感知掉队者的编码多项式聚合

Xi Zhong[1],JörgKliewer[2] 和 Mingyue Ji[1]1电气与计算机工程系，佛罗里达大学，美国佛罗里达州盖恩斯维尔 电子邮件：{xi.zhong,mingyueji}@ufl.edu[2]电气与计算机工程系，新泽西理工学院，美国新泽西州纽瓦克 电子邮件：jkliewer@njit.edu

arXiv:2602.03074v1 [cs.IT] 3 Feb 2026

摘要——**分布式计算系统中的编码多项式聚合（CPA）使主节点无需单独解码每一项，即可直接恢复多项式计算的加权聚合，从而减少所需的工作者响应数量。然而，现有 CPA 方案仅适用于无法容忍掉队者的理想化场景。本文将 CPA 扩展至具有预先指定非掉队模式的掉队感知分布式计算系统，其中要求对给定的一组允许的非掉队者集合实现精确恢复。我们的主要结果表明，恢复目标聚合所需的工作者响应数量少于基于单独解码的多项式码所需的数量；可行性则由非掉队模式的交集结构决定。具体而言，我们建立了掉队感知 CPA 实现精确恢复的充要条件。我们确定了一个足以保证精确恢复的交集大小阈值。当允许的非掉队者集合数量足够大时，我们进一步证明，从一般意义上说，该阈值也是必要的。我们还给出了显式构造：只要交集大小超过推导出的阈值，即可构造可行的 CPA 方案。最后，仿真验证了我们的理论结果，展示了可行性在预测的交集阈值处发生急剧转变。**

I. 引言

分布式计算通过将计算任务拆分到多个工作节点，并在主节点汇总其响应，实现大规模数据处理。编码技术已成为提升分布式计算系统可靠性和效率的有力工具，尤其是在存在掉队者的情况下；这些节点无法在合理时间内返回响应。

对于矩阵乘法和多项式计算任务，大量已有研究采用基于多项式的编码技术实现精确恢复，其中包括最大距离可分（MDS）编码计算 [1], [2],多项式码 [3], MatDot 和 PolyDot编码 [4], 纠缠多项式码 [5], [6], ，以及它们在数值稳定和异构场景中的扩展 [7], [8]。拉格朗日编码计算 [9] 通过将子计算编码为拉格朗日多项式的求值，进一步降低了解码复杂度。除精确恢复之外，还有一些研究探讨了在数值或概率保证下针对一般函数的近似编码计算 [10]–[12]。

大多数现有编码计算方案的一个共同特点是依赖单独解码，即主节点先对所有子计算逐一解码，再计算所需结果。此外，这些方案通常设有恢复阈值，并旨在容忍给定规模的所有掉队者集合。一旦非掉队工作节点的数量超过恢复阈值，无论哪些工作节点返回结果，都能保证精确恢复。

尽管对任意掉队者集合具备鲁棒性足以保证精确恢复，但在实践中未必需要如此。在许多分布式系统中，掉队行为并非完全随机，某些非掉队者集合出现的频率远高于其他集合。这一观察促使人们在编码计算中放宽最坏情况下的掉队鲁棒性，转而利用掉队行为中的统计或结构规律。例如， [12] 中的作者研究了概率掉队模型，并建立了高概率恢复保证。此外，对于涉及聚合的计算任务，通过单独解码来恢复每个子计算会引入不必要的冗余。这一观察促使此前的研究将目标从单个计算转向聚合输出。梯度编码 [13] 通过在工作节点之间引入冗余，专注于恢复梯度之和。另一项研究考察了线性可分计算 [14], ，其目标是精确恢复任意计算的加权聚合。在我们的同期工作 [15], 中，我们提出了编码多项式聚合（CPA），目标是计算多项式计算的加权和。通过利用聚合结构和多项式的代数性质，我们刻画了无需单独解码即可实现精确恢复所需的工作者响应数量。

然而， [15] 中的分析局限于一种理想化情形，即假设所有工作节点都会响应，也就是说，不考虑掉队容忍。在实际中，掉队者不可避免，而且返回结果的工作节点子集可能随时间变化。此外，要求在所有可能的掉队者集合下实现精确恢复可能过于保守，因为实际系统往往具有一定规律性，某些工作节点比其他节点更可靠。

本文将 CPA 扩展到具备掉队感知能力的分布式计算系统，并考虑预先指定的非掉队模式，其中要求针对给定的一组允许的非掉队集合实现精确恢复。我们证明，相较于基于单独解码的多项式码，实现所需聚合的精确恢复所需的工作者响应更少；此外，可行性由预先指定的非掉队模式的交集结构决定。

主要贡献总结如下。

1) 我们建立了掉队感知CPA中精确恢复的充要条件。
2) 我们确定了一个阈值，该阈值保证在预先指定的非掉队模式的交集大小上实现精确恢复。此外，当可容许的非掉队者集合数量足够大时，我们证明该阈值在一般意义上是必要的。3) 只要非掉队模式的交集大小超过所推导的阈值，我们就给出可实现精确恢复的显式CPA方案。
4) 仿真表明，在预测的交集阈值处存在急剧的可行性转变，从而印证了理论结果。

