Efficient Algorithms for Sum-of-Minimum Optimization

Lisang Ding [1] **Ziang Chen** [2] **Xinshang Wang** [3] **Wotao Yin** [3]

Abstract

In this work, we propose a novel optimization model termed "sum-of-minimum" optimization. This model seeks to minimize the sum or average of N objective functions over k parameters, where each objective takes the minimum value of a predefined sub-function with respect to the k parameters. This universal framework encompasses numerous clustering applications in machine learning and related fields. We develop efficient algorithms for solving sum-of-minimum optimization problems, inspired by a randomized initialization algorithm for the classic k-means (Arthur & Vassilvitskii, 2007) and Lloyd's algorithm (Lloyd, 1982). We establish a new tight bound for the generalized initialization algorithm and prove a gradient-descent-like convergence rate for generalized Lloyd's algorithm. The efficiency of our algorithms is numerically examined on multiple tasks, including generalized principal component analysis, mixed linear regression, and small-scale neural network training. Our approach compares favorably to previous ones based on simpler-but-less-precise optimization reformulations.

1. Introduction

In this paper, we propose the following "sum-of-minimum" optimization model:

$$\underset{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k}{\text{minimize}}\ F(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k) := \frac{1}{N}\sum_{i=1}^{N}\min\{f_i(\mathbf{x}_1),f_i(\mathbf{x}_2),\ldots,f_i(\mathbf{x}_k)\}, \tag{1}$$

where $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k$ are unknown parameters to determine. The cost function F is the average of N objectives where the i-th objective is f_i evaluated at its "optimal" out of the k parameter choices. This paper aims to develop efficient algorithms for solving (1) and analyze their performance.

Write $[k]=\{1,2,\ldots,k\}$ and $[N]=\{1,2,\ldots,N\}$. Let $(C_1,C_2,\ldots,C_k)$ be a partition of $[N]$, *i.e.*, C_i's are disjoint subsets of $[N]$ and their union equals $[N]$. Let $\mathcal{P}_N^k$ denote the set of all such partitions. Then, (1) is equivalent to

$$\underset{(C_1,C_2,\ldots,C_k)\in\mathcal{P}_N^k}{\text{minimize}}\ \min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k}\ \frac{1}{N}\sum_{j=1}^{k}\sum_{i\in C_j} f_i(\mathbf{x}_j). \tag{2}$$

It is easy to see $(C_1^*,C_2^*,\ldots,C_k^*)$ and $(\mathbf{x}_1^*,\mathbf{x}_2^*,\ldots,\mathbf{x}_k^*)$ are optimal to (2) if and only if $(\mathbf{x}_1^*,\mathbf{x}_2^*,\ldots,\mathbf{x}_k^*)$ is optimal to (1) and

$$i\in C_j^*\ \Rightarrow\ f_i(\mathbf{x}_j^*)=\min\{f_i(\mathbf{x}_1^*),f_i(\mathbf{x}_2^*),\ldots,f_i(\mathbf{x}_k^*)\}.$$

Reformulation (2) reveals its clustering purpose. It finds the optimal partition $(C_1^*,C_2^*,\ldots,C_k^*)$ such that using the parameter $\mathbf{x}_j^*$ to minimize the average of f_i's in the cluster C_j leads to the minimal total cost.

Problem (1) generalizes **k-means clustering**. Consider N data points $\mathbf{y}_1,\mathbf{y}_2,\ldots,\mathbf{y}_N$ and a distance function $d(\cdot,\cdot)$. The goal of k-means clustering is to find clustering centroids $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k$ that minimize

$$F(\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k)=\frac{1}{N}\sum_{i=1}^{N}\min_{j\in[k]}\{d(\mathbf{x}_j,\mathbf{y}_i)\},$$

which is the average distance from each data point to its nearest cluster center. The literature presents various choices for the distance function $d(\cdot,\cdot)$. When $d(\mathbf{x},\mathbf{y})=\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$, this optimization problem reduces to the classic k-means clustering problem, for which numerous algorithms have been proposed (Krishna & Murty, 1999; Arthur & Vassilvitskii, 2007; Na et al., 2010; Sinaga & Yang, 2020; Ahmed et al., 2020). Bregman divergence is also widely adopted as a distance measure(Banerjee et al., 2005; Manthey & Röglin, 2013; Liu & Belkin, 2016), defined as

$$d(\mathbf{x},\mathbf{y})=h(\mathbf{x})-h(\mathbf{y})-\langle\nabla h(\mathbf{y}),\mathbf{x}-\mathbf{y}\rangle,$$

with h being a differentiable convex function.

[1]Department of Mathematics, University of California, Los Angeles, Los Angeles, CA, USA [2]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA [3]Decision Intelligence Lab, Alibaba US, Bellevue, WA, USA. Correspondence to: Ziang Chen <ziang@mit.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

A special case of (1) is **mixed linear regression**, which generalizes linear regression and models the dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ by multiple linear models. A linear model is a function $g(\mathbf{a}; \mathbf{x}) = \mathbf{a}^\top \mathbf{x}$, which utilizes $\mathbf{x}$ as the coefficient vector for each model. Make k copies of the linear model and set the j-th linear coefficient as $\mathbf{x}_j$. The loss for each data pair $(\mathbf{a}_i, b_i)$ is computed as the squared error from the best-fitting linear model, specifically $\min_{j\in[k]}\{\frac{1}{2}(g(\mathbf{a}_i;\mathbf{x}_j) - b_i)^2\}$. We aim to search for optimal parameters $\{\mathbf{x}_j\}_{j=1}^k$ that minimizes the average loss

$$\frac{1}{N}\sum_{i=1}^{N} \min_{j\in[k]} \left\{ \frac{1}{2}\left(g(\mathbf{a}_i;\mathbf{x}_j) - b_i\right)^2 \right\}. \tag{3}$$

Paper (Zhong et al., 2016) simplifies this non-smooth problem to the sum-of-product problem:

$$\underset{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_k}{\text{minimize}} \frac{1}{N}\sum_{i=1}^{N} \prod_{j\in[k]} \left(g(\mathbf{a}_i;\mathbf{x}_j) - b_i\right)^2, \tag{4}$$

which is smooth. Although (4) is easier to approach due to its smooth objective function, problem (3) is more accurate. Various algorithms are proposed to recover k linear models from mixed-class data (Yi et al., 2014; Shen & Sanghavi, 2019; Kong et al., 2020; Zilber & Nadler, 2023).

In (3), the function $g(\cdot;\mathbf{x})$ parameterized by $\mathbf{x}$ can be any nonlinear function such as neural networks, and we call this extension **mixed nonlinear regression**.

An application of (1) is **generalized principal component analysis (GPCA)** (Vidal et al., 2005; Tsakiris & Vidal, 2017), which aims to recover k low-dimensional subspaces, $V_1, V_2, \ldots, V_k$, from the given data points $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N$, which are assumed to be located on or close to the collective union of these subspaces $V_1 \cup V_2 \cup \cdots \cup V_k$. This process, also referred to as subspace clustering, seeks to accurately segment data points into their respective subspaces (Ma et al., 2008; Vidal, 2011; Elhamifar & Vidal, 2013). Each subspace V_j is represented as $V_j = \{\mathbf{y} \in \mathbb{R}^d : \mathbf{y}^\top \mathbf{A}_j = 0\}$ where $\mathbf{A}_j \in \mathbb{R}^{d\times r}$ and $\mathbf{A}_j^\top \mathbf{A}_j = I_r$, with r being the co-dimension of V_j. From an optimization perspective, the GPCA task can be formulated as

$$\underset{\mathbf{A}_j^\top \mathbf{A}_j = I_r}{\text{minimize}} \frac{1}{N}\sum_{i=1}^{N} \min_{j\in[k]} \left\{ \frac{1}{2}\|\mathbf{y}_i^\top \mathbf{A}_j\|^2 \right\}. \tag{5}$$

Similar to (4), (Peng & Vidal, 2023) works with the less precise reformulation using the product of $\|\mathbf{y}_i^\top \mathbf{A}_j\|^2$ for smoothness and introduces block coordinate descent algorithm.

When $k = 1$, problem (1) reduces to the finite-sum optimization problem

$$\min_{\mathbf{x}} F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}), \tag{6}$$

widely used to train machine learning models, where $f_i(\mathbf{x})$ depicts the loss of the model at parameter $\mathbf{x}$ on the i-th data point. When the underlying model lacks sufficient expressiveness, problem (6) alone may not yield satisfactory results.To enhance a model's performance, one can train the model with multiple parameters, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_k, k \geq 2$, and utilize only the most effective parameter for every data point. This strategy has been successfully applied in various classic tasks, including the aforementioned k-means clustering, mixed linear regression, and the generalized principal component analysis. These applications share a common objective: to segment the dataset into k groups and identify the best parameter for each group. Although no single parameter might perform well across the entire dataset, every data point is adequately served by at least one of the k parameters. By aggregating the strengths of multiple smaller models, this approach not only enhances model expressiveness but also offers a cost-efficient alternative to deploying a singular larger model.

Although one might expect that algorithms and analyses for the sum-of-minimum problem (1) to be weaker as (1) subsumes the discussed previous models, we find our algorithms and analyses for (1) to enhance those known for the existing models. Our algorithms extend the `k-means++` algorithm (Arthur & Vassilvitskii, 2007) and Lloyd's algorithm (Lloyd, 1982), which are proposed for classic k-means problems. We obtain new bounds of these algorithms for (1). Our contributions are summarized as follows:

- We propose the sum-of-minimum optimization problem, adapt `k-means++` to the problem for initialization, and generalize Lloyd's algorithm to approximately solve the problem.
- We establish theoretical guarantees for the proposed algorithms. Specifically, under the assumption that each f_i is L-smooth and μ-strongly convex, we prove the output of the initialization is $\mathcal{O}(\frac{L^2}{\mu^2}\ln k)$-optimal and that this bound is tight with respect to both k and the condition number $\frac{L}{\mu}$. When reducing to k-means optimization, our result recovers that of (Arthur & Vassilvitskii, 2007). Furthermore, we prove an $\mathcal{O}(\frac{1}{T})$ convergence rate for generalized Lloyd's algorithms.
- We numerically verify the efficiency of the proposed framework and algorithms on several tasks, including generalized principal component analysis, ℓ_2-regularized mixed linear regression, and small-scale neural network training. The results reveal that our optimization model and algorithm lead to a higher successful rate in finding the ground-truth clustering, compared to existing approaches that resort to less accurate reformulations for the sake of smoother optimization landscapes. Moreover, our initialization shows signif-

icant improvements in both convergence speed and chance of obtaining better minima.

Our work significantly generalizes classic k-means to handles more complex nonlinear models and provides new perspectives for improving the model performance. The rest of this paper is organized as follows. We introduce the preliminaries and related works in Section 2. We present the algorithms in Section 3. The algorithms are analyzed theoretically in Section 4 and numerically in Section 5. The paper is concluded in Section 6.

Throughout this paper, the ℓ_2-norm and ℓ_2-inner product are denoted by $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$, respectively. We employ $|\cdot|$ as the cardinal number of a set.

2. Related Work and Preliminary

2.1. Related work

Lloyd's algorithm (Lloyd, 1982), a well-established iterative method for the classic k-means problem, alternates between two key steps (MacKay, 2003): 1) assigning $\mathbf{y}_i$ to $\mathbf{x}_j^{(t)}$ if $\mathbf{x}_j^{(t)}$ is the closest to $\mathbf{y}_i$ among $\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_k^{(t)}\}$; 2) updating $\mathbf{x}_j^{(t+1)}$ as the centroid of all $\mathbf{y}_i$'s assigned to $\mathbf{x}_j^{(t)}$. Although Lloyd's algorithm can be proved to converge to stationary points, the results can be highly suboptimal due to the inherent non-convex nature of the problem. Therefore, the performance of Lloyd's algorithm highly depends on the initialization. To address this, a randomized initialization algorithm, $k-$`means++` (Arthur & Vassilvitskii, 2007), generates an initial solution in a sequential fashion. Each centroid $\mathbf{x}_j^{(0)}$ is sampled recurrently according to the distribution

$$\mathbb{P}(\mathbf{x}_j^{(0)} = \mathbf{y}_i) \propto \min_{1\le j'\le j-1} \|\mathbf{x}_{j'} - \mathbf{y}_i\|^2, \quad i \in [N]. \tag{7}$$

The idea is to sample a data point farther from the current centroids with higher probability, ensuring the samples to be more evenly distributed across the dataset. It is proved in (Arthur & Vassilvitskii, 2007) that

$$\mathbb{E}F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}) \le 8(\ln k + 2)F^*, \tag{8}$$

where F^* is the optimal objective value of F. This seminal work has inspired numerous enhancements to the $k-$`means++` algorithm, as evidenced by contributions from (Bahmani et al., 2012; Zimichev et al., 2014; Bachem et al., 2016a;b; Wu et al., 2021; Ren et al., 2022). Our result generalizes the bound in (8), broadening its applicability in sum-of-minimum optimization.

2.2. Definitions and assumptions

In this subsection, we outline the foundational settings for our algorithm and theory. For each sub-function f_i, we establish the following assumptions.

Assumption 2.1. Each f_i is L-smooth, satisfying

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x-y\|, \forall\, x, y \in \mathbb{R}^d,\ i \in [N].$$

Assumption 2.2. Each f_i is μ-strongly convex, for all $x, y \in \mathbb{R}^d$ and $i \in [N]$,

$$f_i(y) \ge f_i(x) + \nabla f_i(x)^\top (y - x) + \frac{\mu}{2}\|x - y\|^2.$$

Let $\mathbf{x}_i^*$ denote the optimizer of $f_i(\mathbf{x})$ such that $f_i^* = f_i(\mathbf{x}_i^*)$, and let

$$S^* = \{\mathbf{x}_i^* : 1 \le i \le N\}$$

represent the solution set. If S^* comprises $l < k$ different elements, the problem (1) possesses infinitely many global minima. Specifically, we can set the variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_l$ to be the l distinct elements in S^*, while leaving $\mathbf{x}_{l+1}, \mathbf{x}_{l+2}, \ldots, \mathbf{x}_k$ as free variables. Given these k variables, $F(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) = \frac{1}{N}\sum_{i=1}^N f_i^*$. If S^* contains more than k distinct components, we have the following proposition.

Proposition 2.3. *Under Assumption 2.2, if $|S^*| \ge k$, the optimization problem (1) admits finitely many minimizers.*

Expanding on the correlation between the number of global minimizers and the size of S^*, we introduce well-posedness conditions for S^*.

Definition 2.4 (k-separate and (k, r)-separate)**.** We call S^* k-separate if it contains at least k different elements, *i.e.*, $|S^*| \ge k$. Furthermore, we call S^* (k, r)-separate if there exists $1 \le i_1 < i_2 < \cdots < i_k \le N$ such that $\|\mathbf{x}_{i_j}^* - \mathbf{x}_{i_{j'}}^*\| > 2r$ for all $j \ne j'$.

Finally, we address the optimality measurement in (1). The norm of the (sub)-gradient is an inappropriate measure for global optimality due to the problem's non-convex nature. Instead, we utilize the following optimality gap.

Definition 2.5 (Optimality gap)**.** Given a point $\mathbf{x}$, the optimality gap of f_i at $\mathbf{x}$ is $f_i(\mathbf{x}) - f_i^*$. Given a finite point set $\mathcal{M}$, the optimality gap of f_i at $\mathcal{M}$ is $\min_{\mathbf{x}\in\mathcal{M}} f_i(\mathbf{x}) - f_i^*$. When $\mathcal{M} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$, the averaged optimality gap of $f_1, f_2, \ldots, f_N$ at $\mathcal{M}$ is the shifted objective function

$$F(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) - \frac{1}{N}\sum_{i=1}^N f_i^*. \tag{9}$$

The averaged optimality gap in (9) will be used as the optimality measurement throughout this paper. Specifically, in the classic k-means problem, one has $f_i^* = 0$, so the function $F(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$ directly indicates global optimality.

3. Algorithms

In this section, we introduce the algorithm for solving the sum-of-minimum optimization problem (1). Our approach is twofold, comprising an initialization phase based on k−means++ and a generalized version of Lloyd's algorithm.

3.1. Initialization

As the sum-of-minimum optimization (1) can be considered a generalization of the classic k-means clustering, we adopt k−means++. In k−means++, clustering centers are selected sequentially from the dataset, with each data point chosen based on a probability proportional to its squared distance from the nearest existing clustering centers, as detailed in (7). We generalize this idea and propose the following initialization algorithm that outputs initial parameters $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}$ for the problem (1).

First, we select an index i_1 at random from $[N]$, following a uniform distribution, and then utilize a specific method to determine the minimizer $\mathbf{x}_{i_1}^*$, setting

$$\mathbf{x}_1^{(0)} = \mathbf{x}_{i_1}^* = \operatorname*{argmin}_{\mathbf{x}} f_{i_1}(\mathbf{x}). \tag{10}$$

For $j = 2, 3, \ldots, k$, we sample i_j based on the existing variables $\mathcal{M}_j = \{\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_{j-1}^{(0)}\}$, with each index i sampled based on a probability proportional to the optimality gap of f_i at $\mathcal{M}_j$. Specifically, we compute the minimal optimality gaps

$$v_i^{(j)} = \min_{1 \le j' \le j-1} \left(f_i(\mathbf{x}_{j'}^{(0)}) - f_i^* \right), \quad i \in [N], \tag{11}$$

as probability scores. Each score $v_i^{(j)}$ can be regarded as an indicator of how unresolved an instance f_i is with the current variables $\{\mathbf{x}_{j'}^{(0)}\}_{j'=1}^{j-1}$. We then normalize these scores

$$w_i^{(j)} = \frac{v_i^{(j)}}{\sum_{i'=1}^N v_{i'}^{(j)}}, \quad i \in [N], \tag{12}$$

and sample $i_j \in [N]$ following the probability distribution $\mathbf{w}^{(j)} = \left(w_1^{(j)}, \ldots, w_N^{(j)}\right)$. The j-th initialization is determined by optimizing f_{i_j},

$$\mathbf{x}_j^{(0)} = \mathbf{x}_{i_j}^* = \operatorname*{argmin}_{\mathbf{x}} f_{i_j}(\mathbf{x}). \tag{13}$$

We terminate the selection process once k variables $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}$ are determined. The pseudo-code of this algorithm is shown in Algorithm 1.

