GIST: Greedy Independent Set Thresholding for Max-Min Diversification with Submodular Utility

Matthew Fahrbach[*]
Google
fahrbach@google.com

Srikumar Ramalingam[*]
Google
rsrikumar@google.com

Morteza Zadimoghaddam[*]
Google
zadim@google.com

Sara Ahmadian
Google
sahmadian@google.com

Gui Citovsky
Google
gcitovsky@google.com

Giulia DeSalvo
Google
giuliad@google.com

Abstract

This work studies a novel subset selection problem called *max-min diversification with monotone submodular utility* (MDMS), which has a wide range of applications in machine learning, e.g., data sampling and feature selection. Given a set of points in a metric space, the goal of MDMS is to maximize $f(S) = g(S) + \lambda \cdot \texttt{div}(S)$ subject to a cardinality constraint $|S| \leq k$, where $g(S)$ is a monotone submodular function and $\texttt{div}(S) = \min_{u,v \in S: u \neq v} \text{dist}(u, v)$ is the *max-min diversity* objective. We propose the GIST algorithm, which gives a $1/2$-approximation guarantee for MDMS by approximating a series of maximum independent set problems with a bicriteria greedy algorithm. We also prove that it is NP-hard to approximate within a factor of 0.5584. Finally, we show in our empirical study that GIST outperforms state-of-the-art benchmarks for a single-shot data sampling task on ImageNet.

1 Introduction

Subset selection is a ubiquitous and challenging problem at the intersection of machine learning and combinatorial optimization, finding applications in areas such as feature selection, recommendation systems, and data summarization, including the critical task of designing pretraining sets for large language models [59, 3]. Data sampling in particular is an increasingly important problem given the unprecedented and continuous streams of data collection. For example, one self-driving vehicle generates ~80 terabytes of data daily from LiDAR and imaging devices [34], and academic datasets have scaled dramatically from 1.2M images in ImageNet [54] to 5B in Laion [56].

Subset selection often involves balancing competing objectives: we rely on the *utility* (or weight) of individual items to prioritize them, while simultaneously trying to avoid selecting duplicate or near-duplicate items to ensure *diversity*. When selecting a small subset, this process should guarantee that the chosen set is a good representation of the original dataset—often called *coverage*. The intricate trade-offs between utility, diversity, and coverage are often expressed through an objective function. It is often a significant challenge to design efficient algorithms with strong approximation guarantees for constrained subset selection problems, even when leveraging tools and techniques from combinatorial optimization such as submodular maximization, k-center clustering, and convex hull approximations.

This paper studies a novel subset selection problem called *max-min diversification with monotone submodular utility* (MDMS). Given n points V in a metric space, a nonnegative monotone submodular

[*]Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

function $g : 2^V \to \mathbb{R}_{\geq 0}$, diversity strength $\lambda \geq 0$, and cardinality constraint k, our goal is to solve

$$S^* = \arg\max_{S \subseteq V} \; g(S) + \lambda \cdot \texttt{div}(S) \quad \text{subject to} \;\; |S| \leq k, \tag{1}$$

where $\texttt{div}(S) = \min_{u,v \in S: u \neq v} \text{dist}(u, v)$ is the max-min diversification term [2, 40, 38]. Maximizing the submodular term $g(S)$ aims to select points that are the most valuable or informative. A purely utility-driven algorithm, however, can lead to redundancies where very similar points are chosen. To counteract this, we add the $\lambda \cdot \texttt{div}(S)$ term to encourage selecting points that are well spread out in the metric space, which effectively penalizes subsets with closely clustered elements. Note that λ is a knob that controls the trade-off between these two terms, acting as a regularization strength for subset selection problems.

The most relevant prior work is by Borodin et al. [13]. They combine a monotone submodular utility and the *max-sum diversity* term $\sum_{u,v \in S} \text{dist}(u, v)$, in contrast to our max-min $\texttt{div}(S)$ in (1). They give a greedy $1/2$-approximation algorithm and extend their results to general matroid constraints via local search. An earlier contribution by Gollapudi and Sharma [26] combined a *linear utility* (instead of a monotone submodular function) and the max-sum diversity term, subject to the strict constraint $|S| = k$, to design a ranking system for search engines. These approaches focus on maximizing the sum of pairwise distances, but our max-min formulation ensures that the selected points have high utility and are maximally separated, leading to less redundant and more representative subsets.

