量数量较多的大数据，双重稳健逆概率加权法可能代价高昂。

2.2.2 平衡方法。
提出平衡方法是为了克服建模方法的局限性。该方法无需显式建模处理概率，即可直接平衡协变量。熵平衡（EB） [14] 是一种开创性方法，它通过最小化估计权重与基准权重之间的 Kullback-Leibler 散度来求得权重。通常，基准权重设为均匀权重，并可从目标函数中省略。EB 权重通过求解以下优化问题计算得出

$$\begin{aligned} \min & \sum_{T_i=0} w_i \log(w_i) \\ \text{s.t.} & \sum_{T_i=0} w_i c_j(X_i) = \frac{1}{N_1} \sum_{T_i=1} c_j(X_i) \ \forall j, \\ & \sum_{T_i=0} w_i = 1, \\ & w_i \geq 0 \ \forall i, \end{aligned} \tag{5}$$

其中，函数 c_j 是协变量的矩函数。将上述优化问题的解代入方程 2，可得 PATT 的 EB 估计量：

$$\widehat{\tau_{\text{PATT}}^{\text{EB}}} = \sum_{T_i=1} \frac{1}{N_1} Y_i - \sum_{T_i=0} w_i^{\text{EB}} Y_i. \tag{6}$$

通过施加协变量平衡约束，EB 隐式拟合了一个线性回归模型。已证明 EB 估计量具有双重稳健性，并能达到效率界 [49]。这表明，如果真实的倾向得分模型或结果模型是线性的，PATT 就能得到相合估计。第4节的模拟结果显示，在 balance 和偏差方面，EB 的性能远优于 IPW；其性能与双重稳健 IPW 相近。

基于 EB 有若干种扩展。最小化问题的目标函数不一定采用 K-L 散度，也可以采用其他形式，包括带有 $D(w_i, 1)$ [8],[1] 的经验平衡校准权重、通过 $(w_i - 1/r)^2$ [50],限制极端权重出现可能性的稳定平衡权重，以及一种特殊的稳定平衡权重——MicroSynth，其形式为 $\frac{1}{2}(w_i - 1)^2$ [33]。如果满足优化问题 5 中的约束条件，这些扩展方法也具有双重稳健性 [49]。

[1] $D(x, x_0)$ 是固定 $x_0 \in \mathbb{R}$ 的距离度量，它关于 $x_0 \in \mathbb{R}$ 连续可微，关于 x 非负且严格凸。

另一种广受欢迎的平衡方法是协变量平衡倾向得分（CBPS）[24]。与 EB 及其扩展方法不同，CBPS 使用平衡权重显式拟合倾向得分模型。这种方法结合了建模方法和平衡方法。然而，我们的模拟表明，CBPS 可能无法实现令人满意的 balance，导致估计的处理组平均处理效应容易受到偏差影响。

值得一提的是，平衡方法并不局限于估计处理组平均处理效应（PATT）。近期一些研究将这一框架扩展至估计平均处理效应（ATE）[11, 25]。

近期出现了更先进的平衡方法。核平衡 [17]通过平衡核矩阵而非原始协变量，使处理组和对照组的协变量估计多变量密度近似相同。能量平衡 [21] 旨在平衡协变量的分布。然而，与熵平衡及其扩展方法相比，这些方法的计算成本高得多，因此在样本量和维度方面存在实际限制。

3 大规模平衡方法

本文的主要目标是为熵平衡和 MicroSynth 等平衡方法提供可扩展的解决方案。平衡方法中的凸优化问题可用不同形式的损失函数概括为公式 5。熵平衡的原始论文 [14] 及其实现 [16] 建议通过牛顿法求解优化问题，并在拉格朗日对偶问题上进行线搜索。牛顿法需要计算海森矩阵的逆矩阵，以确定参数更新方向。然而，在高维场景下，计算逆矩阵的成本可能很高。MicroSynth 也存在类似的计算问题：其目标函数为平方损失，优化问题则转化为同时具有等式和不等式约束的高维二次规划。我们的目标是提供一种随个体数量线性扩展的解决方案，并利用 Spark 或 Hadoop 等分布式计算框架。

3.1 熵平衡

为简化符号，以下各节考虑一阶矩函数$c_j(X_i) = X_i$，所得结果同样适用于高阶矩。为求解熵平衡优化问题，我们应用拉格朗日乘子 $\xi \in \mathbb{R}^d$ 以获得原始问题：

$$\min\left\{\mathcal{L}^{\text{EB}} = \sum_{T_i=0} w_i \log(w_i) + \xi^\top \Big(\sum_{T_i=0} w_i X_i - \tilde{X}\Big) + \xi_0\Big(\sum_{T_i=0} w_i - 1\Big)\right\}, \tag{7}$$

其中，$\tilde{X} = \frac{1}{N_1}\sum_{T_i=1} X_i$ 是协变量的处理组均值，ξ_0 是归一化约束上的拉格朗日乘子。求解$\partial \text{LEB}/\partial w_i = 0$ 可得到 w_i的解：

$$\hat{w}_i = w_i(\hat{\xi}) = \frac{\exp(-\hat{\xi}^\top X_i)}{\sum_{T_i=0}\exp(-\hat{\xi}^\top X_i)}. \tag{8}$$

注意，该表达式自动满足 $w_i \geq 0$ 的非负条件。给定 $\hat{}w_i$，拉格朗日对偶问题为

$$\min_{\xi\in\mathbb{R}^d}\left\{\mathcal{L}(\xi) = \log\left(\sum_{T_i=0}\exp(-\xi^\top X_i)\right) + \xi^\top \tilde{X}\right\}. \tag{9}$$

考虑到数据规模较大，无法装入单机的情况，我们寻求分布式方法来求解公式 9 中的优化问题。虽然实现大规模分布式 L-BFGS 求解器仍是一个选择，但我们采用梯度下降求解器，因为它简单且计算效率高。对偶问题的梯度可表示为

$$\frac{\partial \mathcal{L}}{\partial \xi} = \tilde{X} - \sum_{T_i=0}\frac{\exp(-\xi^\top X_i)}{\sum_{T_i=0}\exp(-\xi^\top X_i)} X_i = \tilde{X} - \sum_{T_i=0} w_i(\xi) X_i. \tag{10}$$

在梯度下降的每次更新步骤中，我们首先使用 MapReduce 计算权重 $w_i(\xi)$，然后通过第二个 MapReduce 步骤计算梯度。在实践中，使用动量 [31] 来加快收敛速度。一旦从对偶问题中求出 ξ，即可使用 MapReduce 轻松计算方程 8 中的最终权重。我们在算法 1 中介绍 DistEB。

算法 1 分布式熵平衡（Dist EB）

1：随机初始化 $\xi(0) \leftarrow \text{N}(\mathbf{0}, I_d)$，初始化学习率 $\alpha \leftarrow 0.01, \beta \leftarrow 0.9$。
2：**对于** $k = 1, \ldots, K$ **do** ▷ K：迭代次数
3：　分布式计算权重 $w_i(\xi(k-1))$，定义于方程-式 8。 ▷ MapReduce
4：　$\partial \text{L}/\partial \xi(k-1) = \tilde{X} - \sum_{T_i=0} w_i\big(\xi^{(k-1)}\big) X_i$ ▷ MapReduce
5：　$v(k) \leftarrow -\alpha \partial \text{升}/\partial \xi(k-1) + \beta * v(k-1)$ ▷ 动量步骤
6：　$\xi(k) \leftarrow \xi(k-1) + v(k)$ ▷ 梯度下降
7：**结束循环**
8：为所有控制组用户计算最终权重 w熵平衡$_i = w_i(\xi(k))$。 ▷ MapReduce