We note that the scores $v_i^{(j)}$ defined in (11) rely on the optimal objectives f_i^*, which may be computationally intensive to calculate in certain scenarios. Therefore, we propose a variant of Algorithm 1 by adjusting the scores $v_i^{(j)}$. Specifically, when $j-1$ parameters $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_{j-1}^{(0)}$ are selected, the score is set as the minimum squared norm of the gradient:

$$v_i^{(j)} = \min_{1 \le j' \le j-1} \left\| \nabla f_i(\mathbf{x}_{j'}^{(0)}) \right\|^2. \tag{14}$$

This variant involves replacing the scores in Step 3 of Algorithm 1 with (14), which is further elaborated in Appendix B.

In the context of classic k-means clustering where $f_i(\mathbf{x}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}_i\|^2$ for the i-th data point $\mathbf{y}_i$, the score $v_i^{(j)}$ in both (11) and (14) reduces to

$$\min_{1 \le j' \le j-1} \|\mathbf{x}_{j'}^{(0)} - \mathbf{y}_i\|^2,$$

up to a constant scalar. This initialization algorithm, whether utilizing scores from (11) or (14), aligns with the approach of the classic k−means++ algorithm.

Algorithm 1 Initialization

1: Sample i_1 uniformly at random from $[N]$ and compute $\mathbf{x}_1^{(0)}$ via (10).
2: **for** $j = 2, 3, \ldots, k$ **do**
3: Compute $\mathbf{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_N^{(j)}\right)$ via (11).
4: Compute $\mathbf{w}^{(j)} = \left(w_1^{(j)}, \ldots, w_N^{(j)}\right)$ via (12).
5: Sample $i_j \in [N]$ according to the weights $\mathbf{w}^{(j)}$ and compute $\mathbf{x}_j^{(0)}$ via (13).
6: **end for**

3.2. Generalized Lloyd's algorithm

Lloyd's algorithm is employed to minimize the loss in k-means clustering by alternately updating the clusters and their centroids (Lloyd, 1982; MacKay, 2003). This centroid update process can be regarded as a form of gradient descent applied to group functions, defined by the average distance between data points within a cluster and its centroid (Bottou & Bengio, 1994). For our problem (1), we introduce a novel gradient descent algorithm that utilizes dynamic group functions. Our algorithm is structured into two main phases: reclassification and group gradient descent.

Reclassification. The goal is for $C_j^{(t)}$ to encompass all $i \in [N]$ where f_i is active at $\mathbf{x}_j^{(t)}$, allowing us to use the sub-functions f_i within $C_j^{(t)}$ to update $\mathbf{x}_j^{(t)}$. This process leads to the reclassification step as follows:

$$C_j^{(t)} = \left\{ i \in [N] : f_i(\mathbf{x}_j^{(t)}) \le f_i(\mathbf{x}_{j'}^{(t)}), \forall\, j' \in [k] \right\} \Big\backslash \left(\bigcup_{l<j} C_l^{(t)} \right), \quad j = 1, 2, \ldots, k. \tag{15}$$

Algorithm 2 Generalized Lloyd's Algorithm

Generate the initialization $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}$ and set r, γ.
for $t = 0, 1, 2, \ldots, T$ **do**
  if $t \equiv 0 \pmod r$ **then**
    Compute the partition $\{C_j^{(t)}\}_{j=1}^k$ via (15).
  else
    $C_j^{(t)} = C_j^{(t-1)}, \quad 1 \le j \le k.$
  end if
  Compute $\mathbf{x}_j^{(t+1)}$ via (17).
end for

Given that reclassification may incur non-negligible costs in practice, a reclassification frequency r can be established, performing the update in (15) every r iterations while keeping $C_j^{(t)} = C_j^{(t-1)}$ constant during other iterations.

Group gradient descent. With $C_j^{(t)}$ indicating the active f_i at $\mathbf{x}_j^{(t)}$, we can define the group objective function:

$$F_j^{(t)}(\mathbf{z}) = \begin{cases} \frac{1}{|C_j^{(t)}|} \sum_{i \in C_j^{(t)}} f_i(\mathbf{z}), & C_j^{(t)} \neq \emptyset, \\ 0, & C_j^{(t)} = \emptyset, \end{cases} \tag{16}$$

In each iteration, gradient descent is performed on $\mathbf{x}_j^{(t)}$ individually as:

$$\mathbf{x}_j^{(t+1)} = \mathbf{x}_j^{(t)} - \gamma \nabla F_j^{(t)}(\mathbf{x}_j^{(t)}). \tag{17}$$

Here, $\gamma > 0$ is the chosen step size. Alternatively, one might opt for different iterative updates or directly compute:

$$\mathbf{x}_j^{(t+1)} = \operatorname*{argmin}_{\mathbf{x}} \sum_{i \in C_j^{(t)}} f_i(\mathbf{x}),$$

especially if the minimizer of $\sum_{i \in C_j^{(t)}} f_i(\mathbf{x})$ admits a closed form or can be computed efficiently. The pseudo-code consisting of the above two steps is presented in Algorithm 2.

Momentum Lloyd's Algorithm. We enhance Algorithm 1 by incorporating a momentum term. The momentum for $\mathbf{x}_j^{(t)}$ is represented as $\mathbf{m}_j^{(t)}$, with $0 < \beta < 1$ and $\gamma > 0$ serving as the step sizes for the momentum-based updates. We use the gradient of the group function $F_j^{(t)}$ to update the momentum $\mathbf{m}_j^{(t)}$. The momentum algorithm admits the following form:

$$\mathbf{x}_j^{(t+1)} = \mathbf{x}_j^{(t)} - \gamma \mathbf{m}_j^{(t)}, \tag{18}$$

$$\mathbf{m}_j^{(t+1)} = \beta \mathbf{m}_j^{(t)} + \nabla F_j^{(t+1)}(\mathbf{x}_j^{(t+1)}). \tag{19}$$

A critical aspect of the momentum algorithm involves updating the classes $C_j^{(t)}$ between (18) and (19). Rather than reclassifying based on f_i evaluated at $\mathbf{x}_j^{(t+1)}$, reclassification leverages an acceleration variable:

$$\mathbf{u}_j^{(t+1)} = \frac{1}{1-\beta}(\mathbf{x}_j^{(t+1)} - \beta \mathbf{x}_j^{(t)}). \tag{20}$$

The index i will be classified to $C_j^{(t+1)}$ where $f_i(\mathbf{u}_j^{(t+1)})$ attains the minimal value. Furthermore, to mitigate abrupt shifts in each class C_j, we implement a *controlled* reclassification scheme that limits the extent of change in each class:

$$\frac{1}{\alpha}|C_j^{(t)}| \le |C_j^{(t+1)}| \le \alpha |C_j^{(t)}|, \tag{21}$$

where $\alpha > 1$ serves as a constraint factor. Details of the momentum algorithm are provided in Appendix B. We display the pseudo-code in Algorithm 3.

Algorithm 3 Momentum Lloyd's Algorithm

Generate the initialization $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}$. Set $\mathbf{m}_1^{(0)}, \mathbf{m}_2^{(0)}, \ldots, \mathbf{m}_k^{(0)}$ to be $\mathbf{0}$. Set r, α, β, γ.
for $t = 0, 1, 2, \ldots, T$ **do**
  Update $\mathbf{x}_j^{(t)}$ using (18).
  if $t \equiv 0 \pmod r$ **then**
    Compute $\mathbf{u}_j^{(t+1)}$ via (20).
    Update $C_j^{(t+1)}$ with $\mathbf{u}_j^{(t+1)}$ in control, such that (21) holds.
  else
    $C_j^{(t+1)} = C_j^{(t)}, \quad 1 \le j \le k.$
  end if
  Update the momentum $\mathbf{m}_j^{(t)}$ via (19).
end for

4. Theoretical Analysis

In this section, we prove the efficiency of the initialization algorithm and establish the convergence rate of Lloyd's algorithm. For the initialization Algorithm 1, we show that the ratio between the optimality gap of $\{\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}\}$ and the smallest possible optimality gap is $\mathcal{O}(\frac{L^2}{\mu^2} \ln k)$. Additionally, by presenting an example where this ratio is $\Omega(\frac{L^2}{\mu^2} \ln k)$, we illustrate the bound's tightness. For Lloyd's Algorithms 2 and 3, we establish a gradient decay rate of $\mathcal{O}(\frac{1}{T})$, underscoring the efficiency and convergence properties of these algorithms.

4.1. Error bound of the initialization algorithm

We define the set of initial points selected by the randomized initialization Algorithm 1,

$$\mathcal{M}_{\text{init}} = \{\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}\} = \{\mathbf{x}_{i_1}^*, \mathbf{x}_{i_2}^*, \ldots, \mathbf{x}_{i_k}^*\},$$

as the starting configuration for our optimization process. For simplicity, we use $F(\mathcal{M}_{\text{init}}) = F(\mathbf{x}_{i_1}^*, \mathbf{x}_{i_2}^*, \ldots, \mathbf{x}_{i_k}^*)$ to

represent the function value at these initial points. Let F^* be the global minimal value of F, and let $f^* = \frac{1}{N}\sum_{i=1}^N f_i^*$ denote the average of the optimal values of sub-functions. The effectiveness of Algorithm 1 is evaluated by the ratio between $\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^*$ and $F^* - f^*$, which is the expected ratio between the averaged optimality gap at $\mathcal{M}_{\text{init}}$ and the minimal possible averaged optimality gap. The following theorem provides a specific bound.

Theorem 4.1. *Suppose that Assumptions 2.1 and 2.2 hold. Assume that the solution set S^* is k-separate. Let $\mathcal{M}_{\text{init}}$ be a random initialization set generated by Algorithm 1. We have*

$$\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^* \le 4(2+\ln k)\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)(F^*-f^*).$$

Theorem 4.1 indicates that the relative optimality gap at the initialization set is constrained by a factor of $\mathcal{O}(\frac{L^2}{\mu^2}\ln k)$ times the minimal optimality gap. The proof of Theorem 4.1 is detailed in Appendix C. In the classic k-means problem, where $L = \mu$, this result reduces to Theorem 1.1 in (Arthur & Vassilvitskii, 2007). Moreover, the upper bound $\mathcal{O}(\frac{L^2}{\mu^2}\ln k)$ is proven to be tight via a lower bound established in the following theorem.

Theorem 4.2. *Given a fixed cluster number $k > 0$, there exists an integer $N > 0$. We can construct N sub-functions $\{f_i\}_{i=1}^N$ satisfying Assumptions 2.1–2.2 and guaranteeing the solution set S^* to be k-separate. When applying Algorithm 1 over the instances $\{f_i\}_{i=1}^N$, we have*

$$\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^* \ge \frac{1}{2}\frac{L^2}{\mu^2}\ln k\,(F^*-f^*). \tag{22}$$

The proof of Theorem 4.2 is presented in detail in Appendix C. In both Theorem 4.1 and Theorem 4.2, the performance of Algorithm 1 is analyzed with the assumption that $\mathbf{v}^{(j)}$ and f_i^* in (11) can be computed exactly. However, the accurate computation of f_i^* may be impractical due to computational costs. Therefore, we explore the error bounds when the score $\mathbf{v}^{(j)}$ approximates (11) with some degree of error. We investigate two types of scoring errors.

- **Additive error.** There exists $\epsilon > 0$, we have access to an estimated $\tilde{f}_i^*$ satisfying

$$f_i^* - \epsilon \le \tilde{f}_i^* \le f_i^* + \epsilon. \tag{23}$$

Accordingly, we define:

$$\begin{aligned}\tilde{v}_i^{(j)} &= \min_{1\le j'\le j-1}\left(\max\left(f_i(\mathbf{x}_{j'}^{(0)}) - \tilde{f}_i^*, 0\right)\right)\\ &= \max\left(\min_{1\le j'\le j-1}\left(f_i(\mathbf{x}_{j'}^{(0)}) - \tilde{f}_i^*\right), 0\right).\end{aligned} \tag{24}$$

- **Scaling error.** There exists a deterministic oracle $O_v : [N]\times\mathbb{R}^d \to \mathbb{R}$, such that for any $\mathbf{x}\in\mathbb{R}^d$ and $i\in[N]$,

$$c_1(f_i(\mathbf{x}) - f_i^*) \le O_v(i,\mathbf{x}) \le c_2(f_i(\mathbf{x}) - f_i^*). \tag{25}$$

Set

$$\tilde{v}_i^{(j)} = \min_{1\le j'\le j-1} O_v(i, \mathbf{x}_{j'}^{(0)}). \tag{26}$$

We first analyze the performance of Algorithm 1 using the score $\tilde{v}_i^{(j)}$ with additive error as in (24). We typically require the assumption that the solution set S^* is $(k, \sqrt{\frac{2\epsilon}{\mu}})$-separate, which guarantees that

$$\sum_{i=1}^N \min_{j\in[l]} \max\left((f_i(\mathbf{z}_j) - \tilde{f}_i^*), 0\right) > 0,$$

for any $l < k$ and $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_l \in \mathbb{R}^d$. Hence in the initialization Algorithm 1 with score (24), there is at least one $\tilde{v}_i^{(j)} > 0$ in each round. We have the following generalized version of Theorem 4.1 with additive error.

Theorem 4.3. *Under Assumptions 2.1 and 2.2, suppose that we have $\{\tilde{f}_i^*\}_{i=1}^N$ satisfying (23) for some noise factor $\epsilon > 0$, and that the solution set S^* is $(k, \sqrt{\frac{2\epsilon}{\mu}})$-separate. Then for the initialization Algorithm 1 with the scores in (11) replaced by the noisy scores in (24), we have*

$$\begin{aligned}\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^* \le{}& 4(2+\ln k)\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)(F^*-f^*)\\ &+ \epsilon\cdot\left(1+(2+\ln k)\left(1+\frac{4L}{\mu}\right)\right).\end{aligned} \tag{27}$$

The proof of Theorem 4.3 is deferred to Appendix C. Next, we state a similar result for the scaling-error oracle as in (26), whose proof is deferred to Appendix C.

Theorem 4.4. *Suppose that Assumptions 2.1–2.2 hold and that the solution set S^* is k-separate. Then for the initialization Algorithm 1 with the scores in (11) replaced by the scores in (26), we have the following bound:*

$$\begin{aligned}&\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^*\\ &\quad\le 4\left(\frac{c_2}{c_1}\frac{L}{\mu}+\frac{c_2^2}{c_1^2}\frac{L^2}{\mu^2}\right)(2+\ln k)(F^*-f^*).\end{aligned}$$

Recall that we introduce an alternative score in (14). This score can actually be viewed as a noisy version of (11) with a scaling error. Under Assumptions 2.1 and 2.2, it holds that

$$2\mu(f_i(\mathbf{x}) - f_i^*) \le \|\nabla f_i(\mathbf{x})\|^2 \le 2L(f_i(\mathbf{x}) - f_i^*),$$

for any $i\in[N]$ and $\mathbf{x}\in\mathbb{R}^d$, which satisfies (25) with $c_1 = 2\mu$ and $c_2 = 2L$. Therefore, we have a direct corollary of Theorem 4.4.

Corollary 4.5. *Suppose that Assumptions 2.1 and 2.2 hold and that the solution set S^* is k-separate. For the initialization Algorithm 1 with the scores in (11) replaced by the scores in (14), we have*

$$\mathbb{E}F(\mathcal{M}_{\text{init}}) - f^* \leq 4\left(\frac{L^2}{\mu^2} + \frac{L^4}{\mu^4}\right)(2 + \ln k)(F^* - f^*).$$

4.2. Convergence rate of Lloyd's algorithm

In this subsection, we state convergence results of Lloyd's Algorithm 2 and momentum Lloyd's Algorithm 3, with all proofs being deferred to Appendix D. For Algorithm 2, the optimization process of $\mathbf{x}_j^{(t)}$ follows a gradient descent scheme on a varying objective function $F_j^{(t)}$, which is the average of all active f_i's determined by $C_j^{(t)}$ in (15). We have the following gradient-descent-like convergence rate on the gradient norm $\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|$.

Theorem 4.6. *Suppose that Assumption 2.1 is satisfied and we take the step size $\gamma = \frac{1}{L}$ in Algorithm 2. Then*

$$\frac{1}{T+1}\sum_{t=0}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\left\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\right\|^2 \leq \frac{2L}{T+1}\left(F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}) - F^\star\right).$$

For momentum Lloyd's Algorithm 3, we have a similar convergence rate stated as follows.

Theorem 4.7. *Suppose that Assumption 2.1 holds and that $\alpha > 1$. For Algorithm 3, there exists a constant $\bar{\gamma}(\alpha, \beta, L)$, such that*

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{2(1-\beta)}{\gamma}\cdot\frac{F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}) - F^*}{T},$$

as long as $\gamma \leq \bar{\gamma}(\alpha, \beta, L)$.

5. Numerical Experiments

In this section, we conduct numerical experiments to demonstrate the efficiency of the proposed model and algorithms. Our code with documentation can be found at `https://github.com/LisangDing/Sum-of-Minimum_Optimization`.

5.1. Comparison between the sum-of-minimum model and the product formulation

We consider two optimization models for generalized principal component analysis: the sum-of-minimum formulation (5) and another widely acknowledged formulation given by (Peng & Vidal, 2023; Vidal et al., 2005):

$$\underset{\mathbf{A}_j^\top \mathbf{A}_j = I_r}{\text{minimize}} \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{k}\|\mathbf{y}_i^\top \mathbf{A}_j\|^2. \tag{28}$$

The initialization for both formulations is generated by Algorithm 1. We use a slightly modified version of Algorithm 2 to minimize (5) since the minimization of the group functions for GPCA admits closed-form solutions. In particular, we alternatively compute the minimizer of each group objective function as the update of $\mathbf{A}_j$ and then reclassify the sub-functions. We use the block coordinate descent (BCD) method (Peng & Vidal, 2023) to minimize (28). The BCD algorithm alternatively minimizes $\mathbf{A}_j$ with all other $\mathbf{A}_l$ $(l \neq j)$ being fixed. The pseudo-codes of both algorithms are included in Appendix E.1.

We set the cluster number $k \in \{2, 3, 4\}$, dimension $d \in \{4, 5, 6\}$, subspace co-dimension $r = d - 2$, and the number of data points $N = 1000$. The generalization of the dataset $\{\boldsymbol{y}_i\}_{i=1}^N$ is described in Appendix E.1. We set the maximum iteration number as 50 for Algorithm 2 with (5) and terminate the algorithm once the objective function stops decreasing, i.e., the partition/clustering remains unchanged. Meanwhile, we set the iteration number to 50 for the BCD algorithm (Peng & Vidal, 2023) with (28). The sythetic data generation is elaborated in Appendix E.1. The classification accuracy of both methods is reported in Table 1, where the classification accuracy is defined as the maximal matching accuracy with respect to the ground truth over all permutations. We observe that our model and algorithm lead to significantly higher accuracy. This is because, compared to (28), the formulation in (5) models the requirements more precisely, though it is more difficult to optimize due to the non-smoothness.