II. 问题表述

A. 预先指定的非掉队模式下的 *CPA*

我们考虑一个由主节点和一组 N 工作节点组成的分布式计算系统，工作节点的索引为 $[N]=\{0,1,\ldots,N-1\}$。给定 K 个数据矩阵 $\boldsymbol{X}_k \in \mathbb{C}^{q\times v}$（其中 $k\in[K]$），一个作用于每个数据矩阵各元素、次数为 d 的多项式函数 $F(\cdot)$，以及一个满足$w_k\neq 0$（其中 $k\in[K]$）的权重向量 $\boldsymbol{w}\in\mathbb{C}^K$，系统的目标是计算加权聚合，

$$\boldsymbol{Y} \triangleq \sum_{k=0}^{K-1} w_k\, F(\boldsymbol{X}_k), \tag{1}$$

利用来自预先指定的非掉队模式中的一组 $N-S$ 个非掉队工作节点的响应。具体而言，系统不要求针对每个由 $N-S$ 个非掉队工作节点构成的子集都实现恢复，而是假定系统预先知道一组允许的非掉队集合，并据此设计，以便对每个此类集合实现精确恢复；每个集合都是$[N]$ 的子集，且其基数为 $N-S$。

Th非掉队模式的定义如下

定义 *1*（非掉队模式）：给定正整数 N 和 $^\circ S$，非掉队模式定义为$\boldsymbol{\mathcal{N}}$ $\triangleq\{\mathcal{N}_g:\ g\in[G]\}$，其中每个 $\mathcal{N}_g\subseteq[N]$ 都是满足 $|\mathcal{N}_g| = N-S$ 的非掉队者集合。我们定义交集 $\mathcal{I}\triangleq\bigcap_{g\in[G]}\mathcal{N}_g$ 及其基数 $I\triangleq|\mathcal{I}|$。$\diamond$我们进一步定义 $L\triangleq N-S-I$。参数 G 表示该模式中的非掉队者集合数量。例如，$G=1$ 对应于一个指定的非掉队者集合，而 $G=\binom{N}{N-S}$ 对应于所有非掉队者集合。

基于预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$ 的 CPA方案由以下三个阶段组成。

1) 编码：主节点选取一组 K 个互不相同的数据点 $\{\alpha_k\in\mathbb{C}:\ k\in[K]\}$，并插值构造编码器多项式 $E(z)$，使得对所有 $k\in[K]$均有 $E(\alpha_k) = \boldsymbol{X}_k$。接着，主节点选取一组 N 个互不相同的求值点$\{\beta_n\in\mathbb{C}:\ n\in[N]\}$，满足 $\{\alpha_k:\ k\in[K]\}\cap\{\beta_n:\ n\in[N]\}=\emptyset$。主节点在 $\{\beta_n$处对 $E(z)$求值：$n\in[N]\}$，并将编码矩阵 $E(\beta_n)$发送给工作节点 n。

2) 计算：每个工作节点 $n\in[N]$ 在本地计算 $F(E(\beta_n))$，并将结果返回给主节点。

3) 解码：在收到一组 $N-S$工作节点 $\mathcal{N}\in\boldsymbol{\mathcal{N}}$的响应后，主节点插值得到一个解码多项式$D(z)$，使得对所有 $n\in\mathcal{N}$ 均有 $D(\beta_n) = F(E(\beta_n))$。随后，主节点在数据点 $\{\alpha_k:\ k\in[K]\}$ 处对 $D(z)$ 求值，并得到

$$\widehat{\boldsymbol{Y}}(\mathcal{N}) \triangleq \sum_{k=0}^{K-1} w_k\, D(\alpha_k). \tag{2}$$

我们如下定义CPA方案在预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$ 上的可行性。

定义*2*（非掉队模式上的可行性）：固定正整数 K、d、S 和 N，数据点 $\{\alpha_k:\ k\in[K]\}$以及求值点 $\{\beta_n:\ n\in[N]\}$，以及一个预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$。若对所有 $\mathcal{N}\in\boldsymbol{\mathcal{N}}$都有 $\widehat{\boldsymbol{Y}}(\mathcal{N}) = \boldsymbol{Y}$，则称一个CPA方案在$\boldsymbol{\mathcal{N}}$上可行。$\diamond$

在本文中，我们将数据点 $\{\alpha_k:k\in[K]\}$ 视为固定的系统参数[1]。我们假设它们两两不同，即 $\alpha_i\neq\alpha_j$ 对所有 $i\neq j$均成立，并且是一般的[2]，其含义是它们是在由系统参数 K, N, d, S 和 $\boldsymbol{w}$决定的真代数簇之外选取的。

B. 单独解码基线

关于多项式码 [3]–[9]的现有研究 [9] 通过多项式插值解码所有单独子计算，从而恢复所需的计算。其中，拉格朗日编码计算 [9] 是 CPA 设置下自然的基线，因为它适用于多项式计算任务：将每个数据矩阵 $\boldsymbol{X}_k$ 编码为编码器多项式 $E(z)$的一个取值。应用于 CPA 设置时，拉格朗日编码计算会导出以下解码策略。

定义 *3*（基于单独解码的 *CPA*）：基于单独解码的 *CPA*方案按如下方式运行。主节点收到来自非掉队者工作节点集合的响应后，通过插值一个满足 $D(\alpha_k) = F(\boldsymbol{X}_k)$对所有 $k\in[K]$ 均成立的多项式 $D(z)$，重构所有单独取值 $F(\boldsymbol{X}_k)$，然后计算所需的聚合。