在每次更新步骤中，我们都会遍历整个数据集两次，因此总体时间复杂度为 $O(NdK)$，其中 N 是控制组单元的数量，d 是平衡协变量的维度，K 是迭代次数。DistEB 使用 PySpark 实现。

3.2 MicroSynth

MicroSynth 与 EB 的不同之处在于，它采用截断平方损失，因此解通常会落在约束集的边界上，并产生稀疏权重。此外，MicroSynth 对较大的权重施加比 EB 更大的惩罚，以避免异常权重 [33]。MicroSynth 的优化问题比 EB 更具挑战性，因为其中包含防止出现负权重的约束。为简化起见，我们在协变量向量 $X_{\prime i} = [1, X_{i1}, X_{i2}, \cdots\ X_{id}]^\top$ 中添加一个常数，以及相应的处理组均值 $\tilde{X}_{\prime} = \frac{1}{N_1}\sum_{T_i=1} X^{\prime i}$。设 $A = [X_{\prime 1}, X_{\prime 2}, \cdots\ X_{\prime N_0}]$ 为对照组中所有用户的协变量矩阵，$\boldsymbol{w} = [w_1, w_2, \cdots\ w_{N_0}]^\top$ 为权重向量，则方程 5 中的等式约束可表示为 $A\boldsymbol{w} = \tilde{X}_{\prime}$。我们像在 EB 中那样应用拉格朗日乘子 $\xi \in \mathbb{R}^{d+1}$，原始问题为

$$\min\left\{\mathcal{L}^{\text{MicroSynth}} = \frac{1}{2}||\boldsymbol{w} - \mathbf{1}||_2^2 + \psi_+(\boldsymbol{w}) + \xi^\top(A\boldsymbol{w} - \tilde{X}')\right\}, \tag{11}$$

其中，$\psi_+(\boldsymbol{w}) = \infty$（如果有 $w_i < 0$，则取该值），否则为 0。这是一个同时包含等式和不等式约束的高维二次规划问题。参数数量可达数亿，因此传统求解器（如内点法、投影梯度法、有效集法等）无法扩展至如此规模。我们发现，对偶上升法非常适用于大规模分布式凸优化问题。我们也考虑过另一种方案，即增广拉格朗日法，例如 ADMM [6]，，但发现对偶上升法在计算上更有效。

对偶上升法以迭代方式运行。在每次迭代中，我们使用梯度上升来更新拉格朗日对偶变量，然后通过最小化原始问题来恢复最优原始变量。对偶上升法利用了拉格朗日对偶问题始终具有凹性、无约束且维度相对较低这一事实，因此我们能够以分布式方式使用基于梯度的方法求解对偶问题。令$f(\boldsymbol{w}) = \frac{1}{2}||\boldsymbol{w} - \mathbf{1}||_2^2 + \psi_+(\boldsymbol{w})$，公式11的对偶问题为

$$\max_{\boldsymbol{\xi}} \left\{ \inf_{\boldsymbol{w}} \mathcal{L}^{\text{MicroSynth}} = -f^*(-A^\top \boldsymbol{\xi}) - \tilde{X}'^\top \boldsymbol{\xi} \right\}, \tag{12}$$

其中，f^* 是 f的凸共轭。由于 $f(\boldsymbol{w})$是严格凸的，f^* 是可微的。利用共轭函数的性质，公式12的导数为 $A\boldsymbol{w}^+ - \tilde{X}'$，其中$\boldsymbol{w}^+ = \text{argmin}_{\boldsymbol{w}}$ LMicroSynth$(\boldsymbol{w}, \boldsymbol{\xi})$
因此，对偶上升法从初始值 $\boldsymbol{\xi}^{(0)}$ 开始，并交替更新 $\boldsymbol{w}^{(k)}$ $\boldsymbol{\xi}^{(k)}$，共进行 K 次迭代：

$$\boldsymbol{w}^{(k+1)} = \text{argmin}_{\boldsymbol{w}} \left\{ \frac{1}{2}\|\boldsymbol{w} - \mathbf{1}\|_2^2 + \psi_+(\boldsymbol{w}) + \boldsymbol{\xi}^{(k)\top}(A\boldsymbol{w} - \tilde{X}') \right\}, \tag{13}$$

$$\boldsymbol{\xi}^{(k+1)} = \boldsymbol{\xi}^{(k)} + \alpha^{(k)}(A\boldsymbol{w}^{(k+1)} - \tilde{X}'). \tag{14}$$

根据微积分和投影，方程 13 中的问题有一个闭式解：

$$w_i^{(k+1)} = \begin{cases} 1 - X_i^\top \boldsymbol{\xi}^{(k)}, & \text{if } 1 - X_i^\top \boldsymbol{\xi}^{(k)} > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

请注意，由于权重 w_i 仅取决于用户 i的协变量，因此可以使用 MapReduce 轻松计算该解。在实践中，我们在对偶梯度上升步骤中加入动量，以加快收敛。DistMS 的完整细节见算法 2。

与 DistEB 类似，DistMS 在每次迭代中遍历整个数据集两次，总体时间复杂度为 $O(NdK)$。

Algorithm 2 Distributed MicroSynth (DistMS)

1: Initialize $\boldsymbol{\xi}^{(0)} \leftarrow \mathbf{0}$, momentum $\boldsymbol{v}^{(0)} \leftarrow \mathbf{0}$, learning rate $\alpha = 0.01, \beta^{(0)} = 1$.
2: **for** $k = 1, \ldots, K$ **do** ▷ K: the number of iterations
3: $w_i^{(k)} \leftarrow \begin{cases} 1 - X_i^\top \boldsymbol{\xi}^{(k-1)}, \text{ if } 1 - X_i^\top \boldsymbol{\xi}^{(k-1)} > 0 \\ 0, \text{ otherwise.} \end{cases}$ ▷ Distributed compute weights using Map
4: $\boldsymbol{v}^{(k)} \leftarrow \boldsymbol{\xi}^{(k-1)} + \alpha(A\boldsymbol{w}^{(k)} - \tilde{X}')$ ▷ Dual ascent step, MapReduce is used to compute $A\boldsymbol{w}^{(k)}$
5: $\beta^{(k)} \leftarrow \frac{1+\sqrt{1+4(\beta^{(k-1)})^2}}{2}$ ▷ Increase the 2nd learning rate
6: $\boldsymbol{\xi}^{(k)} \leftarrow \boldsymbol{v}^{(k)} + \left(\frac{\beta^{(k-1)}-1}{\beta^{(k)}}\right)(\boldsymbol{v}^{(k)} - \boldsymbol{v}^{(k-1)})$ ▷ Momentum step
7: **end for**
8: Compute final weights $w_i^{\text{MicroSynth}} = w_i^{(K)}$ for all control users. ▷ MapReduce

3.3 与其他实现的比较

与大多数现有的平衡方法实现一样，我们的实现利用拉格朗日对偶问题。不同之处在于求解对偶问题的方式。ebal 使用 Levenberg–Marquardt方法迭代求解该问题 [15]。其缺点是需要计算海森矩阵，而这在高维设置下不可行。EC 直接使用 scipy.optimize.fsolve 求解对偶问题 [44]。在底层，scipy.optimize.fsolve 通过前向差分近似计算雅可比矩阵，然后使用经过修改的 Powell方法迭代求解。与计算海森矩阵相比，这种方法更快。我们的方法甚至更快，因为它直接计算对偶变量的梯度该梯度直接根据数据计算，而不是通过前向差分步长计算，从而避免大量额外的函数求值。此外，我们的方法也更容易以分布式方式实现。我们将算法的可扩展性与其他实现进行了比较，发现我们的算法比现有的 EB 和 MicroSynth 内存实现更快，并且与 IPW 等数学上更简单的方法相比也具有竞争力。详细结果见第 4.5 节。