1.1 Our contributions

We summarize the main contributions of this work below:

- In Section 2, we formalize the MDMS problem and present a simple 0.387-approximation algorithm as a warm-up to show how the competing terms in the objective function interact.
- In Section 3, we present the GIST algorithm, which achieves a $1/2$-approximation guarantee for MDMS by approximating a series of maximum independent set problems with a bicriteria greedy algorithm and returning the best solution. In the special case of *linear utility functions* $g(S) = \sum_{v \in S} w(v)$, we prove that GIST offers a stronger $2/3$-approximation guarantee.
- In Section 4, we prove that it is NP-hard to approximate MDMS to within a factor of 0.5584 via a careful reduction from the maximum coverage problem. For linear utilities, we *match the guarantees* of GIST and prove a tight $(2/3 + \varepsilon)$-hardness of approximation, for any $\varepsilon > 0$, assuming P $\neq$ NP. In the more restricted case of a linear utility function and the Euclidean metric, we prove APX-completeness.
- In Section 5, our experiments show that GIST outperforms baselines for MDMS on synthetic data (in particular the classic greedy algorithm). Then we show that GIST can be used to build better single-shot subsets of training data for an image classification benchmark on ImageNet compared to margin sampling and k-center algorithms, demonstrating the benefit of optimizing for a blend of utility and diversity.

1.2 Related work

Submodular maximization. Submodular maximization subject to a cardinality constraint k is an NP-hard problem, i.e., $\max_{S \subseteq V: |S| \leq k} g(S)$. For monotone submodular functions, Nemhauser et al. [47] proved that the greedy algorithm achieves a $(1 - 1/e)$-approximation, which is asymptotically optimal unless P = NP [23]. The non-monotone case is substantially less understood, but Buchbinder and Feldman [14] recently gave a 0.401-approximation algorithm and Qi [48] improved the hardness of approximation to 0.478. Over the last decade, there has also been significant research on distributed algorithms for submodular maximization in the MapReduce [44, 43, 9, 10, 39, 33] and low-adaptivity models [7, 19, 22, 8, 5, 16, 17].

Diversity maximization. The related *max-min diversification* problem $\max_{S \subseteq V: |S| = k} \texttt{div}(S)$ has a rich history in operations research due to its connection to facility location, and is also called the *p-dispersion* problem. It uses the *strict constraint* $|S| = k$; otherwise, if $|S| \leq k$ we can maximize $\texttt{div}(S)$ with two diametrical points. Tamir [58] gave a simple greedy $1/2$-approximation algorithm for

max-min diversification, and Ravi et al. [51] proved $(1/2+\varepsilon)$-inapproximability, for any $\varepsilon > 0$, unless P = NP. Indyk et al. [30] gave a distributed $1/3$-approximation algorithm via composable coresets for diversity and coverage maximization. Borassi et al. [12] designed a $1/5$-approximation algorithm in the sliding window model. For asymmetric distances, Kumpulainen et al. [38] recently designed a $1/(6k)$-approximation algorithm. There are also many related works on mixed-integer programming formulations, heuristics, and applications to drug discovery [20, 52, 55, 60, 37, 40, 61, 42].

Bhaskara et al. [11] studied the *sum-min diversity* problem $\max_{S \subseteq V : |S| \leq k} \sum_{u \in S} \text{dist}(u, S \setminus \{u\})$, providing the first constant-factor approximation algorithm with a $1/8$ guarantee. Their algorithm is based on a novel linear-programming relaxation and generalizes to matroid constraints. They also proved the first inapproximability result of $1/2$ under the planted clique assumption.

Diversity maximization has also recently been studied with *fairness constraints*. Given a partition of the points into m groups, they ensure $k_i \in [\ell_i, u_i]$ points are selected from each group $i \in [m]$, subject to $|S| = k = \sum_{i=1}^{m} k_i$. This line of work can be categorized as studying fair max-min [46, 2, 61], sum-min [11, 41], and max-sum [1, 41] diversity objectives.

Data sampling. In the realm of dataset curation and active learning, there have been several lines of work that study the combination of utility and diversity terms. Greedy coreset methods based on k-center have been successfully used for data selection with and without utility terms [50, 57]. Ash et al. [6] use k-means++ seeding over the gradient space of a model to balance uncertainty and diversity. Wei et al. [62] introduce several submodular objectives, e.g., facility location, and use them to diversify a set of uncertain examples in each active learning iteration. Citovsky et al. [18] cluster examples represented by embeddings extracted from the penultimate layer of a partially trained DNN, and use these clusters to diversify uncertain examples in each iteration. Our work differs from these and several others (e.g., Kirsch et al. [35], Zhdanov [63]) in that we directly incorporate the utility and diversity terms into the objective function.