Next, we compare the computational cost for our model and algorithms with that of the product model and the BCD algorithm. We observe that the BCD algorithm exhibited limited improvements in accuracy after the initial 10 iterations. Thus, for a fare comparison, we set both the maximum iterations for our model and algorithms and the iteration number for the BCD algorithm to 10. The accuracy rate and the CPU time are shown in Table 2, from which one can see that the computational costs of our algorithm and the BCD algorithm are competitive, while our algorithm achieves much better classification accuracy.

5.2. Comparison between different initializations

We present the performance of Lloyd's Algorithm 2 combined with different initialization methods. The initialization methods adopted in this subsection are:

- **Normal initialization.** We initialize variables

Table 1. Cluster accuracy percentages of the sum-of-minimum (vs. sum-of-product) GPCA models after 50 iterations.

	$d = 4$	$d = 5$	$d = 6$
$k = 2$	**98.24** (81.88)	**98.07** (75.90)	**98.19** (73.33)
$k = 3$	**95.04** (67.69)	**94.98** (62.89)	**95.94** (60.85)
$k = 4$	**91.30** (62.36)	**92.92** (59.65)	**93.73** (57.89)

Table 2. Averaged cluster accuracy percentages (CPU time in seconds) for GPCA after 10 iterations. In each setting, the first and the second rows display the results for the sum-of-minimum and the sum-of-product models respectively.

	$d = 4$	$d = 5$	$d = 6$
$k = 2$	**97.84 (0.08)**	**97.93 (0.08)**	**98.01 (0.08)**
	81.78 (0.14)	75.76 (0.14)	73.24 (0.15)
$k = 3$	**93.34 (0.19)**	**94.14 (0.19)**	**95.25 (0.16)**
	67.18 (0.20)	62.76 (0.22)	60.80 (0.20)
$k = 4$	**88.62 (0.32)**	**91.78 (0.29)**	**92.62 (0.27)**
	61.52 (0.26)	59.37 (0.27)	57.82 (0.27)

$\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}$ with i.i.d. samples from the d-dimensional standard Gaussian distribution.

- **Uniform seeding index initialization.** We uniformly sample k different indices $i_1, i_2, \ldots, i_k$ from $[N]$, then we set $\mathbf{x}_{i_j}^*$ as the initial value of $\mathbf{x}_j^{(0)}$.
- **Careful seeding index initialization.** We sample the k indices using Algorithm 1 and initialize $\mathbf{x}_j^{(0)}$ with the minimizer of the corresponding sub-function.

Mixed linear regression. Our first example is the ℓ_2-regularized mixed linear regression. We add an ℓ_2 regularization on each sub-function f_i in (3) to guarantee strong convexity, and the sum-of-minimum optimization objective function can be written as

$$\frac{1}{N}\sum_{i=1}^{N} \min_{j\in[k]} \left\{ \frac{1}{2}\left(g(\mathbf{a}_i;\mathbf{x}_j) - b_i\right)^2 + \frac{\lambda}{2}\|\mathbf{x}_j\|^2 \right\},$$

where $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ collects all data points and $\lambda > 0$ is a fixed parameter. The dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ is generated as described in Appendix E.2.

Similar to the GPCA problem, we slightly modify Lloyd's algorithm since the ℓ_2-regularized least-square problem can be solved analytically. Specifically, we use the minimizer of the group objective function as the update of $\mathbf{x}_j$ instead of performing the gradient descent as in (17) or Algorithm 2. We perform the algorithm until a maximum iteration number is met or the objective function value stops decreasing. The detailed algorithm is given in Appendix E.2.

In the experiment, the number of samples is set to $N = 1000$ and we vary k from 4 to 6 and d (the dimension of $\mathbf{a}_i$ and $\mathbf{x}_j$) from 4 to 8. For each problem with fixed cluster number and dimension, we repeat the experiment for 1000 times with different random seeds. In each repeated experiment, we record two metrics. If the output objective value at the last iteration is less than or equal to $F(\mathbf{x}_1^+, \mathbf{x}_2^+, \ldots, \mathbf{x}_k^+)$, where $(\mathbf{x}_1^+, \mathbf{x}_2^+, \ldots, \mathbf{x}_k^+)$ is the ground truth that generates the dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$, we consider the objective function to be nearly optimized and label the algorithm as successful on the task; otherwise, we label the algorithm as failed on the task. Additionally, we record the number of iterations the algorithm takes to output a result. The result is displayed in Table 3.

Mixed nonlinear regression. Our second experiment is on mixed nonlinear regression using 2-layer neural networks. We construct k neural networks with the same structure and let the j-th neural network be:

$$\psi(\mathbf{a}; \mathbf{W}_j, \mathbf{p}_j, \mathbf{q}_j, o_j) = \mathbf{p}_j^\top \mathrm{ReLU}(\mathbf{W}_j\mathbf{a} + \mathbf{q}_j) + o_j.$$

Here, $\mathbf{a}$ is the input data. We let d_I be the input dimension and d_H be the hidden dimension. The dimensions of the variables are $\mathbf{a} \in \mathbb{R}^{d_I}, \mathbf{W}_j \in \mathbb{R}^{d_H \times d_I}, \mathbf{p}_j, \mathbf{q}_j \in \mathbb{R}^{d_H}, o_j \in \mathbb{R}$. We denote $\theta_j = (\mathbf{W}_j, \mathbf{p}_j, \mathbf{q}_j, o_j)$ as the trainable parameters in the neural network. For each trial, we prepare the ground truth θ_j^+ and the dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ as described in Appendix E.2. We use the squared ℓ_2 loss for each neural network and construct the i-th sub-function as:

$$f_i(\theta) = \frac{1}{2}(\psi(\mathbf{a}_i;\theta) - b_i)^2 + \frac{\lambda}{2}\|\theta\|^2,$$

where we still use $\frac{\lambda}{2}\|\theta\|^2, \lambda > 0$ as a regularization term. We perform parallel experiments on training the neural networks via Algorithm 2 using three different initialization methods. During the training process of neural networks, stochastic gradient descent is commonly used to manage limited memory, reduce training loss, and improve generalization. Moreover, the ADAM algorithm proposed in (Kingma & Ba, 2014) is widely applied. This optimizer is empirically observed to be less sensitive to hyperparameters, more robust, and to converge faster. To align with this practice, we replace the group gradient descent in Algorithm 2 and the group momentum method in Algorithm 3 with ADAM optimizer-based backward propagation for the corresponding group objective function.

We use two metrics to measure the performance of the algorithms. In one set of experiments, we train k neural networks until the value of the loss function F under parameters $\theta_1, \theta_2, \ldots, \theta_k$ is less than that under $\theta_1^+, \theta_2^+, \ldots, \theta_k^+$. We record the average iterations required to achieve the optimization loss. In the other set of experiments, we train k neural networks for a fixed number of iterations. Then, we compute the training and testing loss of the trained neural

Table 3. The failing rate (average iteration number) of three initialization methods when solving mixed linear regression problems with different cluster numbers and dimensionality. A smaller failure rate and a lower average iteration number indicate better performance. The least failure rate among the three methods is bolded, and the least average iteration number under the same cluster number and dimension settings is underlined.

	Init. Method	$d=4$	$d=5$	$d=6$	$d=7$	$d=8$
$k=4$	*normal*	0.056 (17.577)	**0.031** (18.378)	0.038 (19.923)	0.058 (21.631)	0.071 (22.344)
	unif. seeding	0.057 (16.139)	0.034 (16.885)	0.050 (18.022)	0.055 (18.708)	0.075 (19.959)
	caref. seeding	**0.050** (14.551)	0.036 (15.276)	**0.034** (16.020)	**0.044** (16.936)	**0.051** (17.409)
$k=5$	*normal*	0.161 (26.355)	0.156 (28.844)	0.172 (32.247)	0.238 (35.042)	0.321 (38.324)
	unif. seeding	**0.145** (23.728)	0.136 (25.914)	**0.143** (27.671)	0.198 (29.935)	0.256 (32.662)
	caref. seeding	0.162 (21.552)	**0.130** (23.476)	**0.143** (25.933)	**0.161** (27.268)	**0.217** (29.086)
$k=6$	*normal*	0.363 (35.831)	0.382 (41.043)	0.504 (43.999)	0.594 (47.918)	0.739 (48.730)
	unif. seeding	0.347 (31.536)	0.350 (35.230)	0.408 (39.688)	0.524 (42.453)	0.596 (43.117)
	caref. seeding	**0.339** (29.610)	**0.312** (33.460)	**0.389** (36.068)	**0.463** (39.010)	**0.563** (40.320)

network, where the training loss on the dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ is defined as $\frac{1}{N}\sum_{i=1}^N \min_j \left(\frac{1}{2}(\psi(\mathbf{a}_i;\theta_j)-b_i)^2\right)$ and the testing loss is defined in a similar way.

In our experiments, the training dataset size is $N=1000$ and the testing dataset size is 200. The testing dataset is generated from the same distribution as the training data. Benefiting from ADAM's robust nature regarding hyperparameters, we use the default ADAM learning rate $\gamma = 1e{-}3$. We set $r=10$ in Lloyd's Algorithm 2 and fix the cluster number $k=5$. We test on three different (d_I, d_H) tuples: $(5,3)$, $(7,5)$, and $(10,5)$. The results can be found in Table 4 and 5.

Table 4. Average epochs for different seeding methods to achieve the ground truth model training loss.

(d_I, d_H)	(5,3)	(7,5)	(10,5)
normal	329.4	132.1	130.8
unif. Seeding	233.1	71.2	67.6
caref. Seeding	**181.4**	**49.3**	**47.2**

Table 5. The training (testing) errors (unit: 10e-3) of Lloyd's algorithm with fixed training epoch numbers.

(d_I, d_H) / Iter.	(5,3) / 300	(7,5) / 150	(10,5) / 150
normal	4.26 (4.63)	4.57 (5.54)	4.62 (5.82)
unif. Seeding	3.86 (4.25)	3.96 (4.77)	3.56 (4.52)
caref. Seeding	**3.44** (3.93)	**3.51** (4.37)	**3.39** (4.34)

We can conclude from Table 3, 4, and 5 that the careful seeding Algorithm 1 generates the best initialization in most cases. This initialization algorithm results in the fewest iterations required by Lloyd's algorithm to converge, the smallest final loss, and the highest probability of finding the ground-truth clustering.

6. Conclusion

This paper proposes a general framework for sum-of-minimum optimization, as well as efficient initialization and optimization algorithms. Theoretically, tight bounds are established for smooth and strongly convex sub-functions f_i. Though this work is motivated by classic algorithms for the k-means problem, we extend the ideas and theory significantly for a broad family of tasks. Furthermore, the numerical efficiency is validated for generalized principal component analysis and mixed linear and nonlinear regression problems. Future directions include developing algorithms with provable guarantees for non-convex f_i and exploring empirical potentials on large-scale tasks.

Acknowledgements

Lisang Ding receives support from Air Force Office of Scientific Research Grants MURI-FA9550-18-1-0502. A major part of the work of Ziang Chen was completed during his internship at Alibaba US DAMO Academy. We thank Liangzu Peng for fruitful discussions on GPCA.

Impact Statement

The proposed unified framework for sum-of-minimum optimization covers a vast array of classic problems and modern machine learning tasks. From a long-range perspective, the ultimate goal of this paper is to provide guidance for improving the performance of machine learning models by using multiple sets of parameters. Therefore, the algorithms and ideas can potentially be applied in almost all fields of machine learning, though this paper only paves the first step.

References

Ahmed, M., Seraj, R., and Islam, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.

Arthur, D. and Vassilvitskii, S. K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

Bachem, O., Lucic, M., Hassani, H., and Krause, A. Fast and provably good seedings for k-means. *Advances in neural information processing systems*, 29, 2016a.

Bachem, O., Lucic, M., Hassani, S. H., and Krause, A. Approximate k-means++ in sublinear time. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016b.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. Scalable k-means++. *arXiv preprint arXiv:1203.6402*, 2012.

Banerjee, A., Merugu, S., Dhillon, I. S., Ghosh, J., and Lafferty, J. Clustering with bregman divergences. *Journal of machine learning research*, 6(10), 2005.

Bottou, L. and Bengio, Y. Convergence properties of the k-means algorithms. *Advances in neural information processing systems*, 7, 1994.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11): 2765–2781, 2013.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kong, W., Somani, R., Song, Z., Kakade, S., and Oh, S. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404. PMLR, 2020.

Krishna, K. and Murty, M. N. Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439, 1999.

Liu, C. and Belkin, M. Clustering with bregman divergences: an asymptotic analysis. *Advances in Neural Information Processing Systems*, 29, 2016.

Lloyd, S. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Ma, Y., Yang, A. Y., Derksen, H., and Fossum, R. Estimation of subspace arrangements with applications in modeling and segmenting mixed data. *SIAM review*, 50 (3):413–458, 2008.

MacKay, D. An example inference task: Clustering. *Information theory, inference and learning algorithms*, 20: 284–292, 2003.

Manthey, B. and Röglin, H. Worst-case and smoothed analysis of k-means clustering with bregman divergences. *Journal of computational geometry*, 4(1):94–132, 2013.

Na, S., Xumin, L., and Yong, G. Research on k-means clustering algorithm: An improved k-means clustering algorithm. In *2010 Third International Symposium on intelligent information technology and security informatics*, pp. 63–67. Ieee, 2010.

Peng, L. and Vidal, R. Block coordinate descent on smooth manifolds: Convergence theory and twenty-one examples. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023.

Ren, Q., Zhang, D., Zhao, X., Yan, L., Rui, J., et al. A novel hybrid method of lithology identification based on k-means++ algorithm and fuzzy decision tree. *Journal of Petroleum Science and Engineering*, 208:109681, 2022.

Shen, Y. and Sanghavi, S. Iterative least trimmed squares for mixed linear regression. *Advances in Neural Information Processing Systems*, 32, 2019.

Sinaga, K. P. and Yang, M.-S. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

Tsakiris, M. C. and Vidal, R. Filtrated algebraic subspace clustering. *SIAM Journal on Imaging Sciences*, 10(1): 372–415, 2017.

Vidal, R. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011.

Vidal, R., Ma, Y., and Sastry, S. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005.

Wu, J., Shi, L., Yang, L., XiaxiaNiu, Li, Y., XiaodongCui, Tsai, S.-B., and Zhang, Y. User value identification based on improved rfm model and k-means++ algorithm for complex data analysis. *Wireless Communications and Mobile Computing*, 2021:1–8, 2021.

Yi, X., Caramanis, C., and Sanghavi, S. Alternating minimization for mixed linear regression. In *International Conference on Machine Learning*, pp. 613–621. PMLR, 2014.

Zhong, K., Jain, P., and Dhillon, I. S. Mixed linear regression with multiple components. *Advances in neural information processing systems*, 29, 2016.

Zilber, P. and Nadler, B. Imbalanced mixed linear regression. *arXiv preprint arXiv:2301.12559*, 2023.

Zimichev, E. A., Kazanskii, N. L., and Serafimovich, P. G. Spectral-spatial classification with k-means++ particional clustering. *Computer Optics*, 38(2):281–286, 2014.

A. Proof of Proposition 2.3

In this section, we provide a proof of the proposition in Section 2.

Proposition A.1 (Restatement of Proposition 2.3)**.** *Under Assumption 2.2, if $|S^*| \geq k$, the optimization problem (1) admits finitely many minimizers.*

Proof. If $|S^*| = k$, then the only minimizer up to a permutation of indices is $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$, such that

$$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\} = S^*.$$

Next we consider the case where $|S^*| > k$. Let $\mathcal{R}$ be the set of all minimizers of (1). Due to the μ-strong convexity of f_i, the set $\mathcal{R}$ is nonempty. Let $\mathcal{T}$ be the set of all partitions $C_1, C_2, \ldots, C_k$ of $[N]$, such that $C_j \neq \emptyset$ for all $j \in [k]$. The set $\mathcal{T}$ is finite. Next, we show there is an injection from $\mathcal{R}$ to $\mathcal{T}$. For $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) \in \mathcal{R}$, we recurrently define

$$C_j^{\mathbf{X}} = \{i \in [N] \,|\, f_i(\mathbf{x}_j) = \min_l(f_i(\mathbf{x}_l))\} \backslash \left(\cup_{1 \leq j' \leq j-1} C_{j'}^{\mathbf{X}}\right).$$

We claim that all $C_j^{\mathbf{X}}$'s are nonempty. Otherwise, if there is an index j such that $C_j^{\mathbf{X}} = \emptyset$, we have a $\mathbf{z} \in S^* \backslash \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$. Replacing the j-th parameter $\mathbf{x}_j$ with $\mathbf{z}$, we have

$$F(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k) > F(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{j-1}, \mathbf{z}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_k).$$

This contradicts the assumption that $\mathbf{X}$ is a minimizer of (1). Therefore, $\mathbf{X} \to (C_1^{\mathbf{X}}, C_2^{\mathbf{X}}, \ldots, C_k^{\mathbf{X}})$ is a well-defined map from $\mathcal{R}$ to $\mathcal{T}$. Consider another $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k) \in \mathcal{R}$. If $C_j^{\mathbf{X}} = C_j^{\mathbf{Y}}$ for all $j \in [k]$, due to the μ-strong convexity of f_i's, we have

$$\mathbf{y}_j = \operatorname*{argmin}_{\mathbf{z}} \sum_{i \in C_j^{\mathbf{Y}}} f_i(\mathbf{z}) = \operatorname*{argmin}_{\mathbf{z}} \sum_{i \in C_j^{\mathbf{X}}} f_i(\mathbf{z}) = \mathbf{x}_j, \quad \forall j \in [k].$$

Thus, the map defined above is injective. Overall, $\mathcal{R}$ is a finite set. □

B. Algorithm details

In this section, we provide the details of the algorithms presented in Section 3.

B.1. Initialization with alternative scores

When the score function $v_i^{(j)}$ is taken as the squared gradient norm as in (14), the pseudo-code of the initialization can be found in Algorithm 4.