3.4 分布式与内存内

虽然我们的重点是为 EB 和 MicroSynth 设计分布式算法，但这些算法也可按串行方式运行并以内存方式实现。我们使用 Python 实现了 DistEB 和 DistMS 的内存版本，并利用 NumPy 快速的向量化运算。当数据集较小，能够装入单机时，内存版本的效率很高，因为内存操作的速度比机器间的数据交换快几个数量级。然而，并非总能获得内存容量达 TB 级的机器，此时就需要将工作负载分配到工作节点集群上。此外，数据规模较大时，将数据载入内存会很慢，并占用总运行时间的很大一部分。我们在补充材料（第 C 节）中提供了 EB 和 MicroSynth 的内存实现及 Spark 实现的关键代码片段和其他细节。

4 评估

4.1 数据生成过程

我们使用蒙特卡洛模拟评估不同加权方法在半合成数据上的性能。我们将 DistEB 与 DistMS、基于逻辑回归的逆概率加权（IPW）、[2] 协变量平衡倾向得分（CBPS）、[3] 以及

[2]其他 IPW 方法，例如基于机器学习的方法（如提升回归）[29]，也可用于计算倾向得分。然而，经过模型选择和超参数调优后，这些基于机器学习的方法在大数据场景下可能代价高昂。因此，我们不讨论这些方法的性能。[3]我们使用 Meta 的 Python 软件包 balance 实现协变量平衡倾向得分（CBPS）。我们重点关注过度识别的参数化 CBPS，它会同时拟合倾向得分模型并生成平衡权重。另有一种恰好识别的 CBPS 版本，只生成平衡权重，而不最大化处理选择预测能力。

这些方法与公式 4 中的双重稳健（DR）过程相结合，并使用简单的普通最小二乘回归拟合结果变量。为使数据生成过程尽可能贴近现实，我们没有使用完全模拟的正态分布数据，而是基于半合成数据进行模拟，使模拟反映 Snapchat 平台上为期 9 周的真实用户数据。我们将第 1 周至第 8 周定义为处理前时期，将第 9 周定义为处理后时期。

我们模拟了包含487个基线协变量的数据。其中包括456个连续协变量，用于表示第1周至第8周57项主要参与度指标的每周用户参与度。许多用户参与度指标呈偏态，且与零膨胀负二项分布相似。此外，我们还有31个二元协变量，代表第8周的主要用户属性。我们选取第9周的三项代表性用户参与度指标——活跃天数、应用打开次数和使用时长——作为结果变量。用户自行选择的处理分配指标为操作系统类型。操作系统为 iOS 的用户属于处理组，操作系统为 Android 的用户属于对照组。

我们假设所有487个协变量都可能影响处理分配和结果变量。我们重点考察结果模型 $\widehat{g_j}$ 和选择模型 f 的三种设定：线性模型、含交互作用的线性模型以及随机森林模型。详细的数据生成步骤见补充材料（算法3）。在线性设定下，结果模型和选择模型均设定正确。在此设定中，为避免模型设定错误偏差，逆概率加权法未加入正则化参数。在另外两种设定中，若仅纳入一阶矩，则使用 DistEB、DistMS、逆概率加权法和协变量平衡倾向得分时都会出现模型设定错误问题。我们对正则化参数进行调优，以实现最佳协变量平衡（详情见补充材料 D节）。超参数调优提升了逆概率加权法在这两种设定下的性能。这样做的目的是比较处理选择模型和结果模型均设定错误时几种加权方法的性能，因为在现实应用中，我们无法知道正确的模型。

我们在三种不同样本量下评估每种方法的性能：10K、100K 和 1 Million，以展示样本量增加时性能的提升。我们通过随机抽样，为每种样本量生成 100 个数据集。

4.2 MetaMetrics评估

我们从偏差、方差、balance 和权重稳定性等方面比较不同方法和样本量。

我们使用绝对平均偏差衡量每个结果变量 Y_j 的偏差：$AMB_j = \frac{1}{100}\left|\sum_{s=1}^{100} \widehat{\delta_{js}} - \delta_j\right|$，其中 $\widehat{\delta_{js}} = \frac{\widehat{\tau_{js\ PATT}}}{\sum_{T_i=1} - \widehat{\tau_{js\ PATT}}}$。由于不同结果变量的量级各不相同，我们不关注处理组平均处理效应，而是关注处理组与重新加权的对照组之间的百分比差异。由于我们没有引入任何额外的处理效应，真实百分比变化 δ_j 始终为 0。

我们用标准差衡量方差：$SD_j = sd(\widehat{\delta_{js}})$。值得注意的是，另一项广受欢迎的评估元指标根均方根误差 *(RMSE)* 可以分解为偏差和方差：$RMSE_j^2 = AMB_j^2 + SD_j^2$。我们不单独报告该指标。

我们可以通过协变量平衡性进一步了解偏差和方差。平衡性是指处理组与合成控制组在处理前协变量方面的相似程度。我们采用文献中针对加权方法的一些常用平衡性指标（[13, 22]），如下所示：标准化均值差（*SMD*）、方差比（*VR*）、重叠系数（*OVL*）、马氏平衡（*MB*）和柯尔莫哥洛夫–斯米尔诺夫距离（*KS*）。这些指标的详细定义见补充材料（B节）。我们汇总了所有维度上的平均平衡性。在这些平衡性指标中，原始指标的 SMD（X_j）和 MB 衡量均值处的平衡性。二次项的 SMD（X_j^2）、交互项的 SMD（X_jX_k）以及 VR 衡量二阶矩处的平衡性。OVL 和 KS 衡量整体分布的平衡性。

另一组诊断指标是权重稳定性。权重稳定性低或变异性高通常意味着少数个体获得了非常大的权重，从而导致 PATT 估计量不稳定。接下来 [9],我们重点关注以下用于衡量权重稳定性的元指标：归一化权重的标准差（*SD*）、权重极端程度（通过权重的最大值（最大值（*MAX*））和第99百分位数（第*99*百分位数（*P99*））衡量），以及有效样本量（*ESS*）。这些指标的详细定义见补充材料（B节）。ESS水平较低表明存在少数具有极端权重的个体。因此，该指标的值越高，表示权重稳定性越好。在我们的模拟中，我们使用 *ESS* 除以对照组中的个体总数来报告 ESS 比率。该比率在实践中可以作为独立的诊断指标使用 [9]。

4.3 不同数据生成过程中的偏差、方差、平衡性和权重稳定性

我们在三种数据生成情景下评估不同加权方法的性能，主要关注样本量为100K的情况。图1显示，无论是在模型设定正确的情况下（线性），还是在模型设定错误的情况下（带交互项的线性模型和随机森林），DistEB 和 DistMS 的偏差始终远小于 IPW 和 CBPS，方差也更小或相近。当选择模型和结果模型均存在模型设定错误时，DistEB 和 DistMS 仍然具有非常小的偏差。这意味着，通过实现协变量平衡，结果能够对模型设定错误保持稳健。在现实世界中，我们或多或少总会遇到模型设定错误问题，因为我们永远不知道真实值数据生成过程。DistEB 和 DistMS 在偏差和方差方面与 IPW+双重稳健和 CBPS+双重稳健的性能相近。DistMS 在所有情况下的方差都略小于 DistEB。双重稳健显著降低了 IPW 和 CBPS 的偏差与方差，但未能提升 DistEB 和 DistMS 的性能。这与相关理论一致，即 EB 和 MicroSynth 具有隐式双重稳健性 [49]。在我们的数据中，CBPS 的偏差比 IPW 更差，但方差更好。这与其他文献中的结论一致：在模型设定正确和模型设定错误两种情况下，与 IPW 相比，CBPS 的偏差性能可能更差，但方差性能更好 [49]。