2 Preliminaries

Submodular function. For any $g : 2^V \to \mathbb{R}_{\geq 0}$ and $S, T \subseteq V$, let $g(S \mid T) = g(S \cup T) - g(T)$ be the *marginal gain* of g at S with respect to T. A function g is submodular if for every $S \subseteq T \subseteq V$ and $v \in V \setminus T$, we have $g(v \mid S) \geq g(v \mid T)$, where we overload the marginal gain notation for singletons. A submodular function g is *monotone* if for every $S \subseteq T \subseteq V$, $g(S) \leq g(T)$.

Max-min diversity. For a set of n points V in a metric space, define the *max-min diversity* function as

$$\texttt{div}(S) = \begin{cases} \min_{u,v \in S : u \neq v} \text{dist}(u, v) & \text{if } |S| \geq 2, \\ \max_{u,v \in V} \text{dist}(u, v) & \text{if } |S| \leq 1. \end{cases}$$

We take $\texttt{div}(S)$ to be the diameter of V if $|S| \leq 1$ so that it is monotone decreasing. We extend the distance function to take subsets $S \subseteq V$ as input in the standard way, i.e., $\text{dist}(u, S) = \min_{v \in S} \text{dist}(u, v)$, and define $\text{dist}(u, \varnothing) = \infty$.

MDMS problem statement. For any nonnegative monotone submodular function $g : 2^V \to \mathbb{R}_{\geq 0}$ and $\lambda \geq 0$, let

$$f(S) = g(S) + \lambda \cdot \texttt{div}(S). \tag{2}$$

The *max-min diversification with monotone submodular utility* (MDMS) problem is to maximize $f(S)$ subject to a cardinality constraint k:

$$S^* = \operatorname*{arg\,max}_{S \subseteq V : |S| \leq k} f(S).$$

Let $\text{OPT} = f(S^*)$ denote the optimal objective value.

Remark 2.1. The objective function $f(S)$ is *not submodular* (see Appendix A for a counterexample).

Intersection graph. Let $G_d(V)$ be the *intersection graph* of V for distance threshold d, i.e., with nodes V and edges $E = \{(u, v) \in V^2 : u \neq v, \text{dist}(u, v) < d\}$. For any independent set S of $G_d(V)$ and pair of distinct nodes $u, v \in S$, we have $\text{dist}(u, v) \geq d$.

2.1 Warm-up: Simple 0.387-approximation algorithm

The objective function $f(S)$ is composed of two terms, each of which is easy to (approximately) optimize in isolation. For the submodular term $g(S)$, run the greedy algorithm to get S_1 satisfying

$$\begin{aligned} g(S_1) &\geq (1 - 1/e) \cdot \max_{|S| \leq k} g(S) \\ &\geq (1 - 1/e) \cdot g(S^*). \end{aligned}$$

For the $\texttt{div}(S)$ term, let S_2 be a pair of diametrical points in V. Then, return the better of the two solutions. This gives a 0.387-approximation guarantee because, for any $0 \leq p \leq 1$, we have:

$$\begin{aligned} \text{ALG}_{\text{simple}} &= \max\{f(S_1), f(S_2)\} \\ &\geq p \cdot g(S_1) + (1-p) \cdot \lambda \cdot \texttt{div}(S_2) \\ &\geq p \cdot (1 - 1/e) \cdot g(S^*) + (1-p) \cdot \lambda \cdot \texttt{div}(S^*) \\ &= \frac{e-1}{2e-1} \cdot \text{OPT}, \end{aligned}$$

where we set $p = e/(2e-1)$ at the end by solving $p \cdot (1 - 1/e) = 1 - p$. One of our main goals is to improve over this baseline and prove complementary hardness of approximation results.

One of the most common algorithms for subset selection problems is the greedy algorithm that iteratively adds the item with maximum marginal value with respect to the objective function f. We show in Appendix B that the greedy algorithm does not provide a constant-factor approximation guarantee.

3 Algorithm

In this section, we present the GIST algorithm for the MDMS problem and prove that it achieves an approximation ratio of $1/2 - \varepsilon$. At a high level, GIST works by sweeping over multiple distance thresholds d and calling a GreedyIndependentSet subroutine on the intersection graph $G_d(V)$ to find a high-valued maximal independent set. This is in contrast to the warm-up 0.387-approximation algorithm, which only considers the two extreme thresholds $d \in \{0, d_{\max}\}$.

GreedyIndependentSet builds the set S (starting from the empty set) by iteratively adding $v \in V \setminus S$ with the highest marginal gain with respect to the submodular utility g while satisfying $\text{dist}(v, S) \geq d$. This subroutine runs until either $|S| = k$ or S is a maximal independent set of $G_d(V)$.