以下引理刻画了基于单独解码的 CPA方案在任意掉队模式下实现可行性所需的最少工作节点数。

引理 *1*：对于整数 K、d、S 和 N，当且仅当 $N\geq d(K-1)+S+1$ 时，基于单独解码的 CPA方案在任意 非掉队模式下可行。

证明：在单独解码下，主节点对多项式 $D(z) = F(E(z))$ 进行插值。由于 $\deg(E)\leq K-1$ 且$F(z)$ 的次数为 d，我们有 $\deg(F(E))\leq d(K-1)$。因此，对 $F(E(z))$ 进行插值至少需要 $d(K-1)+1$ 个不同的响应。为确保在任意掉队模式下的可行性，必须能够根据任意大小为 $N-S$ 的非掉队者集合所返回的响应，唯一地插值得到多项式 $F(E(z))$。这要求 $N-S\geq d(K-1)+1$。反过来，若 $N-S\geq d(K-1)+1$，则任意此类非掉队者集合提供的响应都至少包含 $d(K-1)+1$ 个不同的取值，

[1]允许联合设计数据点 $\{\alpha_k:\ k\in[K]\}$和求值点 $\{\beta_n:\ n\in[N]\}$，或可进一步拓展 CPA方案的可行设计空间。

[2]关于一般性和代数簇的概念背景，请读者参阅 [16]。

它唯一确定了 $D(z) = F(E(z))$，从而确定了所有单独的值 $D(\alpha_k) = F(E(\alpha_k)) = F(\boldsymbol{X}_k)$。 ■

由引理 1 可知，对于基于单独解码的 CPA方案，要在任意掉队模式下保证精确恢复，工作节点数量须满足 $N \geq d(K-1) + S + 1$。

本文研究预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$下、处于区间 $N \leq d(K-1) + S$的 CPA；在该区间内，通过单独解码在任意 掉队模式下实现精确恢复是不可行的。我们证明，通过直接利用聚合结构可以实现精确恢复。

III. 主要结果

A. 预先指定的非掉队模式下 CPA 可行性的充要条件

我们不再依赖单独解码，而是直接研究区间 $N \leq d(K-1) + S$内针对 $\mathcal{N}_g \in \boldsymbol{\mathcal{N}}$所产生的恢复误差 $\widehat{\boldsymbol{Y}}(\mathcal{N}_g) - \boldsymbol{Y}$。

定理 *1*：对于满足$S+2 \leq N \leq d(K-1)+S$的正整数 K、 d、 S和 N ，令 $C \triangleq d(K-1)+S+1-N$。对于给定的预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$，CPA方案在$\boldsymbol{\mathcal{N}}$ 上可行当且仅当数据点 $\{\alpha_k \in \mathbb{C} : k \in [K]\}$ 与求值点 $\{\beta_n \in \mathbb{C} : n \in [N]\}$满足 $\{\alpha_k : k \in [K]\} \cap \{\beta_n : n \in [N]\} = \emptyset$ 且

$$\sum_{k=0}^{K-1} w_k P_g(\alpha_k)\, \alpha_k^j = 0, \forall\, j \in [C],\ \forall\, g \in [G], \tag{3}$$

其中 $P_g(z) \triangleq \prod_{n \in \mathcal{N}_g}(z - \beta_n)$， $\mathcal{N}_g \in \boldsymbol{\mathcal{N}}$。

证明：证明见附录 A。 ■

定理 1 表明，当且仅当数据和求值点满足式 (3) 给出的正交条件系统时，CPA方案才能在 $\boldsymbol{\mathcal{N}}$ 上实现。具体而言，每个非掉队者集合 $\mathcal{N}_g$ 都会导出与多项式 $P_g(z)$相关的 C 个正交条件。

式 (3) 中所提条件背后的直观解释如下。由此产生的恢复误差 $\boldsymbol{Y}(\mathcal{N}_g) - \boldsymbol{Y}$可表示为 $\sum_{k=0}^{K-1} w_k P_g(\alpha_k)\alpha_k^j$ （$j \in [C]$）的线性组合。当满足式 (3) 中的正交条件时，所有这些量都等于零，从而消除恢复误差，并实现对 $\boldsymbol{Y}$ 的精确恢复。

注 *1*：在定理 1 中，参数 C 表示引理 1 中单独解码所需的工作节点数量（$d(K-1)+S+1$）与仅恢复聚合时所需的工作节点数量（N）之间的差值。

B. 基于交集结构的充分条件

根据定理 1，对于固定的数据点 $\{\alpha_k : k \in [K]\}$ 和给定的权重向量 $\boldsymbol{w}$，在$\boldsymbol{\mathcal{N}}$ 上设计一个可行的 CPA 方案，归结为选择求值点 $\{\beta_n : n \in [N]\}$，使其同时满足所有正交条件。

可以看出，(3) 展现出一种由非掉队者集合的交集所导出的共同代数结构。具体而言，考虑交集

$$\mathcal{I} \triangleq \bigcap_{g \in [G]} \mathcal{N}_g \qquad P_{\mathcal{I}}\ z\ \triangleq \text{并定义 ( ) ( )。}$$