DistEB 和 DistMS 的优异性能可归因于它们平衡协变量的能力。值得注意的是，我们在计算权重时没有纳入高阶矩。这两种平衡方法不仅在均值方面表现良好，

我们也在模拟中应用这种方法，发现结果与过度识别的 CBPS 非常相似。我们没有实现非参数 CBPS，因为它涉及大量计算。

在二阶矩和整体分布方面也表现良好。在所有情况下，DistEB 和 DistMS 的 balance 均超过文献中推荐的阈值。[4] IPW 经常低于该阈值。我们的模拟表明，即使轻微的不平衡也可能导致显著偏差。DistEB 和 DistMS 可通过改善 balance 来减轻偏差，而 CBPS 虽然旨在改善 balance，但在我们的数据中未能做到。

PATT 估计量的稳定性与权重稳定性有关。我们发现，尽管协变量平衡倾向得分在偏差和 balance 方面表现不佳，但其权重稳定性却最佳。理论上，熵平衡比 MicroSynth [33] 更容易产生极端权重。我们的模拟证实了这一说法。逆概率加权法在不同模拟中的表现不稳定，且逊于 DistMS 和协变量平衡倾向得分。

需要注意的是，逆概率加权法在模型选择和超参数调优方面成本都很高，而且无法保证选出的模型在期望意义上和特定有限样本中都具有最低偏差。原因在于，当倾向得分模型设定正确时，逆概率加权法只能平衡协变量，并在期望意义上实现无偏，但在任何特定有限样本中都无法保证这一点。我们的研究结果表明，当数据生成过程为线性时，在每次模拟中为实现最佳协变量平衡而调优超参数，并不能选出设定正确的模型或偏差最低的模型。

4.4 按样本量比较偏差、方差、平衡性和权重稳定性

我们重点针对由随机森林方法生成的数据，评估了不同方法在三种样本量下的性能：1万、10万和100万。图2显示，从1万增加到10万，偏差和 balance 都显著改善；但在大多数情况下，比较10万和100万时，两者的表现相近。随着样本量增加，方差和部分权重稳定性度量始终会降低。这意味着，较大的样本量对于获得稳定的估计量和检测微小变化是必要的。

4.5 运行时间与可扩展性

我们在 Google Cloud Platform 通用机器上评估了 DistEB 和 DistMS 的内存版本与分布式版本，详情见补充材料。

我们首先在单节点机器上评估内存版本，并比较我们的方法与现有开源软件包在不同数据点规模（10K、100K、100万）下的性能，包括 EB、Google 的 EC 软件包中的 MicroSynth、Meta 的 balance 软件包中的 CBPS [44], 以及 IPW。如表1所示，与现有的 EB 和 MicroSynth 软件包相比，我们的内存版本在运行时间方面的性能相当或更好。总体而言，MicroSynth 的表现优于 EB，这是因为它收敛更快，因此所需的迭代次数更少。MicroSynth 和 EB 的表现都显著优于 CBPS。采用 L2 正则化的 IPW 与 MicroSynth 的表现相当。不过，要获得最佳性能，IPW 需要进行参数调优，而参数调优所需的时间未计入运行时间。

[4]我们将文献中推荐的阈值列于补充材料中。详情请参见B节。

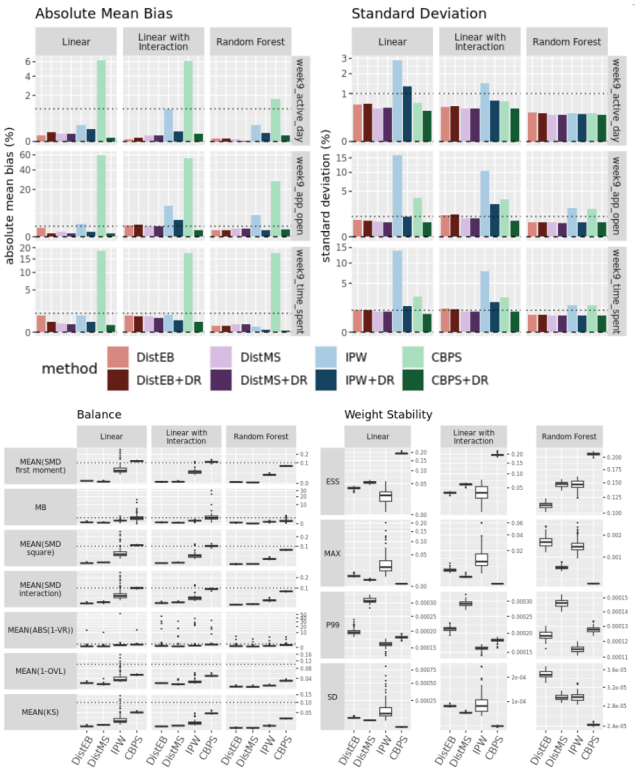

图1：在三个数据生成过程（DGP）中，加权方法在偏差、方差、balance 和权重稳定性方面的比较。每个 DGP 均生成 100 次模拟，样本量为 100,000。DistEB 和 DistMS 的表现始终优于 IPW 和 CBPS。左上图展示了所有模拟中的绝对平均偏差所衡量的偏差比较。右上图展示了由标准差衡量的方差。左下图和右下图分别展示了由七项平衡性指标衡量的平衡情况，以及由四项指标衡量的权重稳定性。绝对平均偏差和标准差的点线表示 1%，平衡性指标的点线表示文献中建议的阈值。为便于观察，Y轴采用平方根变换。

表1：单节点机器上的运行时间（分钟）

	数据规模		
	10K	100K	100万
DistEB（本研究）	0.248	1.375	5.597
熵平衡（经验校准）	2.259	6.066	124.317
DistMS（本研究）	0.05	0.142	0.995
MicroSynth（经验校准）	2.342	0.125	0.99
CBPS	6.411	44.628	433
IPW	0.015	0.213	0.783

对于 DistMS 的分布式版本，我们在包含 1000万数据集的不同数量工作节点的集群上运行算法，以展示其可扩展性。如图3所示，运行时间会随着工作节点数量的增加而缩短；当 Spark 中的并行任务数量接近文件分区数量时，运行时间趋于稳定。分布式计算也避免了在单机上耗时较长的数据加载过程。