GIST computes multiple solutions and returns the one with maximum value $f(S)$. It first runs the classic greedy algorithm for monotone submodular functions to get an initial solution, which is equivalent to calling GreedyIndependentSet with distance threshold $d = 0$. Then, it considers the set of distance thresholds $D \leftarrow \{(1+\varepsilon)^i \cdot \varepsilon d_{\max}/2 : (1+\varepsilon)^i \leq 2/\varepsilon \text{ and } i \in \mathbb{Z}_{\geq 0}\}$. The set D contains a threshold close to the target $d^*/2$ where $d^* = \texttt{div}(S^*)$.[2] For each $d \in D$, GIST calls $\texttt{GreedyIndependentSet}(V, g, d, k)$ to find a set T of size at most k. If T has a larger objective value than the best solution so far, it updates $S \leftarrow T$. After iterating over all thresholds in D, it returns the highest-value solution among all candidates.

Theorem 3.1. *For any $\varepsilon > 0$,* GIST *outputs a set $S \subseteq V$ with $|S| \leq k$ and $f(S) \geq (1/2 - \varepsilon) \cdot \text{OPT}$ using $O(nk \log_{1+\varepsilon}(1/\varepsilon))$ submodular value oracle queries.*

The main building block in our design and analysis of GIST is the following lemma, which is inspired by the greedy 2-approximation algorithm for metric k-center and adapted for monotone submodular utility functions.

Lemma 3.2. *Let S_d^* be a maximum-value set of size at most k with diversity at least d. In other words,*

$$S_d^* = \operatorname*{arg\,max}_{S : |S| \leq k, \texttt{div}(S) \geq d} g(S).$$

Let T be the output of $\texttt{GreedyIndependentSet}(V, g, d', k)$. *If $d' < d/2$, then $g(T) \geq g(S_d^*)/2$.*

[2] If this is not true, we are in the special case where $d^* \leq \varepsilon d_{\max}$, which we analyze separately.

Algorithm 1 Max-min diversification with submodular utility via greedy weighted independent sets.

function GIST(points V, monotone submodular function $g : 2^V \to \mathbb{R}_{\geq 0}$, budget k, error ε)
  Initialize $S \leftarrow$ GreedyIndependentSet$(V, g, 0, k)$ ▷ classic greedy algorithm
  Let $d_{\max} = \max_{u,v \in V} \text{dist}(u, v)$ be the diameter of V
  Let $T \leftarrow \{u, v\}$ be two points such that $\text{dist}(u, v) = d_{\max}$
  if $f(T) > f(S)$ and $k \geq 2$ **then**
    Update $S \leftarrow T$
  Let $D \leftarrow \{(1+\varepsilon)^i \cdot \varepsilon d_{\max}/2 : (1+\varepsilon)^i \leq 2/\varepsilon \text{ and } i \in \mathbb{Z}_{\geq 0}\}$ ▷ distance thresholds
  for threshold $d \in D$ **do**
    Set $T \leftarrow$ GreedyIndependentSet(V, g, d, k)
    if $f(T) \geq f(S)$ **then**
      Update $S \leftarrow T$
  return S

function GreedyIndependentSet(points V, monotone submodular function $g : 2^V \to \mathbb{R}_{\geq 0}$, distance d, budget k)
  Initialize $S \leftarrow \varnothing$
  for $i = 1$ to k **do**
    Let $C \leftarrow \{v \in V \setminus S : \text{dist}(v, S) \geq d\}$
    if $C = \varnothing$ **then**
      return S ▷ S is a maximal independent set of $G_d(V)$
    Find $t \leftarrow \arg\max_{v \in C} g(v \mid S)$
    Update $S \leftarrow S \cup \{t\}$
  return S

Proof. Let $k' = |T|$ and $t_1, t_2, \ldots, t_{k'}$ be the points in T in selection order. Let $B_i = \{v \in V : \text{dist}(t_i, v) < d'\}$ be the points in V in the radius-d' open ball around t_i. Since the distance between any pair of points in B_i is at most $2d' < d$, each set B_i contains at most one point in S_d^*.

First we construct an injective map $h : S_d^* \to T$. We say that B_i *covers* a point $v \in V$ if $v \in B_i$. For each covered $s \in S_d^*$, we map it to t_i, where i is the minimum index for which B_i covers s. If there is an $s \in S_d^*$ not covered by any set B_i, then GreedyIndependentSet must have selected k points, i.e., $|T| = k$. We map the uncovered points in S_d^* to arbitrary points in T while preserving the injective property.