B.2. Details on momentum Lloyd's Algorithm

In this section, we elaborate on the details of momentum Lloyd's Algorithm 3. We use $\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_k^{(t)}$ as the k variables to be optimized. Correspondingly, we introduce $\mathbf{m}_1^{(t)}, \mathbf{m}_2^{(t)}, \ldots, \mathbf{m}_k^{(t)}$ as their momentum. We use the same notation $F_j^{(t)}$ in (16) as the group objective function. In each iteration, we update $\mathbf{x}$ using momentum gradient descent and update $\mathbf{m}$ using the gradient of the group function.

$$\begin{aligned}
\mathbf{x}_j^{(t+1)} &= \mathbf{x}_j^{(t)} - \gamma \mathbf{m}_j^{(t)}, \\
\mathbf{m}_j^{(t+1)} &= \beta \mathbf{m}_j^{(t)} + \nabla F_j^{(t+1)}(\mathbf{x}_j^{(t+1)}).
\end{aligned}$$

The update of $C_j^{(t)}$ in the momentum algorithm is different from the Lloyd's Algorithm 2. We introduce an acceleration quantity

$$\mathbf{u}_j^{(t+1)} = \frac{1}{1-\beta}(\mathbf{x}_j^{(t+1)} - \beta \mathbf{x}_j^{(t)}).$$

Each class is then renewed around the center $\mathbf{u}_j^{(t+1)}$. We update index $i \in [N]$ to the class $C_j^{(t+1)}$ where $f_i(\mathbf{u}_j^{(t+1)})$ attains the minimum value among all $j \in [k]$. To ensure the stability of the momentum accumulation, we further introduce a

Algorithm 4 Initialization

1: Sample i_1 uniformly at random from $[N]$ and compute

$$\mathbf{x}_1^{(0)} = \mathbf{x}_{i_1}^* = \operatorname*{argmin}_{\mathbf{x}} f_{i_1}(\mathbf{x}).$$

2: **for** $j = 2, 3, \ldots, k$ **do**

3: Compute scores $\mathbf{v}^{(j)} = \left(v_1^{(j)}, v_2^{(j)}, \ldots, v_N^{(j)}\right)$ via

$$v_i^{(j)} = \min_{1 \le j' \le j-1} \left\| \nabla f_i(\mathbf{x}_{j'}^{(0)}) \right\|^2.$$

4: Compute the sampling weights $\mathbf{w}^{(j)} = \left(w_1^{(j)}, \ldots, w_N^{(j)}\right)$ by normalizing $\{v_i^{(j)}\}_{i=1}^N$,

$$w_i^{(j)} = \frac{v_i^{(j)}}{\sum_{i'=1}^N v_{i'}^{(j)}}.$$

5: Sample $i_j \in [N]$ according to the weights $\mathbf{w}^{(j)}$ and compute

$$\mathbf{x}_j^{(0)} = \mathbf{x}_{i_j}^* = \operatorname*{argmin}_{\mathbf{x}} f_{i_j}(\mathbf{x}).$$

6: **end for**

controlled reclassification method. We set a reclassification factor $\alpha > 1$. We update $C_j^{(t)}$ to $C_j^{(t+1)}$ in the following way to ensure

$$\frac{1}{\alpha}|C_j^{(t)}| \le |C_j^{(t+1)}| \le \alpha |C_j^{(t)}|.$$

The key idea is to carefully reclassify each index one by one until the size of one class breaks the above restriction. We construct $C_{j,0} = C_j^{(t)}, j \in [k]$ as the initialization of the reclassification. We randomly, non-repeatedly pick indices i from $[N]$ one by one. For l looping from 1 to N, we let $C_{j,l-1}, j \in [k]$ be the classification before the l-th random index is picked. Let i_l be the l-th index sampled. We reassign i_l to the j-th class, such that

$$f_{i_l}(\mathbf{u}_j^{(t+1)}) = \min_{j' \in [k]} f_{i_l}(\mathbf{u}_{j'}^{(t+1)}).$$

There will be at most two classes changed due to the one-index reassignment. We update the class notations from $C_{j',l-1}$ to $C_{j',l}$ for all $j' \in [N]$. If there is any change between $C_{j',l-1}$ and $C_{j',l}$, we check whether

$$\frac{1}{\alpha}|C_{j'}^{(t)}| \le |C_{j'}| \le \alpha |C_{j'}^{(t)}|$$

holds. If the above restriction holds for all $j' \in [N]$, we accept the reclassification and move on to the next index sample. Otherwise, we stop the process and return $C_j^{(t+1)} = C_{j,l-1}, j \in [k]$. If the reclassification trial successfully loops to the last index. We assign $C_j^{(t+1)} = C_{j,N}, j \in [k]$.

C. Initialization error bounds

In this section, we prove the error bounds of the initialization Algorithms 1 and 4. Before our proof, we prepare the following concepts and definitions.

Definition C.1. For any nonempty $C \subset [N]$, we define

$$\Delta_C := \frac{1}{|C|} \sum_{i \in C} \sum_{i' \in C} \|\mathbf{x}_i^* - \mathbf{x}_{i'}^*\|^2.$$

Definition C.2. Let $\mathcal{I} \subset [N]$ be an index set, $\mathcal{M} \subset \mathbb{R}^d$ be a finite set, we define

$$\mathcal{A}(\mathcal{I}, \mathcal{M}) = \sum_{i \in \mathcal{I}} \min_{\mathbf{z} \in \mathcal{M}} (f_i(\mathbf{z}) - f_i(\mathbf{x}_i^*)),$$
$$\mathcal{D}(\mathcal{I}, \mathcal{M}) = \sum_{i \in \mathcal{I}} \min_{\mathbf{z} \in \mathcal{M}} \|\nabla f_i(\mathbf{z})\|^2.$$

Under the μ-strong convexity and L-smooth Assumptions 2.1 and 2.2, we immediately have

$$\frac{1}{2L}\mathcal{D}(\mathcal{I}, \mathcal{M}) \le \mathcal{A}(\mathcal{I}, \mathcal{M}) \le \frac{1}{2\mu}\mathcal{D}(\mathcal{I}, \mathcal{M}).$$

Besides, for disjoint index sets $\mathcal{I}_1, \mathcal{I}_2$, we have

$$\mathcal{A}(\mathcal{I}_1 \cup \mathcal{I}_2, \mathcal{M}) = \mathcal{A}(\mathcal{I}_1, \mathcal{M}) + \mathcal{A}(\mathcal{I}_2, \mathcal{M}),$$
$$\mathcal{D}(\mathcal{I}_1 \cup \mathcal{I}_2, \mathcal{M}) = \mathcal{D}(\mathcal{I}_1, \mathcal{M}) + \mathcal{D}(\mathcal{I}_2, \mathcal{M}).$$

For the problem (1), the optimal solution exists due to the strong convexity assumption on f_i's. We pick one set of optimal solutions $\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_k^*$. We let

$$\mathcal{M}_{\text{OPT}} = \{\mathbf{z}_1^*, \mathbf{z}_2^*, \ldots, \mathbf{z}_k^*\}.$$

Based on this optimal solutions, we introduce $(A_1, A_2, \ldots, A_k)$ as a partition of $[N]$. A_j's are disjoint with each other and

$$\bigcup_{j \in [k]} A_j = [N].$$

Besides, for all $i \in A_j$, $f_i(\mathbf{x})$ attains minimum at $\mathbf{z}_j^*$ over $\mathcal{M}_{\text{OPT}}$,

$$f_i(\mathbf{z}_j^*) - f_i(\mathbf{x}_i^*) = \min_{j' \in [k]} \left(f_i(\mathbf{z}_{j'}^*) - f_i(\mathbf{x}_i^*)\right).$$

The choice of $\mathcal{M}_{\text{OPT}}$ and $(A_1, A_2, \ldots, A_k)$ is not unique. We carefully choose them so that A_j are non-empty for each $j \in [k]$.

Lemma C.3. *Suppose that Assumption 2.1 holds. Let $\mathcal{I}$ be a nonempty index subset of $[N]$ and let i be sampled uniformly at random from $\mathcal{I}$. We have*

$$\mathbb{E}_i \, \mathcal{A}(\mathcal{I}, \{\mathbf{x}_i^*\}) \le \frac{L}{2}\Delta_{\mathcal{I}}.$$

Proof. We have the following direct inequality.

$$\begin{aligned}
\mathbb{E}_i \, \mathcal{A}(\mathcal{I}, \{\mathbf{x}_i^*\}) &= \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathcal{A}(\mathcal{I}, \{\mathbf{x}_i^*\}) \\
&= \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{i' \in \mathcal{I}} (f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) \\
&\le \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{i' \in \mathcal{I}} \frac{L}{2} \|\mathbf{x}_i^* - \mathbf{x}_{i'}^*\|^2 \\
&= \frac{L}{2}\Delta_{\mathcal{I}}.
\end{aligned}$$

□

Lemma C.4. *Let $\mathcal{M}$ be a fixed finite set in $\mathbb{R}^d$. For two indices $i \neq i'$, we have*

$$\mathcal{A}(\{i\}, \mathcal{M}) \le \frac{2L}{\mu}\mathcal{A}(\{i'\}, \mathcal{M}) + L\|\mathbf{x}_i^* - \mathbf{x}_{i'}^*\|^2$$

Proof. We have the following inequality.

$$\begin{aligned}
\mathcal{A}(\{i\},\mathcal{M}) &= \min_{\mathbf{z}\in\mathcal{M}} (f_i(\mathbf{z}) - f_i(\mathbf{x}_i^*)) \\
&\le \min_{\mathbf{z}\in\mathcal{M}} \frac{L}{2}\|\mathbf{z}-\mathbf{x}_i^*\|^2 \\
&\le \min_{\mathbf{z}\in\mathcal{M}} L(\|\mathbf{z}-\mathbf{x}_{i'}^*\|^2 + \|\mathbf{x}_{i'}^*-\mathbf{x}_i^*\|^2) \\
&\le \frac{2L}{\mu}\min_{z\in\mathcal{M}} (f_{i'}(\mathbf{z}) - f_{i'}(\mathbf{x}_{i'}^*)) + L\|\mathbf{x}_{i'}^*-\mathbf{x}_i^*\|^2 \\
&= \frac{2L}{\mu}\mathcal{A}(\{i'\},\mathcal{M}) + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2.
\end{aligned}$$

□

Lemma C.5. *Given an index set $\mathcal{I}$ and a finite point set $\mathcal{M}$, suppose that $\mathcal{A}(\mathcal{I},\mathcal{M}) > 0$. If we randomly sample an index $i \in \mathcal{I}$ with probability $\frac{\mathcal{A}(\{i\},\mathcal{M})}{\mathcal{A}(\mathcal{I},\mathcal{M})}$, then we have the following inequality,*

$$\mathbb{E}\mathcal{A}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\}) \le \left(\frac{L^2}{\mu}+L\right)\Delta_{\mathcal{I}}.$$

Proof. We consider the expectation of $\mathcal{A}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\})$ over $i\in\mathcal{I}$. We have the following inequality bound.

$$\begin{aligned}
\mathbb{E}\mathcal{A}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\}) &= \sum_{i\in\mathcal{I}} \frac{\mathcal{A}(\{i\},\mathcal{M})}{\mathcal{A}(\mathcal{I},\mathcal{M})}\mathcal{A}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\}) \\
&= \sum_{i\in\mathcal{I}} \frac{\mathcal{A}(\{i\},\mathcal{M})}{\mathcal{A}(\mathcal{I},\mathcal{M})} \sum_{i'\in\mathcal{I}} \min(\mathcal{A}(\{i'\},\mathcal{M}), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) \\
&\overset{(a)}{\le} \sum_{i\in\mathcal{I}} \frac{\frac{1}{|\mathcal{I}|}\sum_{i''\in\mathcal{I}}\left(\frac{2L}{\mu}\mathcal{A}(\{i''\},\mathcal{M}) + L\|\mathbf{x}_{i''}^*-\mathbf{x}_i^*\|^2\right)}{\mathcal{A}(\mathcal{I},\mathcal{M})} \sum_{i'\in\mathcal{I}} \min(\mathcal{A}(\{i'\},\mathcal{M}), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) \\
&= \frac{2L}{\mu}\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i'\in\mathcal{I}} \min(\mathcal{A}(\{i'\},\mathcal{M}), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) \\
&\quad + \frac{L}{\mathcal{A}(\mathcal{I},\mathcal{M})|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i''\in\mathcal{I}}\|\mathbf{x}_{i''}^*-\mathbf{x}_i^*\|^2 \sum_{i'\in\mathcal{I}} \min(\mathcal{A}(\{i'\},\mathcal{M}), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) \\
&\le \frac{2L}{\mu}\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i'\in\mathcal{I}}\frac{L}{2}\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + \frac{L}{\mathcal{A}(\mathcal{I},\mathcal{M})|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i''\in\mathcal{I}}\|\mathbf{x}_{i''}^*-\mathbf{x}_i^*\|^2\sum_{i'\in\mathcal{I}}\mathcal{A}(\{i'\},\mathcal{M}) \\
&= \left(\frac{L^2}{\mu}+L\right)\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i'\in\mathcal{I}}\|\mathbf{x}_{i'}^*-\mathbf{x}_i^*\|^2.
\end{aligned}$$

Here, (a) holds when applying Lemma C.4. □

Lemma C.6. *For any A_l in the optimal partition $(A_1, A_2, \ldots, A_k)$, we have*

$$\Delta_{A_l} \le \frac{4}{\mu}\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}}).$$

Proof. We let $\bar{\mathbf{y}}_l = \frac{1}{|A_l|}\sum_{i\in A_l}\mathbf{x}_i^*$ be the geometric center of optimal f_i solutions of index set A_l.

$$\begin{aligned}
\Delta_{A_l} &= \frac{1}{|A_l|}\sum_{i\in A_l}\sum_{i'\in A_l}\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 \\
&= \frac{1}{|A_l|}\sum_{i\in A_l}\sum_{i'\in A_l}\|\mathbf{x}_i^*-\bar{\mathbf{y}}_l+\bar{\mathbf{y}}_l-\mathbf{x}_{i'}^*\|^2
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{|A_l|} \sum_{i\in A_l} \sum_{i'\in A_l} \left(\|\mathbf{x}_i^* - \bar{\mathbf{y}}_l\|^2 + \|\bar{\mathbf{y}}_l - \mathbf{x}_{i'}^*\|^2\right) \\
&= 2\sum_{i\in A_l} \|\mathbf{x}_i^* - \bar{\mathbf{y}}_l\|^2 \\
&= 2\min_{\mathbf{z}} \sum_{i\in A_l} \|\mathbf{x}_i^* - \mathbf{z}\|^2 \\
&\le \frac{4}{\mu} \min_{\mathbf{z}} \sum_{i\in A_l} (f_i(\mathbf{z}) - f_i(\mathbf{x}_i^*)) \\
&= \frac{4}{\mu} \min_{\mathbf{z}} \mathcal{A}(A_l, \{\mathbf{z}\}) \\
&= \frac{4}{\mu} \mathcal{A}(A_l.\mathcal{M}_{\text{OPT}}).
\end{aligned}
$$

□

Proposition C.7. *Let $\mathcal{I}$ be an index set, and $\mathcal{M}$ be a finite point set. Let $\mathbf{z}^*$ be a minimizer of the objective function $\sum_{i\in\mathcal{I}} (f_i(\mathbf{z}) - f_i(\mathbf{x}_i^*))$. Suppose that $\mathcal{A}(\mathcal{I}, \mathcal{M}) > 0$. If we sample an index $i \in \mathcal{I}$ with probability $\frac{\mathcal{A}(\{i\},\mathcal{M})}{\mathcal{A}(\mathcal{I},\mathcal{M})}$, then we have the following inequality:*

$$
\mathbb{E}\mathcal{A}(\mathcal{I}, \mathcal{M} \cup \{\mathbf{x}_i^*\}) \le 4\left(\frac{L^2}{\mu^2} + \frac{L}{\mu}\right) \mathcal{A}(A_l.\mathcal{M}_{\text{OPT}}). \tag{29}
$$

Proof. The deduction of (29) is a direct combination of Lemma C.5 and Lemma C.6. □

Next we prove that the $\frac{L^2}{\mu^2}$ bound in (29) is tight.