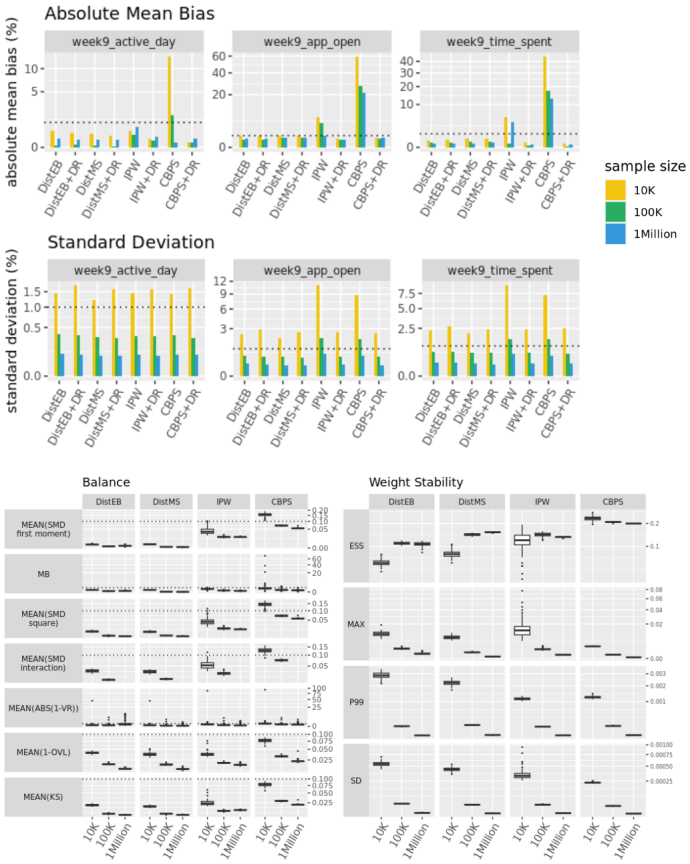

图2：三种样本量下各加权方法在偏差、方差、balance 和权重稳定性方面的比较。样本量越大，所有方法的表现都越好。上方和中间面板分别展示偏差和方差的比较。左下方和右下方面板分别展示由七项平衡性指标衡量的 balance，以及由四项指标衡量的权重稳定性。对于每种样本量，结果均基于 100 次模拟得出，数据生成过程采用随机森林。点线表示绝对平均偏差和标准差的 1%，以及文献中推荐的 balance 阈值。为便于观察，Y轴经过平方根变换。对于 100万样本量，由于计算开销较大，我们不计算均值（标准化均值差交互项）。

5 SNAP INC. 的应用

在 Snap Inc.，我们结合平衡方法与适用于多个处理组单位的合成控制框架 [1]，设计了一套端到端的观察性研究分析工作流，如图 4 所示。借助合成控制框架，我们对多项参与度指标和用户属性的处理前时间序列进行平衡，以减轻未测量混杂因素 [2] 所造成的偏差。我们从对照供体池中构建一个包含数百万用户的合成控制单元。该合成控制单元可作为处理组单位的反事实结果。综合考虑性能、

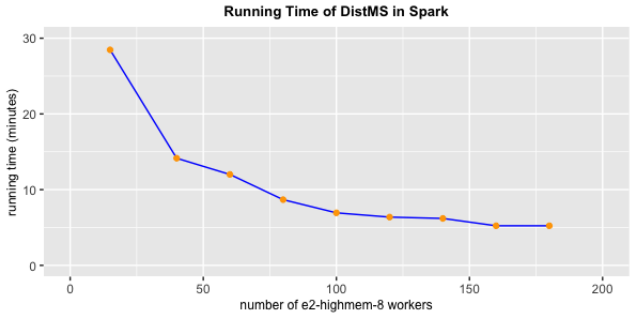

图3：DistMS 在 Spark 中处理 1000万数据集的运行时间

方差、权重稳定性和运行时间，DistMS 的表现优于 DistEB。我们将 DistMS 应用于生产工作流。我们的工具以处理组、对照供体池中的用户标识符以及处理开始日期为输入，然后使用 DistMS 为对照供体池中的用户生成一组权重。重新加权的对照组与处理组具有直接可比性。最后，我们提供一份包含累积差异和时间序列的汇总报告。遵循合成控制方法的最佳实践 [1],，我们会对所有平衡变量进行平衡性检验，包括处理前的用户参与度指标以及国家/地区、年龄、性别等用户属性；这与 A/B 测试中的样本比例不匹配 [12] 概念类似。我们将匹配期中的一周留作验证期，这与 A/B 测试中的 AA 期相仿。置信区间使用自助法计算。熟悉传统 A/B 测试平台的用户可以轻松解读观察性研究的结果，因为两者的报告界面相似，如图 5 和图 6 所示。

这种端到端工作流已应用于 Snap Inc. 的许多场景。工程师可以使用该工具，分析一些自然发生的事件或事故对用户参与度的影响，例如仅在某个地区发生的网络错误。数据科学家还可以设计准实验，例如在某所大学推出 Snapchat 营销活动，并使用合成控制法测量其影响。此外，在随机实验中，应用性能指标（例如应用打开延迟）可能会受到自选择偏差的影响，因为只有当用户打开应用时才会报告该指标，正如 [47] 中所指出的那样。通过针对相关协变量进行调整，可以直接应用平衡方法来减轻这些偏差。其他公司也会面临这些场景，并可以将该系统应用于自身的分析。

6 讨论

终点的系统依赖于某些假设。首先，与大多数使用观察性数据的因果推断方法类似，我们假设不存在未测量混杂因素，且具有重叠性。借助更大的数据集，我们能够平衡众多混杂因素，从而减轻未测量混杂因素造成的偏差。在实践中，我们关注关键参与度指标和用户属性的治疗前时间序列，默认只平衡一阶矩。在某些情况下，平衡更多治疗前协变量有助于减少不可观测混杂因素造成的偏差。处理前的验证期使我们能够检验是否存在不可观测混杂因素的有力证据。

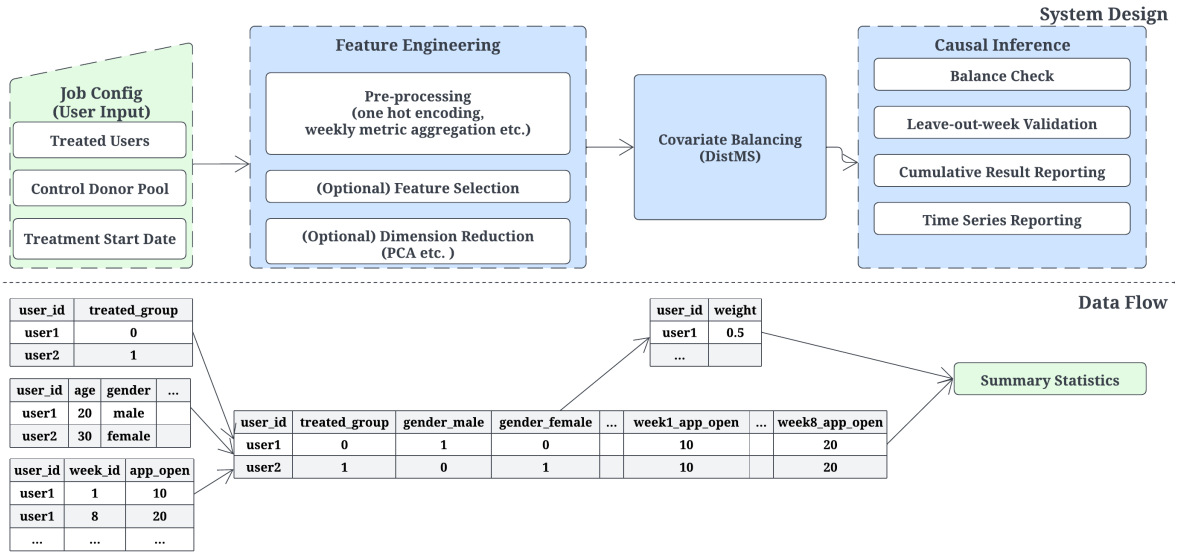

图4：Snap Inc. 观察性研究分析的端到端工作流

metric	pre-treatment matching	pre-treatment leave out week	post treatment cumulative
active_day	0.00% +/- 0.04% (p = 0.979)	0.00% +/- 0.03% (p = 0.746)	-0.02% +/- 0.04% (p = 0.212)
app_close_session_time	0.00% +/- 0.27% (p = 0.989)	0.16% +/- 0.27% (p = 0.249)	-0.29% +/- 0.26% (p = 0.030)
app_open	0.00% +/- 0.25% (p = 0.994)	0.09% +/- 0.24% (p = 0.469)	-0.28% +/- 0.24% (p = 0.024)