$\prod_{n \in \mathcal{I}}\ z - \beta_n \qquad \mathcal{I} \subseteq \mathcal{N}_g \qquad g \in [G]$

由于对于所有 ， 多项式$P_{\mathcal{I}}(z)$ 是所有 $P_g(z)$ $g \in [G]$ 的公因子。因此，(3) 中的每个正交条件都可以关于 $P_{\mathcal{I}}(z)$进行因式分解。通过这种因式分解，可以同时满足 与不同非掉队者集合相关的所有正交条件，而只需满足一组仅依赖于 $P_{\mathcal{I}}(z)$ 的精简条件。因此，我们得到引理 2 中所述的充分条件。

引理 *2*：对于满足$S+2 \leq N \leq d(K-1)+S$的正整数 K、 d、 S和 N ，以及预先指定的非掉队模式$\boldsymbol{\mathcal{N}}$，假设

$$\sum_{k=0}^{K-1} w_k\, P_{\mathcal{I}}(\alpha_k)\, \alpha_k^j = 0, \quad j \in [C+L], \tag{4}$$

其中 $P_{\mathcal{I}}(z) \triangleq \prod_{n \in \mathcal{I}}(z - \beta_n)$。则式 (3) 中的正交条件均得到满足。

证明：证明见附录 B。 ■

根据引理 2，要满足式 (3) 中的正交条件，只需针对公共求值点 $\{\beta_n : n \in \mathcal{I}\}$满足式 (4)。因此，对于固定的数据点集合 $\{\alpha_k : k \in [K]\}$，设计一个可行的 CPA 方案，归结为寻找求值点 $\{\beta_n : n \in \mathcal{I}\}$。这一化简引出了以下关于交集大小 I 的充分下界，该下界保证求值点存在。

定理 *2*：对于满足$S+2 \leq N \leq d(K-1)+S$的正整数 K、 d、 S 和 N，预先指定的非掉队模式 $\boldsymbol{\mathcal{N}}$，以及固定的一般位置的两两不同数据点集合 $\{\alpha_k \in \mathbb{C} : k \in [K]\}$，以下结论成立。

1) 存在一种 $\{\beta_n : n \in \mathcal{I}\}$ 的选择，使得 (4) 成立，且 $\{\beta_n : n \in \mathcal{I}\} \cap \{\alpha_k : k \in [K]\} = \emptyset$，当且仅当

$$I \geq I^*,\ I^* = \begin{cases} \left\lfloor \dfrac{K-1}{2} \right\rfloor + 1, & \text{if } d = 1, \\ (d-1)(K-1) + 1, & \text{if } d \geq 2. \end{cases} \tag{5}$$

2) 若 $I \geq I^*$，则存在一种 $\{\beta_n : n \in [N]\}$的选择，满足 (3) 和 $\{\alpha_k : k \in [K]\} \cap \{\beta_n : n \in [N]\} = \emptyset$。

因此，CPA方案在 $\boldsymbol{\mathcal{N}}$ 上可行。

证明：证明过程概述见附录 C。 ■

定理 2 的第一个结论刻画了交集大小 I 的一个充要条件，使得存在满足引理 2 中简化正交条件的求值点 $\{\beta_n : n \in \mathcal{I}\}$。第二个结论表明，条件 $I \geq I^*$ 足以保证存在满足式 (3) 中原始正交条件的求值点 $\{\beta_n : n \in [N]\}$，从而确保 CPA 方案在非掉队模式 $\boldsymbol{\mathcal{N}}$下具有可行性。我们将 I^* 称为充分阈值。

下面的推论表明，当非掉队集合的数量足够大时，充分阈值 I^* 通常也成为必要条件。

推论 *1*：固定一个由数据点$\{\alpha_k$ 构成的一般位置的两两不同集合： $k \in [K]\}$。假设 $G \geq L+1$。对于几乎所有不同的求值点 $\{\beta_n$ 的选择： $n \in [N]\}$，也就是说，除去一个勒贝格测度为零的集合[3]，，简化后的