Since g is a monotone submodular function, we have

$$g(S_d^*) - g(T) \leq \sum_{s \in S_d^*} g(s \mid T).$$

We use h to account for $g(s \mid T)$ in terms of the values we gained in set T. For any $s \in S_d^*$, let T_s be the set of points added to set T in algorithm GreedyIndependentSet right before the addition of $h(s)$. Since GreedyIndependentSet iteratively adds points and never deletes points from T, we know $T_s \subseteq T$. By submodularity, $g(s \mid T) \leq g(s \mid T_s)$. By the greedy nature of the algorithm, we also know that

$$g(h(s) \mid T_s) \geq g(s \mid T_s).$$

We note that the sum of the former terms, $g(h(s) \mid T_s)$, is at most $g(T)$ since h is injective and g is nondecreasing. Thus, we conclude that

$$g(S_d^*) - g(T) \leq \sum_{s \in S_d^*} g(s \mid T) \leq \sum_{s \in S_d^*} g(h(s) \mid T_s) \leq g(T),$$

which completes the proof. □

Now that we have with this bicriteria approximation, we can analyze the approximation ratio of GIST.

Proof of Theorem 3.1. We first prove the oracle complexity of GIST. There are $1 + \log_{1+\varepsilon}(2/\varepsilon) = O(\log_{1+\varepsilon}(1/\varepsilon))$ thresholds in D. For each threshold, we call GreedyIndependentSet once. In each call for k iterations, we find the maximum marginal-value point by scanning (in the worst case) all points. This requires at most nk oracle calls yielding the overall oracle complexity of the algorithm.

Let d^* be the minimum pairwise distance between points in S^*. The GIST algorithm iterates over a set of thresholds D. The definition of D implies that at least one threshold d is in the interval $[d^*/(2(1+\varepsilon)), d^*/2)$, unless $d^* \leq \varepsilon d_{\max}$. We deal with this special case later and focus on the case when such a d exists.

Let T be the output of GreedyIndependentSet for threshold value d. Since $d < d^*/2$, Lemma 3.2 implies that

$$g(T) \geq \frac{1}{2} \cdot g(S^*_{d^*}) \geq \frac{1}{2} \cdot g(S^*).$$

We also know from our choice of d that

$$\texttt{div}(T) \geq \frac{1}{2(1+\varepsilon)} \cdot \texttt{div}(S^*).$$

Combining these inequalities gives us

$$f(T) \geq \frac{1}{2(1+\varepsilon)} \cdot \text{OPT} > \left(\frac{1}{2} - \varepsilon\right) \cdot \text{OPT}.$$

It remains to prove the claim for the case when $d^* \leq \varepsilon d_{\max}$. GIST considers two initial feasible sets and picks the better of the two as the initial value for T. The first set is the classic greedy solution [47] for the monotone submodular function $g(S)$, and ignores the diversity term. It follows that

$$f(T) \geq \left(1 - \frac{1}{e}\right) \cdot g(S^*) \geq \left(1 - \frac{1}{e}\right) \cdot (\text{OPT} - \lambda \cdot \varepsilon d_{\max}). \tag{3}$$

The second set contains two points with pairwise distance $d_{\max}$, and ignores the submodular term. This yields the lower bound

$$f(T) \geq \lambda \cdot d_{\max} \implies -\lambda \cdot d_{\max} \geq -f(T). \tag{4}$$

Combining (3) and (4), we get a final bound of $f(T) \geq (1 - \frac{1}{e}) \cdot \frac{\text{OPT}}{1+\varepsilon}$, which completes the proof. □

3.1 Linear utility

If $g(S) = \sum_{v \in S} w(v)$ is a linear utility with nonnegative weights $w : V \to \mathbb{R}_{\geq 0}$, then Theorem 3.1 gives us a $(1/2 - \varepsilon)$-approximation since this is a special case of submodularity. However, GIST offers a *stronger approximation ratio* under this assumption.

Theorem 3.3. *Let $g(S) = \sum_{v \in S} w(v)$ be a linear function. For any $\varepsilon > 0$,* GIST *returns $S \subseteq V$ with $|S| \leq k$ and $f(S) \geq (2/3 - \varepsilon) \cdot \text{OPT}$.*

We defer the proof to Appendix C.1, but explain the main differences. For linear functions, Lemma 3.2 can be strengthened to show that GreedyIndependentSet outputs a set T such that $g(T) \geq g(S^*_d)$, for any $d' < d/2$. Then, for some $d \in D$, we get ALG $\geq \max\{g(S^*) + \lambda \cdot d^*/(2(1+\varepsilon)), \lambda \cdot d^*\}$, and the optimal convex combination of these two lower bounds gives the approximation ratio.

4 Hardness of approximation

We begin by summarizing our hardness results for MDMS. Assuming P $\neq$ NP, we prove that:

Submodular utility. There is no polynomial-time 0.5584-approximation algorithm if g is a nonnegative, monotone submodular function.