Proposition C.8. *Fix the dimension $d \ge 1$, there exists an integer N. We can construct N μ-strongly convex and L-smooth sub-functions $f_1, f_2, \ldots, f_N$, and a finite set $\mathcal{M} \subseteq \mathbb{R}^d$. We let $\{f_i\}_{i=1}^N$ be the N sub-functions of the sum-of-minimum optimization problem (1). When we sample an index $i \in [N]$ with probability $\frac{\mathcal{A}(\{i\},\mathcal{M})}{\mathcal{A}([N],\mathcal{M})}$, we have*

$$
\mathbb{E}\mathcal{A}([N], \mathcal{M} \cup \{\mathbf{x}_i^*\}) \ge \frac{L^2}{\mu^2} \mathcal{A}(A_l, \mathcal{M}_{\text{OPT}}).
$$

Proof. For the cases where the dimension $d \ge 2$, we construct the instance in a more concise way. We consider the following $n+1$ points, $\mathbf{x}_i^* = (1, 0, 0, \ldots, 0) \in \mathbb{R}^d$, $i = 1, 2, \ldots, n$, $\mathbf{x}_{n+1}^* = (-1, 0, 0, \ldots, 0) \in \mathbb{R}^d$. All the elements except the first one of $\mathbf{x}_i^*$ are zero. We construct the following functions f_i with minimizers $\mathbf{x}_i^*$.

$$
\begin{aligned}
f_i(y_1, y_2, \ldots, y_d) &= \frac{L}{2}(y_1 - 1)^2 + \frac{\mu}{2}\sum_{j=2}^d y_j^2, \quad i = 1, 2 \ldots, n, \\
f_i(y_1, y_2, \ldots, y_d) &= \frac{\mu}{2}(y_1 + 1)^2 + \frac{L}{2}\sum_{j=2}^d y_j^2, \quad i = n+1.
\end{aligned} \tag{30}
$$

We have $f_i^* := f_i(\mathbf{x}_i^*) = 0$ for all $i \in [n+1]$. We construct the finite set $\mathcal{M}$ in an orthogonal manner. We let $\mathcal{M} = \{(0, \xi)\}$, $\xi \in \mathbb{R}^{d-1}$ be a single point set. Besides, $\|\xi\| = m \gg 1$. The point $\boldsymbol{\xi} = (0, \xi)$ in $\mathcal{M}$ is orthogonal to all $\mathbf{x}_i^*$'s. Consider the expectation over the newly sampled index i, we have

$$
\mathbb{E}\sum_{i'=1}^{n+1} \min(f_{i'}(\boldsymbol{\xi}) - f_{i'}(\mathbf{x}_{i'}^*), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) = \frac{n(L+\mu m^2)}{n(L+\mu m^2) + (\mu + Lm^2)} 2\mu + \frac{\mu + Lm^2}{n(L+\mu m^2) + (\mu + Lm^2)} 2nL
$$

We set $m = \exp(n)$. As $n \to \infty$, we have

$$
\lim_{n\to\infty} \mathbb{E}\sum_{i'=1}^{n+1} \min(f_{i'}(\boldsymbol{\xi}) - f_{i'}(\mathbf{x}_{i'}^*), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*)) = 2\mu + 2\frac{L^2}{\mu}.
$$

In the meanwhile, we have

$$\mathbf{z}^* := \operatorname*{argmin}_{\mathbf{z}} \sum_{i'=1}^{n+1} (f_{i'}(\mathbf{z}) - f_{i'}(\mathbf{x}_{i'}^*)) = \frac{nL-\mu}{nL+\mu},$$
$$\sum_{i'=1}^{n+1} (f_{i'}(\mathbf{z}^*) - f_{i'}(\mathbf{x}_{i'}^*)) = \frac{2\mu nL}{\mu+nL} \overset{n\to\infty}{\to} 2\mu$$

We have the following error rate:

$$\lim_{n\to\infty} \frac{\mathbb{E}\sum_{i'=1}^{n+1} \min(f_{i'}(\boldsymbol{\xi}) - f_{i'}(\mathbf{x}_{i'}^*), f_{i'}(\mathbf{x}_i^*) - f_{i'}(\mathbf{x}_{i'}^*))}{\sum_{i'=1}^{n+1}(f_{i'}(\mathbf{z}^*) - f_{i'}(\mathbf{x}_{i'}^*))} = 1 + \frac{L^2}{\mu^2}.$$

As for the 1D case, we consider the following $n+1$ points. We let $x_i^* = 1, i = 1, 2, \ldots, n$, and $x_{n+1}^* = 0$. We construct:

$$f_i(x) = \begin{cases} \frac{L}{2}(x-1)^2, & x \le 1, \\ \frac{\mu}{2}(x-1)^2, & x \ge 1, \end{cases} \qquad i = 1, 2, \ldots, n.$$
$$f_i(x) = \begin{cases} \frac{\mu}{2}x^2, & x \le 1, \\ \frac{L}{2}(x-1)^2 + \mu\left(x - \frac{1}{2}\right), & x \ge 1, \end{cases} \qquad i = n+1.$$

Each f_i^* has the minimizer x_i^*. Besides, $f_i^* := f_i(x_i^*) = 0$. We let $\mathcal{M} = \{1 + \frac{L}{\mu}\}$ be a single point set. Let $\xi = 1 + \frac{L}{\mu}$. We have

$$f_i(x_{n+1}^*) - f_i^* = \frac{L}{2}, \quad i = 1, 2, \ldots, n,$$
$$f_{n+1}(x_i^*) - f_{n+1}^* = \frac{\mu}{2}, \quad i = 1, 2, \ldots, n,$$
$$f_i\left(1 + \frac{L}{\mu}\right) - f_i^* = \frac{L^2}{2\mu}, \quad i = 1, 2, \ldots, n,$$
$$f_{n+1}\left(1 + \frac{L}{\mu}\right) - f_{n+1}^* = \frac{L^3 + 2\mu^2 L + \mu^3}{2\mu^2}.$$

We have the following expectation:

$$\mathbb{E}\sum_{i'=1}^{n+1} \min(f_{i'}(\xi) - f_{i'}(x_{i'}^*), f_{i'}(x_i^*) - f_{i'}(x_{i'}^*)) = \frac{n\frac{L^2}{2\mu} \cdot \frac{\mu}{2} + \frac{L^3+2\mu^2L+\mu^3}{2\mu^2} \cdot n\frac{L}{2}}{n\frac{L^2}{2\mu} + \frac{L^3+2\mu^2L+\mu^3}{2\mu^2}}$$
$$\overset{n\to\infty}{\to} \frac{3}{2}\mu + \frac{L^2}{2\mu} + \frac{\mu^2}{2L}.$$

Besides, we have the minimizer $z^* = \frac{nL}{nL+\mu}$ of the objective function $\sum_{i=1}^{n+1}(f_i(z) - f_i(x_i^*))$. We have

$$\sum_{i=1}^{n+1} (f_i(z^*) - f_i(x_i^*)) = \frac{nL\mu}{2(nL+\mu)} \overset{n\to\infty}{\to} \frac{\mu}{2}.$$

We have the following asymptotic error bound:

$$\lim_{n\to\infty} \frac{\mathbb{E}\sum_{i=1}^{n+1} \min(f_i(\xi) - f_i(x_i^*), f_i(x_i^*) - f_i(x_i^*))}{\sum_{i=1}^{n+1}(f_i(z^*) - f_i(x_i^*))} = 3 + \frac{L^2}{\mu^2} + \frac{\mu}{L}.$$

□

We remark that the orthogonal technique used in the construction of (30) can be applied in other lower bound constructions in the proofs of the initialization Algorithms 1 and 4 as well.

Lemma C.9. *We consider the sum-of-minimum optimization (1). Suppose that S^* is k-separate. Suppose that we have fixed indices $i_1, i_2, \ldots, i_j$. We define the finite set $\mathcal{M}_j = \{\mathbf{x}^*_{i_1}, \mathbf{x}^*_{i_2}, \ldots, \mathbf{x}^*_{i_j}\}$. We define the index sets $L_j = \{l : A_l \cap \{i_1, i_2, \ldots, i_j\} \neq \emptyset\}, L_j^c = \{l : A_l \cap \{i_1, i_2, \ldots, i_j\} = \emptyset\}, \mathcal{I}_j = \cup_{l \in L_j} A_l, \mathcal{I}_j^c = \cup_{l \in L_j^c} A_l$. Let $u = |L_j^c|$. We sample $t \leq u$ new indices. We let $\mathcal{M}^+_{j,s} = \{\mathbf{x}^*_{i_1}, \mathbf{x}^*_{i_2}, \ldots, \mathbf{x}^*_{i_j}, \mathbf{x}^*_{i_{j+1}}, \ldots, \mathbf{x}^*_{i_{j+s}}\}$ for $0 \leq s \leq t$. In each round of sampling, the probability of $i_{j+s}, s > 0$, being sampled as i is $\frac{\mathcal{A}(\{i\}, \mathcal{M}^+_{j,s-1})}{\mathcal{A}([N], \mathcal{M}^+_{j,s-1})}$. Then we have the following bound,*

$$\mathbb{E}\, \mathcal{A}([N], \mathcal{M}^+_{j,t}) \leq \left(\mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + 4\left(\frac{L^2}{\mu^2} + \frac{L}{\mu}\right)\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_{\text{OPT}})\right)(1 + H_t) + \frac{u-t}{u}\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_j). \tag{31}$$

Here, $H_t = 1 + \frac{1}{2} + \cdots + \frac{1}{t}$ is the harmonic sum.

Proof. We prove by induction on $u = |L_j^c|$ and t. We introduce the notation

$$\Phi_j(i) = \mathcal{A}(\{i\}, \mathcal{M}_j) = \min_{\mathbf{z} \in \mathcal{M}_j} (f_i(\mathbf{z}) - f_i(\mathbf{x}^*_i)).$$

We show that if (31) holds for the case $(u-1, t-1)$ and $(u, t-1)$, then it also holds for the case (u, t). We first prove two base cases.

Case 1: $t = 0, u > 0$.

$$\begin{aligned}\mathbb{E}\, \mathcal{A}([N], \mathcal{M}^+_{j,t}) &= \mathcal{A}([N], \mathcal{M}_j) \\ &= \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + \mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_j).\end{aligned}$$

Case 2: $t = 1, u = 1$. With probability $\frac{\mathcal{A}(\mathcal{I}_j, \mathcal{M}_j)}{\mathcal{A}([N], \mathcal{M}_j)}$, the newly sampled index i_{j+1} will lie in $\mathcal{I}_j$, and with probability $\frac{\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_j)}{\mathcal{A}([N], \mathcal{M}_j)}$, it will lie in $\mathcal{I}_j^c$. We have bounds on the conditional expectation

$$\begin{aligned}\mathbb{E}\left(\mathcal{A}([N], \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j\right) &\leq \mathcal{A}([N], \mathcal{M}_j), \\ \mathbb{E}\left(\mathcal{A}([N], \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j^c\right) &= \mathbb{E}\left(\mathcal{A}(\mathcal{I}_j, \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j^c\right) + \mathbb{E}\left(\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j^c\right) \\ &\leq \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + \sum_{i' \in \mathcal{I}_j^c} \frac{\Phi_j(i')}{\sum_{i \in \mathcal{I}_j^c} \Phi_j(i)} \mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_j \cup \{\mathbf{x}^*_{i'}\}) \\ &\overset{(a)}{\leq} \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + \left(\frac{L^2}{\mu} + L\right)\Delta_{\mathcal{I}_j^c} \\ &\overset{(b)}{\leq} \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + 4\left(\frac{L^2}{\mu^2} + \frac{L}{\mu}\right)\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_{\text{OPT}})\end{aligned}$$

Here, (a) holds when applying Lemma C.5. (b) holds since $\mathcal{I}_j^c$ is identical to a certain A_l as $u = 1$ and we apply Lemma C.6. Overall, we have the bound:

$$\begin{aligned}\mathbb{E}\mathcal{A}([N], \mathcal{M}^+_{j,t}) &= \frac{\mathcal{A}(\mathcal{I}_j, \mathcal{M}_j)}{\mathcal{A}([N], \mathcal{M}_j)} \mathbb{E}\left(\mathcal{A}([N], \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j\right) + \frac{\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_j)}{\mathcal{A}([N], \mathcal{M}_j)} \mathbb{E}\left(\mathcal{A}([N], \mathcal{M}^+_{j,t}) \middle| i_{j+1} \in \mathcal{I}_j^c\right) \\ &\leq \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + 4\left(\frac{L^2}{\mu^2} + \frac{L}{\mu}\right)\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_{\text{OPT}}) + \mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) \\ &= 2\mathcal{A}(\mathcal{I}_j, \mathcal{M}_j) + 4\left(\frac{L^2}{\mu^2} + \frac{L}{\mu}\right)\mathcal{A}(\mathcal{I}_j^c, \mathcal{M}_{\text{OPT}})\end{aligned}$$

Next, we prove that the case (u,t) holds when the inequality holds for cases $(u-1,t)$ and $(u-1,t-1)$. With probability $\frac{\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}$, the first sampled index i_{j+1} will lie in $\mathcal{I}_j$, and with probability $\frac{\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}$, it will lie in $\mathcal{I}_j^c$. Let

$$\alpha = 4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right).$$

We divide into two cases and compute the corresponding conditional expectations. For the case where i_{j+1} lies in $\mathcal{I}_j$, we have the following bound on the conditional expectation.

$$\begin{aligned}
&\mathbb{E}\left(\mathcal{A}([N],\mathcal{M}_{j,t}^+)\,\middle|\, i_{j+1}\in\mathcal{I}_j\right)\leq\\
&\quad\mathbb{E}\left(\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j\cup\{\mathbf{x}^*_{i_{j+1}}\})+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t+1}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j\cup\{\mathbf{x}^*_{i_{j+1}}\})\middle| i_{j+1}\in\mathcal{I}_j\right)\\
&\quad\leq\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t+1}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}$$

For the case where i_{j+1} lies in $\mathcal{I}_j^c$, we have the following inequality:

$$\begin{aligned}
&\mathbb{E}\left(\mathcal{A}([N],\mathcal{M}_{j,t}^+)\,\middle|\, i_{j+1}\in\mathcal{I}_j^c\right)\\
&\leq\sum_{l\in L_j^c}\frac{\sum_{i\in A_l}\Phi_j(i)}{\sum_{i'\in\mathcal{I}_j^c}\Phi_j(i')}\Big[\left(\mathcal{A}(\mathcal{I}_j\cup A_l,\mathcal{M}_j\cup\{\mathbf{x}_i^*\})+\alpha\mathcal{A}(\mathcal{I}_j^c\backslash A_l,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})\\
&\qquad+\frac{u-t}{u-1}\mathcal{A}(\mathcal{I}_j^c\backslash A_l,\mathcal{M}_j\cup\{\mathbf{x}_i^*\})\Big]\\
&\leq\sum_{l\in L_j^c}\frac{\sum_{i\in A_l}\Phi_j(i)}{\sum_{i'\in\mathcal{I}_j^c}\Phi_j(i')}\Big[\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\mathcal{A}(A_l,\mathcal{M}_j\cup\{\mathbf{x}_i^*\})+\alpha(\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})-\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}}))\right)(1+H_{t-1})\\
&\qquad+\frac{u-t}{u-1}\left(\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)-\mathcal{A}(A_l,\mathcal{M}_j)\right)\Big]\\
&\overset{(a)}{\leq}\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t}{u-1}\left(\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)-\sum_{l\in L_j^c}\frac{\mathcal{A}(A_l,\mathcal{M}_j)^2}{\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)}\right)\\
&\overset{(b)}{\leq}\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t}{u-1}\left(\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)-\frac{1}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)\right)\\
&=\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}$$

Here, (a) holds when applying Lemma C.5 and Lemma C.6. (b) holds as

$$\sum_{l\in L_j^c}\mathcal{A}(A_l,\mathcal{M}_j)^2\geq\frac{1}{u}\left(\sum_{l\in L_j^c}\mathcal{A}(A_l,\mathcal{M}_j)\right)^2=\frac{1}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)^2.$$

Overall, we have the bound:

$$\begin{aligned}
\mathbb{E}\mathcal{A}([N],\mathcal{M}_{j,t}^+)&=\frac{\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}\mathbb{E}\left(\mathcal{A}([N],\mathcal{M}_{j,t}^+)\middle| i_{j+1}\in\mathcal{I}_j\right)+\frac{\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}\mathbb{E}\left(\mathcal{A}([N],\mathcal{M}_{j,t}^+)\middle| i_{j+1}\in\mathcal{I}_j^c\right)\\
&\leq\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_{t-1})+\frac{u-t}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)+\frac{1}{u}\frac{\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}\\
&\overset{(a)}{\leq}\left(\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)+\alpha\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)(1+H_t)+\frac{u-t}{u}\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}$$

Here, (a) holds since $u\geq t$ and

$$\frac{\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_j)\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j)}{\mathcal{A}([N],\mathcal{M}_j)}\leq\mathcal{A}(\mathcal{I}_j,\mathcal{M}_j).$$

The proof concludes. □

Theorem C.10 (Restatement of Theorem 4.1)**.** *Suppose that the solution set S^* is k-separate. Let*

$$\mathcal{M}_{\text{init}} = \{\mathbf{x}_{i_1}^*, \mathbf{x}_{i_2}^*, \dots, \mathbf{x}_{i_k}^*\}$$

be the initial points sampled by the random initialization Algorithm 1. We have the following bound:

$$\mathbb{E}\,\mathcal{A}([N], \mathcal{M}_{\text{init}}) \le 4(2+\ln k)\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\mathcal{A}([N], \mathcal{M}_{\text{OPT}}). \tag{32}$$

Proof. We start with a fixed index i_1, let $\mathcal{M}_1 = \{\mathbf{x}_{i_1}^*\}$. Suppose $\mathbf{x}_{i_1} \in A_l$. Then we use Lemma C.9 with $u = k-1, t = k-1$. Let

$$\alpha = 4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right).$$

We have

$$\mathbb{E}\,\mathcal{A}([N], \mathcal{M}_{1,k-1}^+) \le \left(\mathcal{A}(A_l, \mathcal{M}_1) + \alpha\mathcal{A}([N]\backslash A_l, \mathcal{M}_{\text{OPT}})\right)(1+H_{k-1})$$

The term $\mathbb{E}\mathcal{A}([N], \mathcal{M}_{1,k-1}^+)$ can be regarded as the conditional expectation of $\mathcal{A}([N], \mathcal{M}_{\text{init}})$ given i_1.

$$\mathbb{E}\mathcal{A}([N], \mathcal{M}_{1,k-1}^+) = \mathbb{E}\left(\mathcal{A}([N], \mathcal{M}_{\text{init}})|i_1\right)$$

According to Algorithm 1, the first index i_1 is uniformly random in $[N]$. We take the expectation over i_1 and get

$$\begin{aligned}
\mathbb{E}\,\mathcal{A}([N], \mathcal{M}_{\text{init}}) &\le \frac{1}{N}\sum_{l\in[k]}\sum_{i\in A_l}\left(\mathcal{A}(A_l, \{\mathbf{x}_i^*\}) + \alpha\mathcal{A}([N]\backslash A_l, \mathcal{M}_{\text{OPT}})\right)(1+H_{k-1}) \\
&= \left(\frac{1}{N}\sum_{l\in[k]}\sum_{i\in A_l}\mathcal{A}(A_l, \{\mathbf{x}_i^*\}) + \alpha\left(\mathcal{A}([N], \mathcal{M}_{\text{OPT}}) - \frac{1}{N}\sum_{l\in[k]}|A_l|\mathcal{A}(A_l, \mathcal{M}_{\text{OPT}})\right)\right)(1+H_{k-1}) \\
&\overset{(a)}{\le} \left(\frac{1}{N}\sum_{l\in[k]}|A_l|\frac{L}{2}\Delta_{A_l} + \alpha\left(\mathcal{A}([N], \mathcal{M}_{\text{OPT}}) - \frac{1}{N}\sum_{l\in[k]}|A_l|\mathcal{A}(A_l, \mathcal{M}_{\text{OPT}})\right)\right)(1+H_{k-1}) \\
&\overset{(b)}{\le} \left(\frac{1}{N}\sum_{l\in[k]}|A_l|\frac{2L}{\mu}\mathcal{A}(A_l, \mathcal{M}_{\text{OPT}}) + \alpha\left(\mathcal{A}([N], \mathcal{M}_{\text{OPT}}) - \frac{1}{N}\sum_{l\in[k]}|A_l|\mathcal{A}(A_l, \mathcal{M}_{\text{OPT}})\right)\right)(1+H_{k-1}) \\
&\le \alpha\mathcal{A}([N], \mathcal{M}_{\text{OPT}})(1+H_{k-1}) \\
&\le 4(2+\ln k)\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\mathcal{A}([N], \mathcal{M}_{\text{OPT}}).
\end{aligned}$$

Here, (a) holds when applying Lemma C.3. (b) holds as a result of Lemma C.6.