[3]该勒贝格测度为零的集合对应于一个真代数簇。详情请参见附录 D。

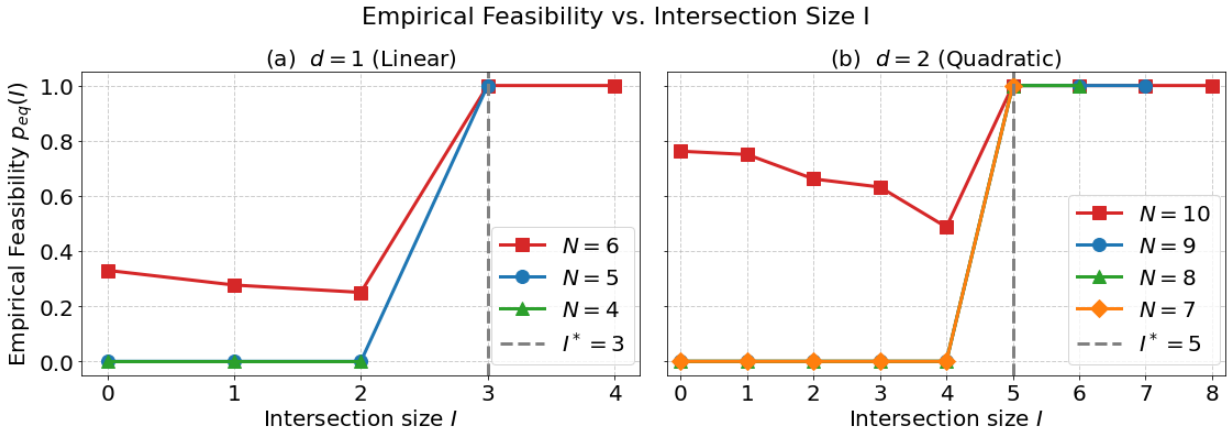


图 1：对于 $K = 5$、$S = 2$ 和 $G_{\max} = 7$，经验可行性 $p_{eq}(I)$ 随交集大小 I 的变化。竖直灰色虚线表示定理 2 中的充分阈值 I^*。对于 $d = 1$，我们考虑 $N = 6$、5、4，分别对应最大交集大小 $I = 4$、3、2；当 $I \le 2$ 时，$N = 5$ 和 $N = 4$ 的曲线重合。对于 $d = 2$，我们考虑 $N = 10$、9、8、7，分别对应最大交集大小 $I = 8$、7、6、5；当 $I \le 5$ 时，$N = 9$、8、7 的曲线重合。

式 (4) 中的正交条件等价于原始的式 (3) 中的正交条件。因此，在此情形下，式 (5) 中的条件 $I \ge I^*$ 对于 CPA 方案的可行性而言一般是必要且充分的。

证明：证明见附录 D。 ■

C. 当 $I \ge I^*$ 时的显式构造

我们给出 $\{\beta_n$ 的显式构造：对于一个满足 $I \ge I^*$ 的$\mathcal{N}$，给定固定的通用两两不同的 $\{\alpha_k : k \in [K]\}$，使得所得 CPA 方案具有可行性。该构造将算法 1 [15] 调整为考虑掉队者的设置：用 $C + L$ 替换 C，并用 I 替换 N。

构造 $\{\beta_n \quad n \in [N]\}$ $\boldsymbol{V} \in$: : $\mathbb{C}^{(C+L)\times K}$ 用 $V[j,k] = \alpha_k^j$、$j \in [C+L]$、$k \in [K]$和 $\boldsymbol{A} \in \mathbb{C}^{K\times(I+1)}$ 以及 $A[k,n] = \alpha_k^n$、$n \in [I+1]$、$k \in [K]$进行构造。计算 $\boldsymbol{U} \triangleq \boldsymbol{V}\mathrm{diag}(\boldsymbol{w})\boldsymbol{A}$，其中$\mathrm{diag}(\boldsymbol{w})$表示对角线为 $\boldsymbol{w}$的对角矩阵。选取非零向量 $\boldsymbol{c} \in \ker(\boldsymbol{U})$，并定义 $P_{\mathrm{form}}(z) = \sum_{n=0}^{I} c_n z^n$。令$\{\beta_n : n \in \mathcal{I}\}$ 为 $P_{\mathrm{form}}(z)$的根。对每个 $g \in [G]$，从 $\mathbb{C} \setminus \{\alpha_k : k \in [K]\}$中选取互异的值 $\{\beta_n : n \in \mathcal{N}_g \setminus \mathcal{I}\}$。

证明：证明见附录 D。 ■

IV. 仿真

本节将针对交集大小 I，对 CPA 的可行性进行经验评估。

A. 仿真设置

固定 K，S，d和 N，我们选择一个整数 $1 \le G_{\max} \le \binom{N}{N-S}$，并改变 $G \in \{1, 2, \ldots, G_{\max}\}$。对于 G 的每个取值，我们从非掉队模式 $\mathcal{N}$ 的实例中均匀抽取 $\min\{100, \binom{\binom{N}{N-S}}{G}\}$ 个互不相同的实例，不放回。对于每个抽取的 $\mathcal{N}$，我们执行以下步骤。固定$\{\alpha_k$：将 $k \in [K]\}$ 设为区间 $[-1, 1]$上的第一类切比雪夫点[17]，即对于 $k \in [K]$，$\alpha_k = \cos\left(\frac{(2k+1)\pi}{2K}\right)$。对权重向量 $\boldsymbol{w}$ 进行抽样，其中各个元素相互独立，且 ——— 均匀取自区间 (0，1)。通过求解满足式 (3) 中正交条件的互异求值点 $\{\beta_n : n \in [N]\}$，对抽取的实例 $\mathcal{N}$ 的可行性进行数值检验。该方法采用非线性最小二乘法，使用 scipy.optimize.least_squares [18]。若找到满足正交条件和互异性条件的数值稳定解，则将实例判定为可行；否则，在进行 10 次随机初始化后，将其判定为不可行。

对于每个交集大小 I，我们计算数值上可行的抽样实例所占的比例。具体而言，对于每个 G，成功率 $p_G(I)$ 定义为交集大小为 I的所有抽样实例中可行实例所占的比例。经验可行性 定义为$p_{\mathrm{eq}}(I) \triangleq \frac{1}{|\mathcal{G}(I)|} \sum_{G \in \mathcal{G}(I)} p_G(I)$，其中 $\mathcal{G}(I)$ 表示~~—~~G 的取值集合，对于其中的每个取值，至少有一个抽样实例的交集大小为 I。