Linear utility.

- There is no polynomial-time $(2/3 + \varepsilon)$-approximation algorithm if g is linear, for any $\varepsilon > 0$, for general distance metrics.
- APX-completeness for linear utility functions even in the Euclidean metric, i.e., there is no *polynomial-time approximation scheme* (PTAS) for this problem.

4.1 General metric spaces

We build on the set cover hardness instances originally designed by Feige et al. [23, 25, 24], and then further engineered by Kapralov et al. [32]. We present the instances in [32, Section 2] as follows:

Hardness of max k-cover. For any constants $c, \varepsilon > 0$ and a given collection/family of sets $\mathcal{F}$ partitioned into groups $\mathcal{F}_1, \mathcal{F}_2, \cdots, \mathcal{F}_k$, it is NP-hard to distinguish between the following two cases:

- YES case: there exists k disjoint sets S_i, one from each $\mathcal{F}_i$, whose union covers the entire universe.
- NO case: for any $\ell \leq c \cdot k$ sets, their union covers at most a $(1 - (1 - 1/k)^{\ell} + \varepsilon)$-fraction of the universe.

The construction above can be done such that each set $S_i \in \mathcal{F} = \bigcup_{r=1}^{k} \mathcal{F}_r$ has the same size. We note that the parameter k should be larger than any constant we set since there is always an exhaustive search algorithm with running time $|\mathcal{F}|^k$ to distinguish between the YES and NO cases. Therefore, we can assume, e.g., that $k \geq 1/\varepsilon$.

We use this max k-cover instance to prove hardness of approximation for the MDMS problem.

Theorem 4.1. *It is* NP*-hard to approximate* MDMS *within a factor of* $\frac{2(1-1/e)}{2(1-1/e)+1} + \delta < 0.55836 + \delta$*, for any* $\delta > 0$*.*

Proof sketch. We reduce an instance of the max k-cover problem to MDMS to demonstrate how the inapproximability result translates. For any pair of sets in the collection $\mathcal{F}$, their distance is defined to be 2 if and only if they are disjoint; otherwise, it is 1. Consequently, a selected sub-collection of $\mathcal{F}$ achieves the higher diversity score of 2 if and only if all chosen sets are mutually disjoint.

Given that all sets in $\mathcal{F}$ are of uniform size and considering the upper bound in the NO case on the union coverage for any collection of sets, a polynomial-time algorithm can only find $O(\varepsilon k)$ pairwise disjoint sets. Thus, if an algorithm aims for a diversity score of 2 (by selecting only disjoint sets), it forgoes most of the potential value from the submodular coverage function in the MDMS formulation. On the other hand, if the algorithm settles for the lower diversity score of 1, it still cannot cover more than an approximate fraction $1 - (1 - 1/k)^k + \varepsilon \approx 1 - 1/e + \varepsilon$ of the universe.

Combining these two upper bounds on any polynomial-time algorithm's performance with the fact that, in the YES case, the optimal solution covers the entire universe and achieves diversity score 2, establishes the claimed hardness of approximation. □

4.2 Linear utility

For the special case of linear utility functions, we prove a tight hardness result to complement our $2/3$-approximation guarantee in Theorem 3.3.

Theorem 4.2. *For any* $\varepsilon > 0$*, there is no polynomial-time* $(2/3 + \varepsilon)$*-approximation algorithm for the* MDMS *problem if* g *is a linear function, unless* P = NP*.*

We defer the proof to Appendix D.2. Our construction builds on the work of Håstad [28] and Zuckerman [64], which shows that the max clique problem does not admit an efficient $n^{1-\theta}$-approximation algorithm, for any constant $\theta > 0$.

4.3 Euclidean metric

Our final result is for the Euclidean metric, i.e., $S \subseteq \mathbb{R}^d$ and $\text{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. We build on a result of Alimonti and Kann [4] showing that the size of a max independent set in a bounded-degree graph cannot be approximated to within a constant factor $1 - \varepsilon$, for any $\varepsilon > 0$, unless P = NP.

Lemma 4.3 (Alimonti and Kann [4, Theorem 3.2])**.** *The maximum independent set problem for graphs with degree at most* 3 *is* APX*-complete.*

Our second ingredient is an embedding function $h_G(v)$ that encodes graph adjacency in Euclidean space.