□

When we take

$$f_i(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}-\mathbf{x}_i^*\|^2,$$

the optimization problem (1) reduces to the k-means problem, and Algorithm 1 reduces to the k-means++ algorithm. Therefore, according to (Arthur & Vassilvitskii, 2007), the bound given in Theorem C.10 is **tight** in $\ln k$ up to a constant. Next, we give a more detailed lower bound considering the conditioning number $\frac{L}{\mu}$.

Theorem C.11 (Restatement of Theorem 4.2)**.** *Given a fixed cluster number $k > 0$, there exists $N > 0$. We can construct N μ-strongly convex and L-smooth sub-functions $\{f_i\}_{i=1}^N$, whose minimizer set S^* is k-separate. Besides, the sum-of-min objective function F satisfies that $F^* > f^*$, so that $\mathcal{A}([N], \mathcal{M}_{OPT}) > 0$. When we apply Algorithm 1 to sample the initial centers $\mathcal{M}_{\text{init}}$, we have the following error bound:*

$$\mathbb{E}\mathcal{A}([N], \mathcal{M}_{\text{init}}) \ge \frac{1}{2}\frac{L^2}{\mu^2}\ln k\mathcal{A}([N], \mathcal{M}_{\text{OPT}}). \tag{33}$$

Proof. We construct the following problem. We fix the cluster number to be k. We let the dimension to be $2k$. We pick the vertices of a k-simplex as the "centers" of k clusters. The k-simplex is embedded in a $k-1$ dimensional subspace. We let the first k elements of the vertices' coordinates to be non-zero, while the other elements are zero. We denote the first k elements of the l-th vertex by $\xi^{(l)} \in \mathbb{R}^k$. We let the k-simplex be centered at the origin, so that the magnitudes $\|\xi^{(l)}\|$'s are the same. We let m be the edge length of the simplex. The functions in each cluster follows the orthogonal construction technique in (30). Specifically, in cluster l, we construct $n+1$ functions mapping from $\mathbb{R}^{2k}$ to $\mathbb{R}$ as

$$\begin{aligned} f_{i,l}(y) &= \frac{\mu}{2}\|y_{1:d} - \xi^{(l)}\|^2 + \frac{\mu}{2}\sum_{j\geq k+1, j\neq k+l} y_j^2 + \frac{L}{2}(y_{k+l}+1)^2, \quad i = 1,2,\ldots,n, \\ f_{i,l}(y) &= \frac{L}{2}\|y_{1:d} - \xi^{(l)}\|^2 + \frac{L}{2}\sum_{j\geq k+1, j\neq k+l} y_j^2 + \frac{\mu}{2}(y_{k+l}-1)^2, \quad i = n+1. \end{aligned} \tag{34}$$

We have a total of $N = k(n+1)$ sub-functions. We let $m = \exp(n)$, $n \gg 1$, so that $\{f_{i,l}\}_{i=1}^{n+1}$ will be assigned in the same cluster when computing the minimizer of the objective function F. We let $\mathbf{e}_l \in \mathbb{R}^k$ be the l-th unit vector, then the minimizers of the above sub-functions are $\mathbf{x}_{i,l}^* = [\xi^{(l)}; -\mathbf{e}_l](i = 1,2,\ldots,n)$ and $\mathbf{x}_{n+1,l}^* = [\xi^{(l)}; \mathbf{e}_l]$. We let S^* be the set of all the minimizers $\{\mathbf{x}_{1,l}\}_{l=1}^k \cup \{\mathbf{x}_{n+1,l}\}_{l=1}^k$. For each cluster l, we can compute

$$\min_y \sum_{i=1}^{n+1}(f_{i,l}(y) - f_{i,l}^*) = \frac{2nL\mu}{nL+\mu}.$$

Thus, we have

$$\mathcal{A}([N], \mathcal{M}_{\text{OPT}}) = k\frac{2nL\mu}{nL+\mu}.$$

Let $\mathcal{M}$ be a nonempty subset of S^*. We study the optimality gap of F when sampling the new centers based on $\mathcal{M}$. We divide the k clusters into 4 classes as follows:

$$\begin{aligned} C_a &= \{l \,|\, \mathbf{x}_{1,l}^* \in \mathcal{M}, \mathbf{x}_{n+1,l}^* \notin \mathcal{M}\}, \\ C_b &= \{l \,|\, \mathbf{x}_{n+1,l}^* \in \mathcal{M}, \mathbf{x}_{1,l}^* \notin \mathcal{M}\}, \\ C_f &= \{l \,|\, \mathbf{x}_{1,l}^* \in \mathcal{M}, \mathbf{x}_{n+1,l}^* \in \mathcal{M}\}, \\ C_u &= \{l \,|\, \mathbf{x}_{1,l}^* \notin \mathcal{M}, \mathbf{x}_{n+1,l}^* \notin \mathcal{M}\}. \end{aligned}$$

We define $a = |C_a|, b = |C_b|, u = |C_u|$. Consider $\mathcal{M}$ as the existing centers, we continue sampling $t \leq u$ new centers using Algorithm 1. Let $\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_t^*$ be the newly sampled centers. We define the quantity

$$\phi_{a,b,u,t} = \mathbb{E}\mathcal{A}([N], \mathcal{M} \cup \{\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_t^*\}),$$

which is the expected optimality gap after sampling. We will prove by induction that

$$\phi_{a,b,u,t} \geq \alpha^{t+1}\left[\frac{1}{2}\left(n\left(\mu m^2 + \mu + L\right) + \left(Lm^2 + L + \mu\right)\right)(u-t) + (2nLb + 2\mu a)(1+H_u) + \left(\frac{2L^2}{\mu} + 2\mu\right)G_u\right]. \tag{35}$$

Here H_u is the harmonic series. G_u is recursively defined as:

$$G_0 = 0, \quad G_u - G_{u-1} = \beta(1 + H_{u-1}).$$

The parameter $0 < \alpha, \beta < 1$ are chosen as

$$\alpha = 1 - \frac{1}{m}, \quad \beta = 1 - \frac{1}{\sqrt{n}}.$$

We denote the right0hand side of (35) as $\alpha^{t+1}\varphi_{a,b,u,t}$.

We consider the case where $t = 0$, we have

$$\phi_{a,b,u,0} = \frac{1}{2}\left(n\left(\mu m^2 + \mu + L\right) + \left(Lm^2 + L + \mu\right)\right)u + 2nLb + 2\mu a.$$

In the mean while,

$$\varphi_{a,b,u,0} = \frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u + (2nLb+2\mu a)(1+H_u) + \left(\frac{2L^2}{\mu}+2\mu\right)G_u.$$

If $u=0$, we have

$$\phi_{a,b,0,0} = \varphi_{a,b,0,0} \geq \alpha\varphi_{a,b,0,0}.$$

If $u \geq 1$, then $\frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u$ becomes the leading term,

$$\begin{aligned}(1-\alpha)\frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u &\geq \frac{1}{2}nm\mu u \\ &\geq (2nLb+2\mu a)(1+H_u) + \left(\frac{2L^2}{\mu}+2\mu\right)G_u \\ &\geq \alpha\left((2nLb+2\mu a)(1+H_u) + \left(\frac{2L^2}{\mu}+2\mu\right)G_u\right).\end{aligned}$$

Rearrange the left-hand side and the right-hand side of the inequality, we have:

$$\begin{aligned}&\frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u \geq \\ &\quad\alpha\left(\frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u + (2nLb+2\mu a)(1+H_u) + \left(\frac{2L^2}{\mu}+2\mu\right)G_u\right) = \alpha\varphi_{a,b,u,0}.\end{aligned}$$

Therefore, we have

$$\phi_{a,b,u,0} \geq \alpha\varphi_{a,b,u,0}.$$

Next, we induct on t. When $t \geq 1$, we have $u \geq 1$. We use the one-step transfer technique. We let

$$\begin{aligned}K &= \frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right)u + 2\mu a + 2nbL, \\ A &= \frac{1}{2}\left(n\left(\mu m^2+\mu+L\right)+\left(Lm^2+L+\mu\right)\right), \\ B &= \frac{2L^2}{\mu}+2\mu.\end{aligned}$$

We have

$$\begin{aligned}&\phi_{a,b,u,t} \\ =&\frac{n(\mu m^2+\mu+L)u}{2K}\phi_{a+1,b,u-1,t-1} + \frac{(Lm^2+L+\mu)u}{2K}\phi_{a,b+1,u-1,t-1} + \frac{2nLb}{K}\phi_{a,b-1,u,t-1} + \frac{2\mu a}{K}\phi_{a-1,b,u,t-1} \\ \geq&\frac{n(\mu m^2+\mu+L)u}{2K}\alpha^t\left[A(u-t)+(2nLb+2\mu a+2\mu)(1+H_{u-1})+BG_{u-1}\right] \\ &+\frac{(Lm^2+L+\mu)u}{2K}\alpha^t\left[A(u-t)+(2nLb+2\mu a+2nL)(1+H_{u-1})+BG_{u-1}\right] \\ &+\frac{2nLb}{K}\alpha^t\left[A(u-t+1)+(2nLb+2\mu a-2nL)(1+H_u)+BG_u\right] \\ &+\frac{2\mu a}{K}\alpha^t\left[A(u-t+1)+(2nLb+2\mu a-2\mu)(1+H_u)+BG_u\right] \\ =&\frac{n(\mu m^2+\mu+L)u}{2K}\alpha^t\varphi_{a,b,u,t} \\ &+\frac{n(\mu m^2+\mu+L)u}{2K}\alpha^t\left[2\mu(1+H_{u-1})+(2nLb+2\mu a)(H_{u-1}-H_u)+B(G_{u-1}-G_u)\right] \\ &+\frac{(Lm^2+L+\mu)u}{2K}\alpha^t\varphi_{a,b,u,t}\end{aligned}$$

$$
\begin{aligned}
&+\frac{(Lm^2+L+\mu)u}{2K}\alpha^t\left[2nL(1+H_{u-1})+(2nLb+2\mu a)(H_{u-1}-H_u)+B(G_{u-1}-G_u)\right]\\
&+\frac{2nLb}{K}\alpha^t\varphi_{a,b,u,t}+\frac{2nLb}{K}\alpha^t(A-2nL(1+H_u))+\frac{2\mu a}{K}\alpha^t\varphi_{a,b,u,t}+\frac{2\mu a}{K}\alpha^t(A-2\mu(1+H_u))\\
=&\alpha^t\varphi_{a,b,u,t}+\frac{1}{K}\alpha^t\left[\frac{1}{2}n(\mu m^2+\mu+L)u\left(2\mu-\beta\left(2\mu+\frac{2L^2}{\mu}\right)\right)(1+H_{u-1})\right]\\
&+\frac{1}{K}\alpha^t\Bigg[(Lm^2+L+\mu)unL(1+H_{u-1})-\frac{1}{2}(Lm^2+L+\mu)u\beta B(1+H_{u-1})\\
&\qquad-4\mu^2a(1+H_u)-4n^2L^2b(1+H_u)\Bigg]\\
=&\alpha^t\varphi_{a,b,u,t}+\frac{1}{K}\alpha^t\left[-\frac{1}{2}\left(n(\mu m^2+\mu+L)+(Lm^2+L+\mu)\right)(2nLb+2\mu a)+A(2nLb+2\mu a)\right]\\
&+\frac{1}{K}\alpha^t\left[\frac{1}{2}n(\mu m^2+\mu+L)u\left(-\frac{2L^2}{\mu}+\frac{1}{\sqrt{n}}(2\mu+\frac{2L^2}{\mu})\right)(1+H_{u-1})+(Lm^2+L+\mu)unL(1+H_{u-1})\right]\\
&+\frac{1}{K}\alpha^t\left[-\frac{1}{2}(Lm^2+L+\mu)u\beta B(1+H_{u-1})-4\mu^2a(1+H_u)-4n^2L^2b(1+H_u)\right]\\
=&\alpha^t\varphi_{a,b,u,t}+\frac{1}{K}\alpha^t\sqrt{n}m^2u(1+H_{u-1})(\mu^2+L^2)\\
&+\frac{1}{K}\alpha^t\Bigg[n(\mu+L)u\left(L-\frac{L^2}{\mu}+\frac{1}{\sqrt{n}}\left(\mu+\frac{L^2}{\mu}\right)\right)(1+H_{u-1})-\frac{1}{2}(Lm^2+L+\mu)u\beta B(1+H_{u-1})\\
&\qquad-4\mu^2a(1+H_u)-4n^2L^2b(1+H_u)\Bigg]\\
\overset{(a)}{\geq}&\alpha^t\varphi_{a,b,u,t}\\
\geq&\alpha^{t+1}\varphi_{a,b,u,t}.
\end{aligned}
$$

For (a), we have

$$
\begin{aligned}
\sqrt{n}m^2u(1+H_{u-1})(\mu^2+L^2)\geq& n(\mu+L)u\left(L-\frac{L^2}{\mu}+\frac{1}{\sqrt{n}}\left(\mu+\frac{L^2}{\mu}\right)\right)(1+H_{u-1})\\
&-\frac{1}{2}(Lm^2+L+\mu)u\beta B(1+H_{u-1})-4\mu^2a(1+H_u)-4n^2L^2b(1+H_u).
\end{aligned}
$$

when $m=\exp(n)$ and $n\gg 1$.

Thus the inequality (35) holds. Let $u=t=k-1$. We have

$$
\phi_{a,b,k-1,k-1}\geq\alpha^k\left[(2nLb+2\mu a)(1+H_{k-1})+\left(\frac{2L^2}{\mu}+2\mu\right)G_{k-1}\right].
$$

Let $n\geq 100k^2$. Since $m=\exp(n)\geq 100k^2$, then

$$
\alpha^k\geq\frac{3}{4},\quad \beta=1-\frac{1}{10k}\geq\frac{9}{10}.
$$

$$
\phi_{a,b,t-1,t-1}\geq\frac{3}{4}\left(\frac{2L^2}{\mu}+2\mu\right)G_{k-1}.
$$

We have the following inequalities:

$$
\begin{aligned}
H_{k-1}=1+\frac{1}{2}+\cdots+\frac{1}{k-1}\geq&\int_1^k\frac{1}{t}\,dt=\ln k,\quad k\geq 1,\\
G_k=\beta\sum_{j=0}^{k-1}(1+H_j)\geq&\beta\left(k+\sum_{j=1}^{k}\ln j\right)\geq\beta\left(k+\int_{t=1}^{k}\ln t\,dt\right)=\beta(k\ln k+1).
\end{aligned}
$$

Therefore, we have

$$\mathbb{E}\mathcal{A}([N],\mathcal{M}_{\text{init}}) \geq \frac{1}{2}k\ln k\left(\frac{2L^2}{\mu}+2\mu\right) = k\ln k\left(\frac{L^2}{\mu}+\mu\right).$$

In the meanwhile, we have an upper bound estimate for $\mathcal{A}([N],\mathcal{M}_{\text{OPT}})$. We pick $\mathcal{M}_\xi = \{[\xi^{(l)};-\mathbf{e}_l]\}_{l=1}^k$ as the centers. We have

$$\mathcal{A}([N],\mathcal{M}_{\text{OPT}}) \leq \mathcal{A}([N],\mathcal{M}_\xi) = 2k\mu.$$

Thus,

$$\mathbb{E}\mathcal{A}([N],\mathcal{M}_{\text{init}}) \geq \frac{1}{2}\ln k\frac{L^2}{\mu^2}\mathcal{A}([N],\mathcal{M}_{\text{OPT}}).$$

□

We prove two different error bounds when the estimate of $f_i(\mathbf{z})-f_i(\mathbf{x}_i^*)$ is not accurate. We consider the additive and multiplicative errors on the oracle $f_i(\mathbf{z})-f_i(\mathbf{x}_i^*)$.