B. 仿真结果

我们在图 1 中绘制了经验可行性 $p_{\mathrm{eq}}(I)$ 随交集大小 I 变化的曲线，分别对应 $d = 1$ 和 $d = 2$。从图 1 中可以得出以下观察结果。对于 $d = 1$ 和 $d = 2$，只要 $I \ge I^*$，经验可行性就达到 100%。交集大小超过阈值后，对于采样得到的非掉队模式，可以找到一组满足正交条件的 $\{\beta_n : n \in [N]\}$ s，以及 $\{\beta_n : n \in [N]\} \cap \{\alpha_k : k \in [K]\} = \emptyset$，这与定理 2 中的充分性阈值 I^* 一致。当 $I < I^*$时，所有满足 $C \ge 2$的情形下，经验可行性均降为 0；这对应于 $N = 5$，4（对于 $d = 1$）以及$N = 9$，8，7（对于 $d = 2$）。这表明，在采样得到的非掉队模式中未观察到可行解。当$C = 1$时，对应于 $d = 1$ 的 $N = 6$ 以及 $d = 2$的 $N = 10$，当 $I < I^*$时会观察到非零经验可行性，因为正交条件的数量 CG 与变量的数量 N相当，因此对于某些非掉队模式可以找到可行解。

附录 A 定理 1 的证明

我们首先考虑标量情形，此时 $\boldsymbol{X}_k$ 简化为一个^标量 x_k，$\boldsymbol{Y}$ 简化为 y，而对于$\mathcal{N}_g \in \boldsymbol{\mathcal{N}}$，$\boldsymbol{Y}(\mathcal{N}_g)$简化为 $\widehat{y}(\mathcal{N}_g)$。矩阵值情形的扩展可逐元素进行，因为多项式 $F(\cdot)$ 对数据矩阵逐元素作用。

定义误差多项式 $\Delta(z) \triangleq D(z) - F(E(z))$。由 $\deg(D) \leq N-S-1$ 以及 $\deg(F(E)) \leq d(K-1)$，我们有 $\deg(\Delta) \leq \max\{\deg(D), \deg(F(E))\} = d(K-1)$。于是，恢复误差为 $\widehat{y}(\mathcal{N}_g) - y = \sum_{k=0}^{K-1} w_k (D(\alpha_k) - F(x_k)) = \sum_{k=0}^{K-1} w_k (D(\alpha_k) - F(E(\alpha_k))) = \sum_{k=0}^{K-1} w_k \Delta(\alpha_k)$。

在解码过程中，$D(\beta_n) = F(E(\beta_n))$ 蕴含 $\Delta(\beta_n) = 0$对所有 $n \in \mathcal{N}_g$ 成立。因此，$\Delta(z)$ 可作因式分解 $\Delta(z) = \prod_{n\in\mathcal{N}_g} z-\beta_n \; R\,z = P_g\,z\,R\,z$ $P_g\,z \triangleq (\,)(\,)(\,)(\,)$, $\prod_{n\in\mathcal{N}_g} z-\beta_n \; R\,z$ R 其中 $(\,)(\,)$，且 $(\,)$ 是满足 $\deg(\,) \leq d\;K-1\;-N+S$ $R\,z$ $(\,)$ 的多项式。

我们将 $(\,) = \sum_{j=0}^{\deg(R)} r_j z^j$ 按任意系数 $\{r_j$ 展开：$j \in [\deg(R)+1]\}$。于是，$\widehat{y}(\mathcal{N}_g) - y = \sum_{k=0}^{K-1} w_k \Delta(\alpha_k) = \sum_{k=0}^{K-1} w_k P_g(\alpha_k) R(\alpha_k) = \sum_{j=0}^{\deg(R)} r_j \left(\sum_{k=0}^{K-1} w_k P_g(\alpha_k)\,\alpha_k^j\right)$。因此，$\widehat{y}(\mathcal{N}_g) - y = 0$ $R\,z$ 对 $(\,)$ 的所有可容许选择成立，当且仅当 $\sum_{k=0}^{K-1} w_k P_g(\alpha_k)\alpha_k^j = 0$ 对所有 $j \in [C]$ 成立。对所有 $\mathcal{N}_g \in$N 强制该条件，即得到 (3) 中的正交条件。同一论证可逐元素应用于矩阵值情形，从而完成证明。

附录 B 引理 2 的证明

定义 $P_{\mathcal{I}}(z) \triangleq \prod_{n\in\mathcal{I}}(z-\beta_n)$。由于 $\mathcal{I} \subseteq \mathcal{N}_g$ 对于所有 $g \in [G]$都成立，$P_{\mathcal{I}}(z)$ 是 $P_g(z)$ 的公因子。因此，我们写作$P_g(z) = Q_g(z)P_{\mathcal{I}}(z)$，其中 $Q_g(z) \triangleq \prod_{n\in\mathcal{N}_g\setminus\mathcal{I}}(z-\beta_n)$ 满足 $\deg(Q_g) = N-S-I \triangleq L$。我们展开 $Q_g(z) = \sum_{l=0}^{L} q_{g,\,l} z^l$，其中 $q_{g,\,l}$ 表示多项式 $Q_g(z)$ 中的系数，且是 $\{\beta_n$ 的函数：$n \in \mathcal{N}_g \setminus \mathcal{I}\}$。然后，(3) 中的每个正交条件都可以改写为

$$\sum_{k=0}^{K-1} w_k P_g(\alpha_k)\alpha_k^j = \sum_{k=0}^{K-1} w_k Q_g(\alpha_k) P_{\mathcal{I}}(\alpha_k)\alpha_k^j = \sum_{l=0}^{L} q_{g,l} \sum_{k=0}^{K-1} w_k P_{\mathcal{I}}(\alpha_k)\alpha_k^{j+l} = 0, \quad (6)$$