Lemma 4.4. *Let* $G = (V, E)$ *be a simple undirected graph with* $n = |V|$*,* $m = |E|$*, and max degree* Δ*. There exists an embedding* $h_G : V \to \mathbb{R}^{n+m}$ *such that if* $\{u, v\} \in E$ *then*

$$\|h_G(u) - h_G(v)\|_2 \leq 1 - \frac{1}{2(\Delta + 1)},$$

and if $\{u, v\} \notin E$ *then* $\|h_G(u) - h_G(v)\|_2 = 1$*.*

Theorem 4.5. MDMS *is* APX*-complete for the Euclidean metric if g is a linear function.*

We sketch the main idea below and defer proofs to Appendix D. Let $\mathcal{I}(G)$ be the set of independent sets of G. Using Lemma 4.4, for any set of nodes $S \subseteq V$ with $|S| \geq 2$ in a graph G with max degree $\Delta \leq 3$, we have the property:

- $S \in \mathcal{I}(G) \implies \texttt{div}(S) = 1$;
- $S \notin \mathcal{I}(G) \implies \texttt{div}(S) \leq 1 - \frac{1}{2(\Delta+1)} \leq 1 - 1/8$.

The gap is at least $1/8$ for all such graphs (i.e., a universal constant), so setting $\lambda = 1$ allows us to prove that MDMS inherits the APX-completeness of bounded-degree max independent set.

5 Experiments

5.1 Warm-up: Synthetic dataset

We first compare GIST against baseline methods on a simple MDMS task with a normalized budget-additive utility function (i.e., monotone submodular) and a set of weighted Guassian points.

Setup. Generate $n = 1000$ points $\mathbf{x}_i \in \mathbb{R}^d$, for $d = 64$, where each $\mathbf{x}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ is i.i.d. and assigned a uniform continuous weight $w_i \sim \mathcal{U}_{[0,1]}$. For $\alpha, \beta \in [0, 1]$, the objective function trades off between the average (capped) utility of the points and their min-distance diversity reward:

$$f(S) = \alpha \cdot \min\left\{\frac{1}{k}\sum_{i \in S} w_i,\ \beta\right\} + (1 - \alpha) \cdot \texttt{div}(S).$$

We consider three baseline methods: `random`, `simple`, and `greedy`. `random` selects k random points, permutes them, and returns the best prefix since the objective function $f(S)$ is non-monotone. `simple` is the 0.387-approximation algorithm in Section 2.1. `greedy` builds S one point at a time by selecting the point with index $i_t^* = \arg\max_{i \in V \setminus S_{t-1}} f(S_{t-1} \cup \{i\}) - f(S_{t-1})$ at each step $t \in [k]$, and returns the best prefix $S = \arg\max_{t \in [k]} f(S_t)$. GIST considers all possible $D \leftarrow \{\text{dist}(u, v)/2 : u, v \in V\}$ since $n = 1000$, which yields an exact $2/3$-approximation ratio.

Results. We plot the values of $f(S_{\text{ALG}})$ for the baseline methods and GIST, for each $k \in [n]$, in Figure 1. GIST dominates `simple`, `greedy`, and `random` in all cases. `greedy` performs poorly for $k \geq 100$, which is surprising since it is normally a competitive method for subset selection tasks. `simple` beats `greedy` for mid-range values of k, i.e., $k \in [250, 900]$, but it is always worse than GIST. In Appendix E.1, we sweep over α and β, and show how these hyperparameters affect the $f(S_{\text{ALG}})$ plots. Finally, we remark that the plots in Figure 1 are decreasing in k because (i) we use a *normalized* budget-additive utility function, and (ii) $\texttt{div}(S)$ is a monotone-decreasing set function.

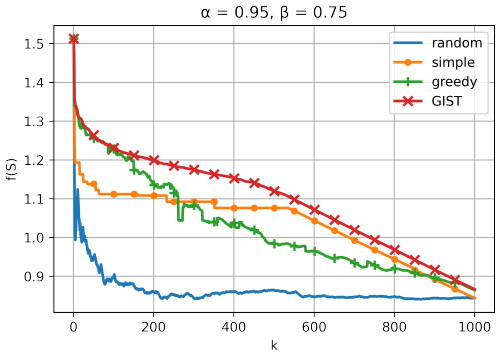

Figure 1: $f(S_{\text{ALG}})$ for baseline methods and GIST, for each cardinality constraint $k \in [n]$, on synthetic data with $n = 1000$, $\alpha = 0.95$, and $\beta = 0.75$.

5.2 Image classification

Our ImageNet data sampling experiment compares the top-1 image classification accuracy achieved by different single-shot subset selection algorithms.