In Algorithm 1, when computing the score $v_i^{(j)}$, we suppose we do not have the exact f_i^*, instead, we have an estimate $\tilde{f}_i^*$, such that

$$|\tilde{f}_i^* - f_i^*| \leq \epsilon$$

for a certain error factor $\epsilon>0$. We define

$$\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M}) = \sum_{i\in\mathcal{I}}\max\left(\min_{\mathbf{z}\in\mathcal{M}}\left(f_i(\mathbf{z})-\tilde{f}_i^*\right),0\right) = \sum_{i\in\mathcal{I}}\min_{\mathbf{z}\in\mathcal{M}}\left(\max\left(f_i(\mathbf{z})-\tilde{f}_i^*,0\right)\right).$$

Lemma C.12. *Let $\mathcal{I}$ be an index set, and $\mathcal{M}$ be a finite point set. Suppose that $\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})>0$. We sample an index $i\in\mathcal{I}$ with probability $\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M})}{\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})}$, then we have the following inequality:*

$$\mathbb{E}\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\}) \leq |\mathcal{I}|\left(1+\frac{4L}{\mu}\right)\epsilon + 4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\min_{\mathbf{z}}\sum_{i\in\mathcal{I}}(f_i(\mathbf{z})-f_i(\mathbf{x}_i^*)). \tag{36}$$

Proof. We have

$$\begin{aligned}
\tilde{\mathcal{A}}(\{i\},\mathcal{M}) &= \max\left(\min_{\mathbf{z}\in\mathcal{M}}(f_i(\mathbf{z})-\tilde{f}_i^*),0\right)\\
&\leq \epsilon + \min_{\mathbf{z}\in\mathcal{M}}(f_i(\mathbf{z})-f_i^*)\\
&\leq \epsilon + \frac{L}{2}\min_{\mathbf{z}\in\mathcal{M}}\|\mathbf{z}-\mathbf{x}_i^*\|^2\\
&\leq \epsilon + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + L\min_{\mathbf{z}\in\mathcal{M}}\|\mathbf{z}-\mathbf{x}_{i'}^*\|^2\\
&\leq \epsilon + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + \frac{2L}{\mu}\min_{\mathbf{z}\in\mathcal{M}}(f_{i'}(\mathbf{z})-f_{i'}(\mathbf{x}_{i'}^*))\\
&\leq \left(1+\frac{2L}{\mu}\right)\epsilon + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + \frac{2L}{\mu}\min_{\mathbf{z}\in\mathcal{M}}(f_{i'}(\mathbf{z})-\tilde{f}_{i'}^*)\\
&\leq \left(1+\frac{2L}{\mu}\right)\epsilon + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + \frac{2L}{\mu}\max\left(\min_{\mathbf{z}\in\mathcal{M}}(f_{i'}(\mathbf{z})-\tilde{f}_{i'}^*),0\right)\\
&= \left(1+\frac{2L}{\mu}\right)\epsilon + L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2 + \frac{2L}{\mu}\tilde{\mathcal{A}}(\{i'\},\mathcal{M}).
\end{aligned}$$

We have

$$\begin{aligned}
&\mathbb{E}\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\})\\
=&\sum_{i\in\mathcal{I}}\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M})}{\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})}\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M}\cup\{\mathbf{x}_i^*\})
\end{aligned}$$

$$
\begin{aligned}
=&\sum_{i\in\mathcal{I}}\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M})}{\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})}\sum_{i''\in\mathcal{I}}\tilde{\mathcal{A}}(\{i''\},\mathcal{M}\cup\{\mathbf{x}_i^*\})\\
=&\sum_{i\in\mathcal{I}}\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M})}{\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})}\sum_{i''\in\mathcal{I}}\min(\tilde{\mathcal{A}}(\{i''\},\mathcal{M}),\max(f_{i''}(\mathbf{x}_i^*)-\tilde{f}_{i''}^*,0))\\
\leq&\sum_{i\in\mathcal{I}}\frac{\frac{1}{|\mathcal{I}|}\sum_{i'\in\mathcal{I}}\left\{\left(1+\frac{2L}{\mu}\right)\epsilon+L\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2+\frac{2L}{\mu}\tilde{\mathcal{A}}(\{i'\},\mathcal{M})\right\}}{\tilde{\mathcal{A}}(\mathcal{I},\mathcal{M})}\sum_{i''\in\mathcal{I}}\min(\tilde{\mathcal{A}}(\{i''\},\mathcal{M}),\max(f_{i''}(\mathbf{x}_i^*)-\tilde{f}_{i''}^*,0))\\
\leq&|\mathcal{I}|\left(1+\frac{2L}{\mu}\right)\epsilon+L\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i'\in\mathcal{I}}\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2+\frac{2L}{\mu}\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i''\in\mathcal{I}}\max(f_{i''}(\mathbf{x}_i^*)-\tilde{f}_{i''}^*,0)\\
\leq&|\mathcal{I}|\left(1+\frac{4L}{\mu}\right)\epsilon+\left(L+\frac{L^2}{\mu}\right)\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\sum_{i'\in\mathcal{I}}\|\mathbf{x}_i^*-\mathbf{x}_{i'}^*\|^2\\
=&|\mathcal{I}|\left(1+\frac{4L}{\mu}\right)\epsilon+2\left(L+\frac{L^2}{\mu}\right)\min_{\mathbf{z}}\sum_{i\in\mathcal{I}}\|\mathbf{x}_i^*-\mathbf{z}\|^2\\
\leq&|\mathcal{I}|\left(1+\frac{4L}{\mu}\right)\epsilon+4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\min_{\mathbf{z}}\sum_{i\in\mathcal{I}}(f_i(\mathbf{z})-f_i(\mathbf{x}_i^*)).
\end{aligned}
$$

□

Lemma C.13. *Suppose that we have fixed indices $i_1,i_2,\ldots,i_j$. We define the finite set $\mathcal{M}_j=\{x_{i_1}^*,x_{i_2}^*,\ldots,x_{i_j}^*\}$. We define the index sets $L_j=\{l:A_l\cap\{i_1,i_2,\ldots,i_j\}\neq\emptyset\},L_j^c=\{l:A_l\cap\{i_1,i_2,\ldots,i_j\}=\emptyset\},\mathcal{I}_j=\cup_{l\in L_j}A_l,\mathcal{I}_j^c=\cup_{l\in L_j^c}A_l$. Let $u=|L_j^c|$. Suppose that $u>0$. We sample $t\leq u$ new indices. We let $\mathcal{M}_{j,s}^+=\{x_{i_1}^*,x_{i_2}^*,\ldots,x_{i_j}^*,x_{i_{j+1}}^*,\ldots,x_{i_{j+s}}^*\}$ for $0\leq s\leq t$. In each round of sampling, the probability of $i_{j+s},s>0$, being sampled as i is $\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M}_{j,s-1}^+)}{\tilde{\mathcal{A}}([N],\mathcal{M}_{j,s-1}^+)}$. Then we have the following bound:*

$$
\mathbb{E}\tilde{\mathcal{A}}([N],\mathcal{M}_{j,t}^+)\leq(1+H_t)\left[\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\left(1+\frac{4L}{\mu}\right)\epsilon+4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right]+\frac{u-t}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j). \tag{37}
$$

Proof. The key idea of the proof is similar to Lemma C.9. We let

$$
\alpha=1+\frac{4L}{\mu},\quad\beta=4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right).
$$

We prove by induction. When $t=0$, the inequality obviously holds. When $t>0,u=1$, we have the inequality:

$$
\begin{aligned}
\mathbb{E}\tilde{\mathcal{A}}([N],\mathcal{M}_{j,t}^+)\leq&\frac{\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}\tilde{\mathcal{A}}([N],\mathcal{M}_j)+\frac{\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}\left(\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right)\\
\leq&\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})+\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}
$$

For the general (t,u) case, $\mathbb{E}\tilde{\mathcal{A}}([N],\mathcal{M}_{j,t}^+)$ can be bounded by two parts. With probability $\frac{\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}$, the first sampled index lies in $\mathcal{I}_j$, and the conditional expectation is bounded by:

$$
(1+H_{t-1})\left[\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\right]+\frac{u-t+1}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j).
$$

With probability $\frac{\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}$, the first sampled index lies in $\mathcal{I}_j^c$. The conditional expectation is bounded by:

$$
\sum_{l\in L^c}\frac{\tilde{\mathcal{A}}(A_l,\mathcal{M}_j)}{\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)}\sum_{i\in A_l}\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M}_j)}{\tilde{\mathcal{A}}(A_l,\mathcal{M}_j)}\left\{(1+H_{t-1})\left(\tilde{\mathcal{A}}(\mathcal{I}_j\cup A_l,\mathcal{M}_j\cup\{\mathbf{x}_i^*\})+|\mathcal{I}_j^c\backslash A_l|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c\backslash A_l,\mathcal{M}_{\text{OPT}})\right)\right.
$$

$$
\begin{aligned}
&+\frac{u-t}{u-1}\tilde{\mathcal{A}}(\mathcal{I}_j^c\backslash A_l, \mathcal{M}_j\cup\{\mathbf{x}_i^*\})\Big\}\\
\le&\sum_{l\in L^c}\frac{\tilde{\mathcal{A}}(A_l,\mathcal{M}_j)}{\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)}\sum_{i\in A_l}\frac{\tilde{\mathcal{A}}(\{i\},\mathcal{M}_j)}{\tilde{\mathcal{A}}(A_l,\mathcal{M}_j)}\Big\{(1+H_{t-1})\Big(\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+\tilde{\mathcal{A}}(A_l,\mathcal{M}_j\cup\{\mathbf{x}_i^*\})+|\mathcal{I}_j^c\backslash A_l|\alpha\epsilon\\
&+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})-\beta\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}})\Big)+\frac{u-t}{u-1}(\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)-\tilde{\mathcal{A}}(A_l,\mathcal{M}_j))\Big\}\\
\le&(1+H_{t-1})\Big(\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\Big)+\frac{u-t}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}
$$

Overall, we have the following inequality:

$$
\begin{aligned}
\mathbb{E}\tilde{\mathcal{A}}([N],\mathcal{M}_{j,t}^+)\le&\frac{\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}\Big\{(1+H_{t-1})\Big[\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\Big]+\frac{u-t+1}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)\Big\}\\
&+\frac{\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)}{\tilde{\mathcal{A}}([N],\mathcal{M}_j)}\Big\{(1+H_{t-1})\Big(\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\Big)+\frac{u-t}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j)\Big\}\\
\le&(1+H_t)\Big(\tilde{\mathcal{A}}(\mathcal{I}_j,\mathcal{M}_j)+|\mathcal{I}_j^c|\alpha\epsilon+\beta\mathcal{A}(\mathcal{I}_j^c,\mathcal{M}_{\text{OPT}})\Big)+\frac{u-t}{u}\tilde{\mathcal{A}}(\mathcal{I}_j^c,\mathcal{M}_j).
\end{aligned}
$$

□

Theorem C.14 (Restatement of Theorem 4.3)**.** *Suppose that the solution set S^* is $(k,\sqrt{\frac{2\epsilon}{\mu}})$-separate. Let*

$$\mathcal{M}_{\text{init}}=\{\mathbf{x}_{i_1}^*,\mathbf{x}_{i_2}^*,\dots,\mathbf{x}_{i_k}^*\}$$

be the initial points sampled by the random initialization Algorithm 1 with noisy oracles $\tilde{f}_i^*$. We have the following bound:

$$\frac{1}{N}\mathbb{E}\mathcal{A}([N],\mathcal{M}_{\text{init}})\le\epsilon+(2+\ln k)\left(1+\frac{4L}{\mu}\right)\epsilon+4(2+\ln k)\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right)\frac{1}{N}\mathcal{A}([N],\mathcal{M}_{\text{OPT}}).$$

Proof. The proof is similar to that of Theorem C.10. We let

$$\alpha=1+\frac{4L}{\mu},\quad \beta=4\left(\frac{L^2}{\mu^2}+\frac{L}{\mu}\right).$$

We fix the first index i_1. Suppose that i_1 lies in A_l, we have

$$\mathbb{E}\,\tilde{\mathcal{A}}([N],\mathcal{M}_{1,k-1}^+)\le\Big(\tilde{\mathcal{A}}(A_l,\{\mathbf{x}_{i_1}^*\})+|[N]\backslash A_l|\alpha\epsilon+\beta\mathcal{A}([N]\backslash A_l,\mathcal{M}_{\text{OPT}})\Big)(1+H_{k-1}).$$

We have

$$
\begin{aligned}
\mathbb{E}\,\tilde{\mathcal{A}}([N],\mathcal{M}_{\text{init}})\le&\left(\frac{1}{N}\sum_{l\in[k]}\sum_{i\in A_l}\tilde{\mathcal{A}}(A_l,\{\mathbf{x}_i^*\})+N\alpha\epsilon-\frac{1}{N}\sum_{l\in[k]}|A_l|^2\alpha\epsilon\right.\\
&\left.+\beta\left(\mathcal{A}([N],\mathcal{M}_{\text{OPT}})-\frac{1}{N}\sum_{l\in[k]}|A_l|\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}})\right)\right)(1+H_{k-1})\\
\overset{(a)}{\le}&\left(\frac{1}{N}\sum_{l\in[k]}\left(|A_l|^2\epsilon+\frac{L}{2}|A_l|\Delta_{A_l}\right)+N\alpha\epsilon-\frac{1}{N}\sum_{l\in[k]}|A_l|^2\alpha\epsilon\right.\\
&\left.+\beta\left(\mathcal{A}([N],\mathcal{M}_{\text{OPT}})-\frac{1}{N}\sum_{l\in[k]}|A_l|\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}})\right)\right)(1+H_{k-1})\\
\overset{(b)}{\le}&\left(\frac{1}{N}\sum_{l\in[k]}\left(|A_l|^2\epsilon+\frac{2L}{\mu}|A_l|\mathcal{A}(A_l,\mathcal{M}_{\text{OPT}})\right)+N\alpha\epsilon-\frac{1}{N}\sum_{l\in[k]}|A_l|^2\alpha\epsilon\right.
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \beta \left(\mathcal{A}([N], \mathcal{M}_{\text{OPT}}) - \frac{1}{N} \sum_{l \in [k]} |A_l| \mathcal{A}(A_l, \mathcal{M}_{\text{OPT}}) \right) \Bigg) (1 + H_{k-1}) \\
&\leq \left(N\alpha\epsilon + \beta \mathcal{A}([N], \mathcal{M}_{\text{OPT}}) \right) (1 + H_{k-1}) \\
&\leq (2 + \ln k) \left(1 + \frac{4L}{\mu} \right) N\epsilon + 4(2 + \ln k) \left(\frac{L^2}{\mu^2} + \frac{L}{\mu} \right) \mathcal{A}([N], \mathcal{M}_{\text{OPT}}).
\end{aligned}
$$

Here, (a) holds when applying Lemma C.3. (b) holds when applying

$$
\frac{L}{2} \Delta_{A_l} = L \min_{\mathbf{z}} \sum_{i \in A_l} \|\mathbf{x}_i^* - \mathbf{z}\|^2 \leq \frac{2L}{\mu} \min_{\mathbf{z}} \sum_{i \in A_l} (f_i(\mathbf{z}) - f_i^*) = \frac{2L}{\mu} \mathcal{A}(A_l, \mathcal{M}_{\text{OPT}}^{(D)}).
$$

Therefore, we have

$$
\begin{aligned}
\mathbb{E}\mathcal{A}([N], \mathcal{M}_{\text{init}}) &\leq N\epsilon + (2 + \ln k) \left(1 + \frac{4L}{\mu} \right) N\epsilon + 4(2 + \ln k) \left(\frac{L^2}{\mu^2} + \frac{L}{\mu} \right) \mathcal{A}([N], \mathcal{M}_{\text{OPT}}), \\
\frac{1}{N} \mathbb{E}\mathcal{A}([N], \mathcal{M}_{\text{init}}) &\leq \epsilon + (2 + \ln k) \left(1 + \frac{4L}{\mu} \right) \epsilon + 4(2 + \ln k) \left(\frac{L^2}{\mu^2} + \frac{L}{\mu} \right) \frac{1}{N} \mathcal{A}([N], \mathcal{M}_{\text{OPT}}).
\end{aligned}
$$

□

The proof of Theorem 4.4 is similar to the proof of Theorem 4.3, we skip the details here.

D. Convergence of Lloyd's algorithm

In this section, we provide a convergence analysis for Algorithms 2 and 3.

Theorem D.1 (Restatement of Theorem 4.6)**.** *In Algorithm 2, we take the step size $\gamma = \frac{1}{L}$. If f_i are L-smooth, we have the following convergence result:*

$$
\frac{1}{T+1} \sum_{t=0}^{T} \sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{2L}{T+1} \left(F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}) - F^\star \right).
$$

Here, $F^\star$ is the minimum of F.

Proof. According to the L-smoothness assumption on f_i, $F_j^{(t)}$ is also L-smooth, which implies that

$$
\begin{aligned}
F_j^{(t)}(\mathbf{x}_j^{(t+1)}) &\leq F_j^{(t)}(\mathbf{x}_j^{(t)}) + \langle \nabla F_j^{(t)}(\mathbf{x}_j^{(t)}), \mathbf{x}_j^{(t+1)} - \mathbf{x}_j^{(t)} \rangle + \frac{L}{2} \|\mathbf{x}_j^{(t+1)} - \mathbf{x}_j^{(t)}\|^2 \\
&= F_j^{(t)}(\mathbf{x}_j^{(t)}) - \frac{1}{2L} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2, \\
\frac{1}{2L} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 &\leq F_j^{(t)}(\mathbf{x}_j^{(t)}) - F_j^{(t)}(\mathbf{x}_j^{(t+1)}), \\
\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 &\leq 2L \left(F_j^{(t)}(\mathbf{x}_j^{(t)}) - F_j^{(t)}(\mathbf{x}_j^{(t+1)}) \right).
\end{aligned}
$$

Averaging over $\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2$ with weights $|C_j^{(t)}|/N$, we have

$$
\sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq 2L \left(F(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_k^{(t)}) - F(\mathbf{x}_1^{(t+1)}, \mathbf{x}_2^{(t+1)}, \ldots, \mathbf{x}_k^{(t+1)}) \right).
$$

Averaging over t from 0 to T, we have

$$
\frac{1}{T+1} \sum_{t=0}^{T} \sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{2L}{T+1} \left(F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \ldots, \mathbf{x}_k^{(0)}) - F^\star \right).
$$

□

Next, we present a convergence theorem for the momentum algorithm. For simplification, we use the notation

$$\mathbf{U}^{(t)} = (\mathbf{u}_1^{(t)}, \mathbf{u}_2^{(t)}, \dots, \mathbf{u}_k^{(t)}).$$

We have the following convergence theorem:

Theorem D.2 (Restatement of Theorem 4.7)**.** *Consider Algorithm 3. Suppose that Assumption 2.1 holds,* $\alpha > 1$*, and*

$$\gamma \leq \min\left(\frac{1-\beta}{2L}, \frac{(1-\beta)^{\frac{3}{2}}(1-\alpha\beta)^{\frac{1}{2}}}{2\alpha^{\frac{1}{2}}L\beta}\right).$$

Then it holds that

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{2(1-\beta)}{\gamma}\cdot\frac{F(\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_k^{(0)}) - F^*}{T}.$$

Proof. The variable $\mathbf{u}_j^{(t)}$ satisfies the following property,

$$\begin{aligned}
\mathbf{u}_j^{(t+1)} - \mathbf{u}_j^{(t)} &= \frac{1}{1-\beta}\left((\mathbf{x}_j^{(t+1)} - \mathbf{x}_j^{(t)}) - \beta(\mathbf{x}_j^{(t)} - \mathbf{x}_j^{(t-1)})\right) \\
&= \frac{-\gamma}{1-\beta}\left(\mathbf{m}_j^{(t)} - \beta\mathbf{m}_j^{(t-1)}\right) \\
&= \frac{-\gamma}{1-\beta}\nabla F_j^{(t)}(\mathbf{x}_j^{(t)}).
\end{aligned}$$

We have the following inequality:

$$\begin{aligned}
F_j^{(t)}(\mathbf{u}_j^{(t+1)}) &\leq F_j^{(t)}(\mathbf{u}_j^{(t)}) + \langle\nabla F_j^{(t)}(\mathbf{u}_j^{(t)}), \mathbf{u}_j^{(t+1)} - \mathbf{u}_j^{(t)}\rangle + \frac{L}{2}\|\mathbf{u}_j^{(t+1)} - \mathbf{u}_j^{(t)}\|^2 \\
&= F_j^{(t)}(\mathbf{u}_j^{(t)}) - \frac{\gamma}{1-\beta}\langle\nabla F_j^{(t)}(\mathbf{u}_j^{(t)}), \nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\rangle + \frac{L}{2}\frac{\gamma^2}{(1-\beta)^2}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\
&= F_j^{(t)}(\mathbf{u}_j^{(t)}) - \frac{\gamma}{1-\beta}\langle\nabla F_j^{(t)}(\mathbf{u}_j^{(t)}) - \nabla F_j^{(t)}(\mathbf{x}_j^{(t)}), \nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\rangle \\
&\quad + \left(\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2} - \frac{\gamma}{1-\beta}\right)\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\
&\leq F_j^{(t)}(\mathbf{u}_j^{(t)}) + \left(\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2} - \frac{\gamma}{1-\beta}\right)\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\
&\quad + \frac{\gamma}{1-\beta}\frac{\epsilon}{2}\|\nabla F_j^{(t)}(\mathbf{u}_j^{(t)}) - \nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 + \frac{\gamma}{1-\beta}\frac{1}{2\epsilon}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\
&\leq F_j^{(t)}(\mathbf{u}_j^{(t)}) + \left(\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2} - \frac{\gamma}{1-\beta} + \frac{1}{2\epsilon}\frac{\gamma}{1-\beta}\right)\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 + \frac{\epsilon}{2}\frac{L^2\beta^2\gamma^3}{(1-\beta)^3}\|\mathbf{m}_j^{(t-1)}\|^2.
\end{aligned}$$