因此，使 $\sum_{k=0}^{K-1} w_k\; P_g(\alpha_k)\;\alpha_k^j = 0$ 对所有 $g \in [G]$ 和 $j \in [C]$ 都成立的一个充分条件是，对于 $t \in [C+L]$，$\sum_{k=0}^{K-1} w_k\; P_{\mathcal{I}}(\alpha_k)\alpha_k^t = 0$。引理 2 的证明到此完成。

附录 C 定理 2 的证明概要

定理 2 中的第一项陈述等价于以下命题：存在 $\{\beta_n : n \in \mathcal{I}\}$ 满足

$$\sum_{k=0}^{K-1} w_k P_{\mathcal{I}}(\alpha_k)\alpha_k^j = 0, j \in [C+L], \quad (7)$$

$$\{\alpha_k : k \in [K]\} \cap \{\beta_n : n \in \mathcal{I}\} = \emptyset, \quad (8)$$

$$\beta_n \neq \beta_{n'} \text{ for all } n, n' \in \mathcal{I} \text{ with } n \neq n', \quad (9)$$

当且仅当 $I \geq I^*$。这可通过直接归约到 [15] 中研究的无掉队者设置得出。具体而言，根据 [15] 的定理 2 及其证明，在我们的设置中，条件 (7)–(9) 与 [15] 中定理 2 的条件 (7)–(9) 具有相同的代数形式。与 [15] 中的无掉队者设置不同的是，我们有 $C+L$ 个正交条件，以及一个次数为 I 的多项式 $P_{\mathcal{I}}(z)$。通过调整 [15]，中定理 2 的证明，将 C 替换为 $C+L$，将多项式 $P(z)$ 替换为 $P_{\mathcal{I}}(z)$，并将 N 替换为I，可知存在 $\{\beta_n : n \in \mathcal{I}\}$ 满足 (7)–(9)，当且仅当 $I \geq I^*$。由此得到第一项陈述。

根据引理 2，满足条件 (7)–(9) 的 $\{\beta_n : n \in \mathcal{I}\}$ 足以确保满足定理 1 中的正交条件。此外，对于 $g \in [G]$，$\{\beta_n : n \in \mathcal{N}_g \setminus \mathcal{I}\}$ 可以从 $\mathbb{C}\setminus\{\alpha_k : k \in [K]\}$ 中任意选取，只要它们彼此不同。定理 2 的证明至此完成。

附录 D 推论 1 的证明

假设 $G \geq L+1$。只需证明：若 (3) 中的原始正交条件成立，则 (4) 中的约化条件也成立。在 (6) 中，我们通过提出公因子 $P_{\mathcal{I}}(\alpha_k)$，表示 (3) 中的每个正交条件。汇总所有 $j \in [C]$和 $g \in [G]$对应的这些方程，得到 $\boldsymbol{Q}\boldsymbol{M} = \boldsymbol{0}$，其中 $\boldsymbol{Q} \in \mathbb{C}^{G\times(L+1)}$的元素为 $Q[g,l] = q_{g,\,l}$，适用于 $g \in [G]$ 和 $l \in [L+1]$；而 $\boldsymbol{M} \in \mathbb{C}^{(L+1)\times C}$ $M[l,j] \triangleq$ 则由 $(\,)$ $\sum_{k=0}^{K-1} w_k\, P_{\mathcal{I}}\;\alpha_k\;\alpha_k^{j+l}$ $l \in [L+1]$ $j \in [C]$ 逐项定义，适用于 和 。

根据 $\boldsymbol{Q}\boldsymbol{M} = \boldsymbol{0}$，对于每个 $j \in [C]$，$\boldsymbol{M}$ 的第 j 列属于 $\boldsymbol{Q}$ 的零空间，记为 $\text{null}(\boldsymbol{Q})$。对于 $\{\beta_n$ 的几乎所有取值：$n \in [N]\}$，也就是说，除去一个勒贝格测度为零的集合[4]，，矩阵 $\boldsymbol{Q}$ 满列秩 $L+1$。由于 $G \geq L+1$，可得 $\dim(\text{null}(\boldsymbol{Q})) = L+1-\text{rank}(\boldsymbol{Q}) = L+1-\min(G, L+1) = 0$。因此，$\boldsymbol{Q}$ 的零空间是平凡的，从而 $\boldsymbol{M} = \boldsymbol{0}$。这意味着，对于所有 $l \in [L+1]$ 和 $j \in [C]$，$\sum_{k=0}^{K-1} w_k\, P_{\mathcal{I}}(\alpha_k)\;\alpha_k^{j+l} = 0$。等价地，对于所有 $j \in [C+L]$，$\sum_{k=0}^{K-1} w_k\, P_{\mathcal{I}}(\alpha_k)\;\alpha_k^j = 0$ 成立，这意味着 (4) 成立。这就证明了引理 2 中条件 (4) 的必要性，并完成了证明。