图5：汇总报告

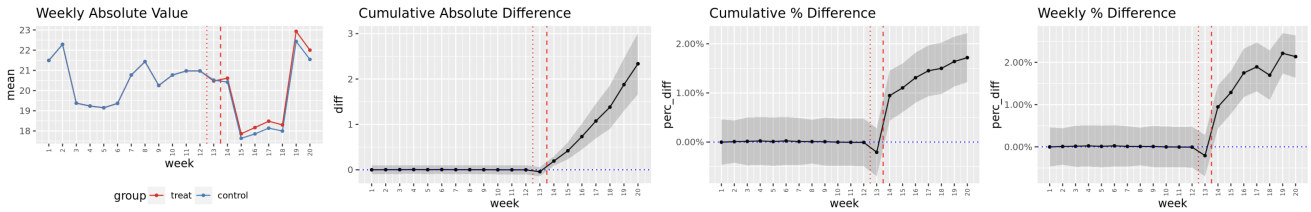

图6：时间序列报告：本例中有12周处理前时期、1周验证期和7周处理后时期。红色点线分隔处理前时期和验证期。红色虚线分隔验证期和处理后时期。

其次，我们采用合成控制框架的假设。该假设认为，良好的处理前拟合可以作为充分考虑所有可观测和不可观测因素的替代指标。匹配干预前结果时间序列有助于控制对多个不可观测因素的异质性响应。基于这一假设，只有处理后时期中无法根据处理前信息准确预测的不可观测混杂因素，才会使我们的结果产生偏差。

最后，我们的系统依赖于平衡方法背后的数据生成假设。该假设要求选择模型或结果模型中的真实数据生成过程为线性。我们发现，在实践中可以放宽这一假设。模拟结果表明，即使真实的数据生成过程是非线性的，与其他加权方法相比，我们的系统仍能提供偏差更小的估计。不过，在某些情况下，纳入高阶矩可能有助于减少模型设定错误造成的偏差 [20]。纳入大量平衡协变量和矩会带来两项挑战。首先，随着加入的约束越来越多，要找到能够平衡所有协变量和矩的解，变得越来越困难。其次，纳入许多平衡约束会增加计算成本。因此，在平衡过程之前采用特征选择或降维方法可能会有所帮助，这方面仍有待未来改进。

总之，我们的研究结果表明，直接平衡混杂因素的平衡方法优于建模方法，因为它能得到偏差和方差较低的因果效应估计量。这种方法也更加有效，因为它无需对多个结果变量建模，也无需调整超参数，因此能够以较低的计算成本获得双重稳健解。较大的样本量也能降低估计方差。为进一步改进这一方案，我们提出了可扩展求解器，其内存计算速度快于现有求解器，并且能够轻松集成到分布式计算框架中。我们的应用将平衡方法与合成控制框架相结合，以处理未测量混杂因素。Snap Inc. 已成功实施这一端到端工作流，利用大规模观察性数据估计因果效应。