Rearranging the inequality, we have

$$\left(\frac{\gamma}{1-\beta} - \frac{L}{2}\frac{\gamma^2}{(1-\beta)^2} - \frac{1}{2\epsilon}\frac{\gamma}{1-\beta}\right)\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq F_j^{(t)}(\mathbf{u}_j^{(t)}) - F_j^{(t)}(\mathbf{u}_j^{(t+1)}) + \frac{\epsilon}{2}\frac{L^2\beta^2\gamma^3}{(1-\beta)^3}\|\mathbf{m}_j^{(t-1)}\|^2.$$

We sum over $j = 1, 2, \dots, k$ with weights $\frac{|C_j|}{N}$ and get

$$\begin{aligned}
&\left(\frac{\gamma}{1-\beta} - \frac{L}{2}\frac{\gamma^2}{(1-\beta)^2} - \frac{1}{2\epsilon}\frac{\gamma}{1-\beta}\right)\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\
&\qquad \leq F(\mathbf{U}^{(t)}) - \sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}F_j^{(t)}(\mathbf{u}_j^{(t+1)}) + \frac{\epsilon}{2}\frac{L^2\beta^2\gamma^3}{(1-\beta)^3}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\mathbf{m}_j^{(t-1)}\|^2.
\end{aligned}$$

Since

$$\sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} F_j^{(t)}(\mathbf{u}_j^{(t+1)}) = \frac{1}{N} \sum_{j=1}^{k} \sum_{i \in C_j^{(t)}} f_i(\mathbf{u}_j^{(t+1)}) \geq F(\mathbf{U}^{(t+1)}),$$

we have

$$\left(\frac{\gamma}{1-\beta} - \frac{L}{2} \frac{\gamma^2}{(1-\beta)^2} - \frac{1}{2\epsilon} \frac{\gamma}{1-\beta} \right) \sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\ \leq F(\mathbf{U}^{(t)}) - F(\mathbf{U}^{(t+1)}) + \frac{\epsilon}{2} \frac{\alpha L^2 \beta^2 \gamma^3}{(1-\beta)^3} \sum_{j=1}^{k} \frac{|C_j^{(t-1)}|}{N} \|\mathbf{m}_j^{(t-1)}\|^2.$$

Summing both sides from $t = 1$ to T, then dividing both sides by T, we have

$$\begin{aligned} &\left(\frac{\gamma}{1-\beta} - \frac{L}{2} \frac{\gamma^2}{(1-\beta)^2} - \frac{1}{2\epsilon} \frac{\gamma}{1-\beta} \right) \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{k} \frac{|C_j^{(t)}|}{N} \|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \\ &\leq \frac{F(\mathbf{U}^{(1)}) - F(\mathbf{U}^{(T+1)})}{T} + \frac{\epsilon}{2} \frac{\alpha L^2 \beta^2 \gamma^3}{(1-\beta)^3} \frac{1}{T} \sum_{t=1}^{T} \sum_{j=1}^{k} \frac{|C_j^{(t-1)}|}{N} \|\mathbf{m}_j^{(t-1)}\|^2. \end{aligned} \tag{38}$$

Now, we consider the average term $\frac{1}{T} \sum_{t=1}^{T} \frac{|C_j^{(t)}|}{N} \|\mathbf{m}_j^{(t)}\|^2$. For $\mathbf{m}_j^{(t)}$, we have

$$\begin{aligned} \mathbf{m}_j^{(t)} &= \beta \mathbf{m}_j^{(t-1)} + \nabla F_j^{(t)}(\mathbf{x}_j^{(t)}) \\ &= \beta^t \mathbf{m}_j^{(0)} + \sum_{l=0}^{t-1} \beta^l \nabla F_j^{(t-l)}(\mathbf{x}_j^{(t-l)}) \\ &= \sum_{l=1}^{t} \beta^{t-l} \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}). \end{aligned}$$

We have the following bound on the squared norm of $\mathbf{m}_j^{(t)}$:

$$\begin{aligned} \|\mathbf{m}_j^{(t)}\|^2 &= \left\| \sum_{l=1}^{t} \beta^{t-l} \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2 \\ &= \left(\sum_{s=1}^{t} \beta^{t-s} \right)^2 \left\| \sum_{l=1}^{t} \frac{\beta^{t-l}}{\sum_{s=1}^{t} \beta^{t-s}} \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2 \\ &\overset{(a)}{\leq} \left(\sum_{s=1}^{t} \beta^{t-s} \right)^2 \sum_{l=1}^{t} \frac{\beta^{t-l}}{\sum_{s=1}^{t} \beta^{t-s}} \left\| \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2 \\ &\leq \frac{1}{1-\beta} \sum_{l=1}^{t} \beta^{t-l} \left\| \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2. \end{aligned}$$

Here, (a) applies as an instance of Jensen's inequality. Averaging the above inequality over $t = 1, 2, \ldots, T$, we obtain

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^{T} \frac{|C_j^{(t)}|}{N} \|\mathbf{m}_j^{(t)}\|^2 &\leq \frac{1}{1-\beta} \frac{1}{N} \frac{1}{T} \sum_{t=1}^{T} \sum_{l=1}^{t} |C_j^{(t)}| \beta^{t-l} \left\| \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2 \\ &\leq \frac{1}{T} \frac{1}{N} \frac{1}{1-\beta} \sum_{l=1}^{T} \sum_{t=l}^{T} |C_j^{(t)}| \beta^{t-l} \left\| \nabla F_j^{(l)}(\mathbf{x}_j^{(l)}) \right\|^2 \end{aligned}$$

$$\leq \frac{1}{T}\frac{1}{N}\frac{1}{1-\beta}\sum_{l=1}^{T}\sum_{t=l}^{T}|C_j^{(l)}|\alpha^{t-l}\beta^{t-l}\left\|\nabla F_j^{(l)}(\mathbf{x}_j^{(l)})\right\|^2$$
$$\leq \frac{1}{T}\frac{1}{1-\beta}\frac{1}{1-\alpha\beta}\sum_{l=1}^{T}\frac{|C_j^{(l)}|}{N}\left\|\nabla F_j^{(l)}(\mathbf{x}_j^{(l)})\right\|^2$$

Substituting the above inequality back into (38), we obtain

$$\left(\frac{\gamma}{1-\beta}-\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2}-\frac{1}{2\epsilon}\frac{\gamma}{1-\beta}-\frac{\epsilon}{2}\frac{\alpha L^2\beta^2\gamma^3}{(1-\beta)^4(1-\alpha\beta)}\right)\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{F(\mathbf{U}^{(1)})-F(\mathbf{U}^{(T+1)})}{T}.$$

We choose

$$\epsilon=\frac{(1-\beta)^{\frac{3}{2}}(1-\alpha\beta)^{\frac{1}{2}}}{\gamma\beta L\alpha^{\frac{1}{2}}}$$

and rearrange the above inequality. Thus, we have

$$\left(\frac{\gamma}{1-\beta}-\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2}-\frac{\alpha^{\frac{1}{2}}L\beta\gamma^2}{2(1-\beta)^{\frac{5}{2}}(1-\alpha\beta)^{\frac{1}{2}}}\right)\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{F(\mathbf{U}^{(1)})-F(\mathbf{U}^{(T+1)})}{T}.$$

Since we initialize $\mathbf{m}_j^{(0)}=\mathbf{0}$, we have

$$\mathbf{x}_j^{(1)}=\mathbf{x}_j^{(0)}-\gamma\mathbf{m}_j^{(0)}=\mathbf{x}_j^{(0)},$$
$$\mathbf{u}_j^{(1)}=\mathbf{x}_j^{(0)}.$$

Besides, since $F(\mathbf{U}^{(T+1)})\geq F^*$, we have

$$\left(\frac{\gamma}{1-\beta}-\frac{L}{2}\frac{\gamma^2}{(1-\beta)^2}-\frac{\alpha^{\frac{1}{2}}L\beta\gamma^2}{2(1-\beta)^{\frac{5}{2}}(1-\alpha\beta)^{\frac{1}{2}}}\right)\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{F(\mathbf{x}_1^{(0)},\mathbf{x}_2^{(0)},\ldots,\mathbf{x}_k^{(0)})-F^*}{T}.$$

When

$$\gamma\leq\min\left(\frac{1-\beta}{2L},\frac{(1-\beta)^{\frac{3}{2}}(1-\alpha\beta)^{\frac{1}{2}}}{2\alpha^{\frac{1}{2}}L\beta}\right),$$

we have

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\frac{|C_j^{(t)}|}{N}\|\nabla F_j^{(t)}(\mathbf{x}_j^{(t)})\|^2 \leq \frac{2(1-\beta)}{\gamma}\cdot\frac{F(\mathbf{x}_1^{(0)},\mathbf{x}_2^{(0)},\ldots,\mathbf{x}_k^{(0)})-F^*}{T}.$$

□

E. Supplementary experiment details

In this section, we provide details on the experiments described in Section 5.

E.1. Supplementary details for Section 5.1

We elaborate on the generation of the synthetic data for the GPCA experiment in Section 5.1.

- First, we uniformly generate k pairs of orthonormal vectors $\{\boldsymbol{\epsilon}_{1,j},\boldsymbol{\epsilon}_{2,j}\}$ for $j=1,2,\ldots,k$. Each pair is generated uniformly at random, with $\boldsymbol{\epsilon}_{1,j}$ and $\boldsymbol{\epsilon}_{2,j}$ forming the basis of the j-th subspace.

- For each data point $i \in [N]$, we independently generate two Gaussian samples $\xi_{1,i}, \xi_{2,i}$. Next, we sample an index $j_i \in [k]$ uniformly at random. We then let $\mathbf{x}_i = \xi_{1,i}\boldsymbol{\epsilon}_{1,j_i} + \xi_{2,i}\boldsymbol{\epsilon}_{2,j_i}$.

We provide in Algorithm 5 a detailed pseudo-code of Lloyd's algorithm for solving the GPCA problem in the sum-of-minimum formulation (5), which consists of two steps in each iteration, say updating the clusters via (15) and precisely compute the minimizer of each group objective function

$$\min_{\mathbf{A}_j^\top \mathbf{A}_j = I_r} \frac{1}{|C_j^{(t)}|} \sum_{i \in C_j^{(t)}} \|\mathbf{y}_i^\top \mathbf{A}_j\|^2 = \frac{1}{|C_j^{(t)}|} \sum_{i \in C_j^{(t)}} \mathrm{tr}\left(\mathbf{A}_j^\top \boldsymbol{y}_i \boldsymbol{y}^\top \mathbf{A}_j\right) = \mathrm{tr}\left(\mathbf{A}_j^\top \left(\frac{1}{|C_j^{(t)}|} \sum_{i \in C_j^{(t)}} \boldsymbol{y}_i \boldsymbol{y}^\top\right) \mathbf{A}_j\right).$$

Algorithm 5 Lloyd's Algorithm for generalized principal component analysis

Initialize $\mathbf{A}_1^{(0)}, \mathbf{A}_2^{(0)}, \ldots, \mathbf{A}_k^{(0)}$. Set $F^{(-1)} = +\infty$.
for $t = 0, 1, 2, \ldots,$ max iterations **do**
  Compute $F^{(t)} = F(\mathbf{A}_1^{(t)}, \mathbf{A}_2^{(t)}, \ldots, \mathbf{A}_k^{(t)})$.
  if $F^{(t)} = F^{(t-1)}$ **then**
    Break.
  end if
  Compute the partition $\{C_j^{(t)}\}_{j=1}^k$ via (15).
  for $j = 1, 2, \ldots, k$ **do**
    if $C_j^{(t)} \neq \emptyset$ **then**
      Compute the matrix

$$\frac{1}{|C_j^{(t)}|} \sum_{i \in C_j^{(t)}} \boldsymbol{y}_i \boldsymbol{y}_i^\top$$

      and its r orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ corresponding to the smallest r eigenvalues.
      Set

$$\mathbf{A}_j^{(t+1)} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_r \end{bmatrix}.$$

    else
      $\mathbf{x}_j^{(t+1)} = \mathbf{x}_j^{(t)}$.
    end if
  end for
end for

We implement the BCD algorithm (Peng & Vidal, 2023) for the following optimization problem:

$$\min_{\mathbf{A}_j^\top \mathbf{A}_j = I_r} \frac{1}{N} \sum_{i=1}^N \prod_{j \in [k]} \|\mathbf{y}_i^\top \mathbf{A}_j\|^2. \tag{39}$$

For any $j \in [k]$, when $\mathbf{A}_l$ is fixed for all $l \in [k] \backslash \{j\}$, the problem in (39) is equivalent to:

$$\min_{\mathbf{A}_j^\top \mathbf{A}_j = I_r} \frac{1}{N} \sum_{i=1}^N w_{ij} \|\mathbf{y}_i^\top \mathbf{A}_j\|^2 = \frac{1}{N} \sum_{i=1}^N w_{ij} \mathrm{tr}\left(\mathbf{A}_j^\top \mathbf{y}_i \mathbf{y}_i^\top \mathbf{A}_j\right) = \mathrm{tr}\left(\mathbf{A}_j^\top \left(\frac{1}{N} \sum_{i=1}^N w_{ij} \mathbf{y}_i \mathbf{y}_i^\top\right) \mathbf{A}_j\right),$$

where the weights w_{ij} are given by:

$$w_{ij} = \prod_{l \neq j} \|\mathbf{y}_i^\top \mathbf{A}_l\|^2.$$

The detailed pseudo-code can be found in Algorithm 6.

Algorithm 6 Block coordinate descent for generalized principal component analysis (Peng & Vidal, 2023)

Initialize $\mathbf{A}_1^{(0)}, \mathbf{A}_2^{(0)}, \dots, \mathbf{A}_k^{(0)}$.
for $t = 0, 1, 2, \dots,$ max iterations **do**
  for $j = 1, 2, \dots, k$ **do**
    Compute the weights:

$$w_{ij}^{(t)} = \prod_{l<j} \|\mathbf{y}_i^\top \mathbf{A}_l^{(t+1)}\|^2 \cdot \prod_{l>j} \|\mathbf{y}_i^\top \mathbf{A}_l^{(t)}\|^2.$$

    Compute the matrix:

$$\frac{1}{N} \sum_{i=1}^{N} w_{ij}^{(t)} \mathbf{y}_i \mathbf{y}_i^\top$$

    and its r orthonormal eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ corresponding to the smallest r eigenvalues.
    Set

$$\mathbf{A}_j^{(t+1)} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_r \end{bmatrix}.$$

  end for
end for

E.2. Supplementary details for Section 5.2

Mixed linear regression Here, we provide the detailed pseudo-code for Lloyd's algorithm used to solve ℓ_2-regularized mixed linear regression problem in Section 5. Each iteration of the algorithm consists of two steps: reclassification and cluster parameter update. We alternatively reclassify indices i to $C_j^{(t)}$ using (15) and update the cluster parameter $\mathbf{x}_j^{(t)}$ for nonempty clusters $C_j^{(t)}$ using:

$$\mathbf{x}_j^{(t+1)} = \left(\sum_{i \in C_j^{(t)}} \mathbf{a}_i \mathbf{a}_i^\top + \lambda |C_j^{(t)}| \mathbf{I} \right)^{-1} \sum_{i \in C_j^{(t)}} b_i \mathbf{a}_i, \tag{40}$$

so that $\mathbf{x}_j^{(t+1)}$ is exactly the minimizer of the group objective function. The algorithm continues until $F^{(t)}$ stops decreasing after $\mathbf{x}^{(t)}$'s update or a max iteration number is reached. The pseudo-code is shown in Algorithm 7.

Algorithm 7 Lloyd's Algorithm for mixed linear regression

Initialize $\mathbf{x}_1^{(0)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_k^{(0)}$. Set $F^{(-1)} = +\infty$.
for $t = 0, 1, 2, \dots,$ max iterations **do**
  Compute $F^{(t)} = F(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \dots, \mathbf{x}_k^{(t)})$.
  if $F^{(t)} = F^{(t-1)}$ **then**
    Break.
  end if
  Compute the partition $\{C_j^{(t)}\}_{j=1}^k$ via (15).
  for $j = 1, 2, \dots, k$ **do**
    if $C_j^{(t)} \neq \emptyset$ **then**
      Compute $\mathbf{x}_j^{(t+1)}$ using (40).
    else
      $\mathbf{x}_j^{(t+1)} = \mathbf{x}_j^{(t)}$.
    end if
  end for
end for

The dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ for the ℓ_2-regularized mixed linear regression is synthetically generated in the following way:

- Fix the dimension d and the number of function clusters k, and sample $\mathbf{x}_1^+, \mathbf{x}_2^+, \dots, \mathbf{x}_k^+ \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}_d)$ as the linear

coefficients of k ground truth regression models.

- For $i = 1, 2, \ldots, N$, we independently generate data $\mathbf{a}_i \sim \mathcal{N}(0, \mathbf{I}_d)$, class index $c_i \sim \text{Uniform}([k])$, noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, and compute $b_i = \mathbf{a}_i^\top \mathbf{x}_{c_i}^+ + \epsilon_i$.

In the experiment, the noise level is set to $\sigma = 0.01$ and the regularization factor is set to $\lambda = 0.01$.

Mixed non-linear regression The ground truth θ_j^+'s are sampled from a standard Gaussian. The dataset $\{(\mathbf{a}_i, b_i)\}_{i=1}^N$ is generated in the same way as in the mixed linear regression experiment. We set the variance of the Gaussian noise on the dataset to $\sigma^2 = 0.01^2$ and use a regularization factor $\lambda = 0.01$.