附录 E 关于 $\{\beta_n$ 之构造的证明概要：$n \in [N]\}$

如附录 C 所示，引理 2 中的条件可以化简为无掉队者 CPA 设置下定理 2 中的条件 [15]。由于 [15] 中的算法 1 旨在求解无掉队者情形下的定理 2，因此只需将 C 替换为 $C+L$、将 N 替换为I，即可直接应用于我们的设置。由此，所得多项式 $P_{\text{form}}(z)$的根给出所需的 $\{\beta_n : n \in \mathcal{I}\}$，它们满足(4)以及 $\{\beta_n : n \in \mathcal{I}\} \cap \{\alpha_k : k \in [K]\} = \emptyset$。其余求值点可以任意选取，只要它们与已构造的求值点以及给定的 $\{\alpha_k : k \in [K]\}$互不相同。

[4]使矩阵 $\boldsymbol{Q}$ 的列秩严格小于 $L+1$ 的 $\{\beta_n : n \in [N]\}$ 的集合，由 $\boldsymbol{Q}$的所有$(L+1)\times(L+1)$子式的零点集刻画。该集合是一个真代数簇，因此在 $\mathbb{C}^K$中勒贝格测度为零。

参考文献

[1] K. Lee、M. Lam、R. Pedarsani、D. Papailiopoulos与K. Ramchandran，“利用编码加速分布式机器学习”，*IEEE*信息论汇刊，第64卷，第3期，第1514–1529页，2018年。[2] K. Lee、C. Suh与K. Ramchandran，“高维编码矩阵乘法”，载于*2017*年*IEEE*国际信息论研讨会（*ISIT*），2017年，第2418–2422页。[3] Q. Yu、M. A. Maddah-Ali与A. S. Avestimehr，“多项式码：高维编码矩阵乘法的最优设计”，2018 [ ] 1705 10464 。在线。可获取：https://arxiv.org/abs/ 。[4] S. Dutta、M. Fahim、F. Haddadpour、H. Jeong、V. Cadambe与P. Grover，“论编码矩阵乘法的最优恢复阈值”，*IEEE*信息论汇刊，第66卷，第1期，第278–301页，2020年。[5] Q. Yu与A. S. Avestimehr，“用于安全、私密与批量分布式矩阵乘法的纠缠多项式码：突破“三次”障碍”，载于*2020*年*IEEE*国际信息论研讨会（*ISIT*），2020年，第245–250页。[6] Q. Yu、M. A. Maddah-Ali与A. S. Avestimehr，“分布式矩阵乘法中的掉队者缓解：基本极限与最优编码”，*IEEE*信息论汇刊，第66卷，第3期，第1920–1933页，2020年。[7] M. Fahim与V. R. Cadambe，“数值稳定的多项式编码计算”，载于*2019*年*IEEE*国际信息论研讨会（*ISIT*），2019年，第3017–3021页。[8] B. Hasırcıo˘glu、J. Gómez-Vilardebó与D. Gündüz，“面向高效分布式矩阵乘法的二元埃尔米特多项式编码”，载于*GLOBECOM 2020 - 2020*年*IEEE*全球通信会议，2020年，第1–6页。[9] Q. Yu、S. Li、N. Raviv、S. M. M. Kalan、M. Soltanolkotabi与S. Avestimehr，“拉格朗日编码计算：面向弹性、安全与隐私的最优设计”，2019年。 [在线]。可获取：https://arxiv.org/abs/1806 00939 。[10] T. Jahani-Nezhad与M. A. Maddah-Ali，“Berrut近似编码计算：超越多项式计算的抗掉队性”，*IEEE*模式分析与机器智能汇刊，第45卷，第111–122页，2023年。[11] P. Moradi、B. Tahmasebi与M. A. Maddah-Ali，“面向弹性分布式计算的编码计算：一种学习理论框架”，2024 [ ] 2406 00300 。在线。可获取：https://arxiv.org/abs/ 。[12] P. Moradi与M. A. Maddah-Ali，“概率掉队机制下的通用编码计算”，载于*2025*年*IEEE*国际信息论研讨会（*ISIT*），2025年，第1–6页。[13] J. Xu、S.-L. Huang、L. Song与T. Lan，“用于规避分布式学习中掉队者的实时梯度补偿”，载于*IEEE INFOCOM 2021 - IEEE*计算机通信会议，2021年，第1–10页。[14] K. Wan、H. Sun、M. Ji与G. Caire，“分布式线性可分计算”，*IEEE*信息论汇刊，第68卷，第2期，第1259–1278页，2022年。[15] X. Zhong、J. Kliewer与M. Ji，“编码多项式聚合的基本极限”，2026年。 [在线]。可获取：https://arxiv.org/abs/2601.10028[16] I. R. Shafarevich，基础代数几何，第2版。美国纽约州纽约：Springer-Verlag，1995年。[17] L. N. Trefethen，逼近论与逼近实践。SIAM，2013年。[18] P. Virtanen、R. Gommers、T. E. Oliphant 等，“SciPy 1.0：Python中科学计算的基础算法”，《自然·方法》，第17卷，第3期，第261–272页，2020年。