参考文献

[1] Alberto Abadie. 2021. 使用合成控制：可行性、数据要求与方法学方面。《经济文献杂志》59, 2 (2021年6月), 391 425 10 1257 20191450 – 。 https://doi.org/ 。/jel. Alberto Abadie, Alexis Diamond, and Jens Hainmueller. 2010. 用于比较案例研究的合成控制方法：估计加利福尼亚州烟草控制计划的效应。J. Amer. Statist. As-soc. 105, 490 (2010), 493–505. https://doi.org/10.1198/jasa.2009.ap08746 arXiv:https://doi.org/ 10 1198 2009 08746 . /jasa. .ap Alberto Abadie and Guido Imbens. . 平均处理效应匹配估计量的大样本性质。《计量经济学》74, 1 (2006), 235–267. https://EconPapers.repec.org/RePEc:ecm:emetrp:v:74:y:2006:i:1:p:235-267[4] Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. 2019. EconML：用于基于机器学习的异质性处理效应估计的 Python 软件包。https://github.com/microsoft/EconML. 版本 0.x. [5] Benjamin Bluhm and Jannic Alexander Cutura. 2022. 大规模计量经济学：用 Spark 激活经济学中的大数据。《数据科学杂志》20, 3 (2022), 413–436. https://doi.org/ 10 6339 22 1035 . / -J DS Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. 通过交替方向乘子法进行分布式优化与统计学习。《机器学习基础与趋势》3, 1 (2011 1 122 3 11), – 。 http://dblp.uni-trier.de/db/journals/ftml/ftml .html#BoydPCPE Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. 2014. 使用贝叶斯结构时间序列模型推断因果影响。《应用统计年鉴》9 (2014), 247–274. https://research.google/pubs/ pub41854/ [8] Kwun Chuen Gary Chan, Sheung Chi Phillip Yam, and Zheng Zhang. 2016. 通过经验平衡校准加权对平均处理效应进行全局有效的非参数推断。《皇家统计学会杂志》B辑（统计方法学）78, 3 (2016), 673–700. http://www.jstor.org/stable/ 24775357[9] Ambarish Chattopadhyay, Christopher Hase, and José Zubizarreta. 2020. 实践中的加权平衡方法与建模方法。《医学统计学》39 09 2020 10 1002 ( ). https://doi.org/ . /sim.[10] Huigang Chen, Totte Harinen, Jeong-Yoon Lee, Mike Yung, and Zhenyu Zhao. 2020. Causalml：用于因果机器学习的 Python 软件包。arXiv 预印本 arXiv:2002.11631 (2020). [11] Rui Chen, Guanhua Chen, and Menggang Yu. 2022. 利用目标样本汇总信息进行因果泛化的熵平衡。(2022年1月)。[12] Aleksander Fabijan, Jayant Gupchup, Somit Gupta, Jeff Omhover, Wen Qin, Lukas Vermeer, and Pavel Dmitriev. 2019. 诊断在线受控实验中的样本比例不匹配。第25届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集 (2019). https://doi.org/ 10.1145/3292500.3330722[13] Jessica Franklin, Jeremy Rassen, Diana Ackermann, Dorothee Bartels, and Sebas-tian Schneeweiss. 2014. 因果效应队列研究中的协变量平衡指标。《医学统计学》 33 05 2014 10 1002 6058 ( ). https://doi.org/ . /sim. Jens Hainmueller. . 用于因果效应的熵平衡：在观察性研究中生成平衡样本的多变量重新加权方法。《政治分析》 20, 1 2012 25 46 10 1093 025 ( ), – 。 https://doi.org/ . /pan/mpr Jens Hainmueller. . ebal：通过熵重新加权创建平衡样本。https://CRAN.R-project.org/package=ebal R 软件包版本 0.1-8. [16] Jens Hainmueller and Yiqing Xu. 2013. ebalance：用于熵平衡的 Stata 软件包。《统计软件杂志》54, 7 (2013), 1–18. https://doi.org/10. 18637 054 07 /jss.v .i Chad Hazlett. . 核平衡：用于估计因果效应的灵活非参数加权程序。《中华统计学志》 30, 3 2020 1155 1189 26968924 ( ), – 。 https://www.jstor.org/stable/ Keisuke Hirano and Guido W. Imbens. . 使用倾向得分加权估计因果效应：右心导管插入术数据的应用。《卫生服务与结果研究方法学》2, 3-4 (2001), 259–278. https: //doi.org/ 10 1023 1020371312283 . /A: Keisuke Hirano, Guido Imbens, and Geert Ridder. 2003. 使用估计倾向得分有效估计平均处理效应。《计量经济学》 71 02 2003 1161 1189 10 1111 1468 0262 00442 ( ), – 。 https://doi.org/ . / - . Melody Y. Huang, Brian G. Vegetabile, Lane F. Burgette, Claude Setodji, and Beth Ann Griffin. 2021. 对因果估计而言，平衡高阶矩至关重要：为 Setodji 等人（2017）的结果提供进一步背景。https://doi.org/10.48550/ ARXIV.2107.03922[21] Jared D. Huling and Simon Mak. 2020. 协变量分布的能量平衡。https://doi.org/ 10 48550 2004 13962 . /ARXIV. . Kosuke Imai, Gary King, and Elizabeth Stuart. . 实验研究者与观察性研究者对因果推断的误解。《皇家统计学会杂志》A辑 171 (2007年12月)。[23] Kosuke Imai, Gary King, and Elizabeth Stuart. 2008. 实验研究者与观察性研究者对因果推断的误解。《皇家统计学会杂志》A辑 171, 第2部分 (2008), 481–502。 [24] Kosuke Imai 和 Marc Ratkovic。2014。协变量平衡倾向评分。《皇家统计学会会刊：B辑（统计方法学）》76, 1 (2014), 243–263。https://doi.org/10.1111/rssb.12027 arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/ 10 1111 12027 . /rssb。Kevin Josey、Seth Berkowitz、Debashis Ghosh 和 Sridharan Raghavan。2021。通过熵平衡迁移实验结果。《医学统计学》 40 05 2021 10 1002 9031 ( )。https://doi.org/ 。/sim。Joseph D. Y. Kang 和 Joseph L. Schafer。。揭开双重稳健性的神秘面纱：从不完整数据估计总体均值的替代策略比较。统计科学 22, 4 2007 10 1214 07 227 (nov)。 https://doi.org/ 。/ -sts Jared K Lunceford 和 Marie Davidian。。通过倾向得分进行分层和加权以估计因果处理效应：一项比较研究。《医学统计学》23 (2004)。[28] Andreas Markoulidakis、Khadijeh Taiyari、Peter Holmans、Philip Pallmann、Mon- ica Busse、Mark Godley 和 Beth Ann Griffin。2022。一篇比较不同协变量平衡方法的教程，并以评估青少年物质使用治疗项目的因果效应为例。《卫生服务与结果研究方法学》(05 2022 10 1007 10742 022 00280 0)。 https://doi.org/ 。/s - - - Daniel Mccaffrey、Greg Ridgeway 和 Andrew Morral。。使用提升回归估计倾向得分，以评估观察性研究中的因果效应。心理学方法 9 (01 2005), 403–25。https://doi.org/10.1037/1082-989X.9.4.403[30] Aurélien Ouattara、Matthieu Bulté、Wan-Ju Lin、Philipp Scholl、Benedikt Veit、Christos Ziakas、Florian Felice、Julien Virlogeux 和 George Dikos。2021。大数据上的可扩展计量经济学——Spark 上的逻辑回归。arXiv 电子印本，文章 arXiv:2106.10341 (2021年6月), arXiv:2106.10341 页。 https://doi.org/10. 48550 2106 10341 2106 10341 [ ] /arXiv. 。arXiv: 。 stat.CO Ning Qian。。关于梯度下降学习算法中的动量项。《神经网络》12, 1 (1999), 145–151。https://doi.org/10.1016/S0893-6080(98) 00116-6[32] Michael W. Robbins 和 Steven Davenport。2021。microsynth：R 中用于细分数据和微观层级数据的合成控制方法。《统计软件杂志》 97, 2 2021 1 31 10 18637 097 02 ( ), – 。 https://doi.org/ 。/jss.v .i Michael W. Robbins、Jessica Saunders 和 Beau Kilmer。。基于高维微观层级数据的合成控制方法框架：评估社区特定犯罪干预。《美国统计学会杂志》112, 517 (2017), 109–126。https://doi.org/10.1080/01621459.2016.1213634 arXiv:https://doi.org/ 10 1080 01621459 2016 1213634 . / 。. James M. Robins 和 Andrea Rotnitzky。。含缺失数据的多元回归模型中的半参数效率。《美国统计学会杂志》90, 429 1995 122 129 ( ), – 。http://www.jstor.org/stable/ 2291135 [35] James M. Robins、Andrea Rotnitzky 和 Lue Ping Zhao。1994。当某些回归变量并非总能观测到时，回归系数的估计。《美国统计学会杂志》89, 427 (1994), 846–866。https://doi.org/10.1080/01621459. 1994 10476818 10 1080 01621459 1994 10476818 . [36] arXiv:https://doi.org/ 。/ 。. Paul R. Rosenbaum 和 Donald B. Rubin。。倾向得分在观察性研究因果效应分析中的核心作用。《生物统计学》70, 1 (04 1983), 41–55。 https://doi.org/10.1093/biomet/70.1.41 arXiv:https://academic.oup.com/biomet/article-pdf/70/1/41/662954/70-1-41.pdf [37] Paul R. Rosenbaum 和 Donald B. Rubin。1984。使用倾向得分子分类减少观察性研究中的偏差。《美国统计学会杂志》79, 387 (1984), 516–524。 https://doi.org/10.1080/01621459.1984.10478078 arXiv:https://www.tandfonline.com/doi/pdf/ 10 1080 01621459 1984 10478078 . / 。. Paul R. Rosenbaum 和 Donald B. Rubin。。利用纳入倾向得分的多元匹配抽样方法构建对照组。《美国统计学家》39, 1 1985 33 38 10 1080 00031305 1985 10479383 ( ), – 。 https://doi.org/ 。/ 。. arXiv:https://www.tandfonline.com/doi/pdf/ 10 1080 00031305 1985 10479383 . / 。. Donald Rubin。。使用倾向得分帮助设计观察性研究：烟草诉讼中的应用。《卫生服务与结果研究方法学》 2 12 2001 169 188 10 1023 1020363010465 ( ), – 。 https://doi.org/ 。/A: Donald B. Rubin。。实验数据的随机化分析：费 Fisher 随机化检验评论。《美国统计学会杂志》75, 371 (1980), 591–593。http://www.jstor.org/stable/ 2287653 [41] Donald B. Rubin。。为实现客观的因果推断，设计胜过分析。《应用统计年刊》 2, 3 2008 10 1214 08 187 (sep)。 https://doi.org/ 。/ -aoas Amit Sharma 和 Emre Kiciman。。DoWhy：端到端因果推断库。arXiv 预印本 arXiv:2011.04216 (2020)。[43] Elizabeth A. Stuart。2010。因果推断的匹配方法：综述与展望。统计科学 25, 1 (2010), 1–21。http://www.jstor.org/stable/ 41058994 [44] Xiaojing Wang、Jingang Miao 和 Yunting Sun。2019。用于经验校准的 Python 库。arXiv 预印本 arXiv:1906.11920 (2019)。[45] Jeffrey Wong、Eskil Forsell、Randall Lewis、Tobias Mao 和 Matthew Wardrop。2021。只压缩一次：用于估计线性模型的最优数据压缩。arXiv 预印本 arXiv:2102.11297 (2021)。[46] Jeffrey C Wong。2020。计算因果推断。arXiv 预印本 arXiv:2007.10979 (2020)。

[47] Yuxiang Xie、Meng Xu、Evan Chow 和 Xiaolin Shi。2020。如何衡量你的应用：在线受控实验中衡量应用性能的若干陷阱及补救方法。第*14*届*ACM*网络搜索与数据挖掘国际会议论文集(2020)。[48] Qingyuan Zhao。2019。通过定制损失函数实现协变量平衡倾向评分。《统计学年鉴》47, 2 (2019), 965–993。https://doi.org/10.1214/ 18-AOS1698[49] Qingyuan Zhao 和 Daniel Percival。2017。熵平衡具有双重稳健性。因果推断期刊5, 1 (2017), 20160010。[50] José Zubizarreta。2015。用于平衡协变量、以估计不完整结局数据的稳定权重。美国统计学会杂志110 (04 2015), 0–0。https: // doi.org/10.1080/01621459.2015.1023805

A 数据模拟详情

A.1 生成半合成数据的步骤

算法3模拟半合成数据

1：从总体中随机选取 2N ($N = 10Million$) 名用户，并将他们平均分为训练数据集和测试数据集。2：使用训练数据集训练回归模型 $\widehat{g_j}(X_i)$，以预测结果指标 Y_{ij}；训练分类模型 $\widehat{f}(X_i)$，以预测处理分配。3：模拟结果指标：生成预测结果 $\widehat{Y_{ij}} = \widehat{g_j}(X_i) + \hat{\epsilon}_{ij}$，其中 $\hat{\epsilon}_{ij}$ 从第一个用于训练的 N 的拟合误差中有放回地抽样得到。4：模拟处理分配：为每位用户生成倾向得分$Prob(T_i = 1) = \widehat{f}(X_i)$。根据用户预测的处理状态，将最后的 N 名用户分配到处理组或对照组；该状态由一个 $Bernoulli(p)$ 确定，其$p = Prob(T_i = 1)$ 为 $= \widehat{f}(X_i)$。

A.2 规格详情

A.2.1 规格 *1*：线性模型。

- 结果模型采用线性回归拟合，包含 487 个协变量。
- 处理选择模型采用二元逻辑回归拟合，包含 487 个协变量。

A.2.2 规格 *2*：含交互作用的线性模型。我们将一个用户属性（即用户画像）与所有其他协变量进行交互。新的特征集包含 1458 个协变量。

- 结果模型采用线性回归拟合，包含 1458 个协变量。
- 处理选择模型采用二元逻辑回归拟合，包含 1458 个协变量。

A.2.3 规格 *3*. 随机森林。

- 结果模型使用随机森林回归器拟合，包含 487 个协变量。树的数量为 10。最大深度为 10。
- 处理选择模型使用随机森林分类器拟合，基于 487 个协变量。树的数量为 10，最大深度为 15。

B balance 和权重稳定性度量详情

标准化均值差 ff 差异 (SMD)：$SMD_d = \frac{|\Delta_d|}{\sqrt{(s_{d1}^2 + s_{d0}^{w2})/2}}$，其中 Δ_d 是处理组与重新加权的对照组之间协变量均值的差异，s_{d1}^2 是处理组中 X_d 的样本方差，s_{d0}^{w2} 是重新加权的对照组中 X_d 的加权样本方差。为总结所有维度的性能，我们进一步计算 $SMD_{first\ moment}$、SMD_{square} 和 $SMD_{interaction}$，分别表示所有协变量、连续变量的所有二次项以及所有可能交互项的平均 SMD。尽管我们在加权过程中未加入二次项和交互项，但我们仍关注不同方法在这些项上的 balance 性能。大多数研究者认为，当 SMD 低于 0.1 时即可实现 balance。这是一个任意设定的阈值。

马氏平衡（MB）：$MB = \sqrt{(X_1 - X_0^w)^\top \sum^{-1} (X_1 - X_0^w)}$，其中 X_1 是处理组中协变量均值的向量，X_0^w）是重新加权的对照组中加权协变量均值的向量，$\sum^{-1}$ 是协变量的样本方差-协方差矩阵。马氏距离为 1 或更低，表明该点恰好位于基准点之中。[5]

“方差比（VR）”：$VR_d = \frac{s_{d0}^{w\ 2}}{s_{d1}^2}$。数值越接近 1，表示二阶中心矩的 balance 越好。为了使其能与其他平衡性指标进行比较，我们通常考虑 $|1 - VR_d|$。为了总结所有维度的性能，我们计算 $|1 - VR_d|$，即所有维度上 $|1 - VR_d|$ 的平均值。若 balance 良好，方差比应介于 0.5 和 2 [39, 43] 之间。在我们的模拟中，$|1 - VR_d|$ 使用 0.5。

重叠系数（OVL）是两个组的协变量分布之间重叠部分所占的比例。其计算方法是求两条曲线中较低者下方的面积。

OVL 值始终介于 0 和 1 之间。越接近 1，表示 balance 越好。为了使其能与其他指标进行比较，我们关注 $1 - OVL$。每个维度都有 $1 - OVL_d$。为了总结所有维度的性能，我们计算 $1 - OVL_d$，即所有维度上 $1 - OVL_d$ 的平均值。文献中没有推荐阈值。我们对 $1 - OVL_d$ 使用 0.1。

柯尔莫哥洛夫–斯米尔诺夫距离（KS）是两个累积分布之间的最大垂直距离。一些文献将 0.1 作为 KS 的 balance 阈值 [28]。

归一化权重的标准差（SD）：$SD = sd(w_{i0})$，其中 $\sum(w_{i0}) = 1$。这是基于矩的变异性度量。数值越低，权重稳定性越好。

权重极端程度：通过权重的最大值（最大值）和第99百分位数（第*99*百分位数）衡量。数值越低，权重稳定性越好。

有效样本量（ESS）：$ESS = \frac{(\sum w_{i0})^2}{\sum w_{i0}^2}$。数值越高，权重稳定性越好。

C 实验的更多细节

此处提供了 EB 和 MicroSynth 的内存实现与 Spark 实现的关键代码片段：https://gist.github.com/xizzzz/02caadbaf760510e267cc1fa9d3a7971

[5] https://www.theinformationlab.co.uk/2017/05/26/mahalanobis-distance/

C.0.1 学习率。 我们将学习率初始化为 0.01，并在检测到振荡时加入衰减。

C.0.2 停止条件。 我们通过启发式方法选择一个阈值，使模型达到理想性能，并将其作为停止条件。我们使用权重总和与目标值之间的差异，具体如下： 1×10^{-4}。

C.0.3 训练机器类型。 实验中，我们使用谷歌云平台的通用型机器。[6] 我们选择不同的机器类型，以满足不同数据规模的计算需求。镜像版本为 2.0.29-Debian 10。以下表格列出了机器类型。数据规模表示每个数据集中的行数。

数据规模	机器类型	内存（GB）
10K	n1-standard-8	30
100K	n1-standard-16	60
1M	n2-highmem-32	256
10M	e2-highmem-8	64 × 工作节点数量

D 逆概率加权超参数调优

在第4节中，我们使用网格搜索来调优逆概率加权法中逻辑回归的超参数。我们调优 3 个超参数：交叉拟合的折数 $k \in \{0, 3, 5\}$、正则化参数 $C \in \{0.0001, 0.0003, 0.001, 0.003, 0.01, 0.03\}$，以及 l1_比率 $l \in \{0, 0.25, 0.5, 0.75, 1\}$。对于每种模拟设置，我们选择标准化均值差最小的超参数。[6]

[6] https://cloud.google.com/compute/docs/general-purpose-machines