Setup. We use the standard vision dataset ImageNet [54] containing ~1.3 million images and 1000 classes. We select 10% of the images uniformly at random and use them to train an initial ResNet-56 model $\boldsymbol{\theta}_0$ [29]. Then we use the model $\boldsymbol{\theta}_0$ to compute a 2048-dimensional unit-length embedding $\boldsymbol{e}_i$

Table 1: Top-1 classification accuracy (%) on ImageNet for different single-shot data downsampling algorithms. The cardinality constraint k is expressed as a percent of the ~1.3 million examples. The results are the average of three trials and the top performance for each k is shown in bold.

k (%)	random	margin	k-center	submod	GIST-margin	GIST-submod
30	66.23 $\pm$ 0.15	65.97 $\pm$ 0.33	66.71 $\pm$ 0.57	66.52 $\pm$ 0.21	66.90 $\pm$ 0.19	**67.24** $\pm$ 0.06
40	69.17 $\pm$ 0.12	69.73 $\pm$ 0.38	70.06 $\pm$ 0.06	70.71 $\pm$ 0.24	70.51 $\pm$ 0.12	**70.76** $\pm$ 0.35
50	71.05 $\pm$ 0.16	72.33 $\pm$ 0.09	73.01 $\pm$ 0.10	72.69 $\pm$ 0.15	72.69 $\pm$ 0.34	**73.15** $\pm$ 0.21
60	72.49 $\pm$ 0.28	73.43 $\pm$ 0.06	73.60 $\pm$ 0.31	74.20 $\pm$ 0.08	**74.34** $\pm$ 0.01	74.30 $\pm$ 0.27
70	73.70 $\pm$ 0.22	74.49 $\pm$ 0.22	74.24 $\pm$ 0.23	75.24 $\pm$ 0.30	**75.41** $\pm$ 0.23	75.32 $\pm$ 0.19
80	74.42 $\pm$ 0.39	75.01 $\pm$ 0.05	75.17 $\pm$ 0.11	75.91 $\pm$ 0.07	**75.96** $\pm$ 0.28	75.45 $\pm$ 0.16
90	75.16 $\pm$ 0.13	75.11 $\pm$ 0.13	75.00 $\pm$ 0.38	75.84 $\pm$ 0.10	**76.12** $\pm$ 0.33	76.03 $\pm$ 0.02

and uncertainty score for each example $\boldsymbol{x}_i$. The *margin-based* uncertainty score of $\mathbf{x}_i$ is given by $u_i = 1 - (\Pr(y = b \mid \mathbf{x}_i;\ \boldsymbol{\theta}_0) - \Pr(y = b' \mid \mathbf{x}_i;\ \boldsymbol{\theta}_0))$, which measures the difference between the probability of the best predicted class label b and second-best label b' for an example. Finally, we use the fast maximum inner product search of Guo et al. [27] to build a Δ-nearest neighbor graph G in the embedding space using $\Delta = 100$ and cosine distance, i.e., $\text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 - \boldsymbol{e}_i \cdot \boldsymbol{e}_j$. We present all model training hyperparameters in Appendix E.

We compare GIST with several state-of-the-art benchmarks:

- random: We draw samples from the dataset uniformly at random without replacement. This is a simple and lightweight approach that promotes diversity in many settings and provides good solutions.
- margin [53]: Margin sampling selects the top-k points using the uncertainty scores u_i, i.e., based on how hard they are to classify. It is not incentivized to output a diverse set of training examples.
- k-center [57]: We run the classic greedy algorithm for k-center on G. We take the distance between non-adjacent nodes to be the max distance among all pairs of adjacent nodes in G.
- submod: We select a subset by greedily maximizing the submodular objective function

$$g(S) = \alpha_s \sum_{i \in S} u_i - \beta_s \sum_{i,j \in S} s(i,j), \tag{5}$$

  subject to the constraint $|S| \leq k$, where $s(i,j) = 1 - \text{dist}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the cosine similarity between adjacent nodes in G. Similar pairwise-diversity submodular objective functions have also been used in [45, 21, 31, 36, 49]. This is a *different diversity objective* than div(S) that keeps $g(S)$ submodular but allows it to be non-monotone. We tuned for best performance by selecting $\alpha_s = 0.9$ and $\beta_s = 0.1$.

Finally, we bootstrap the margin and submod objectives with MDMS and run GIST with $\varepsilon = 0.05$:

- GIST-margin: Let $f(S) = \alpha \cdot \sum_{i \in S} u_i + (1-\alpha) \cdot \text{div}(S)$ for $\alpha \in [0,1]$. This uses the same *linear utility function* as margin sampling. We optimize for performance and set $\alpha = 0.9$.
- GIST-submod: Let $f(S) = \alpha \cdot g(S) + (1-\alpha) \cdot \text{div}(S)$, where $g(S)$ is the same submodular function in (5) with $\alpha_s = 0.9$ and $\beta_s = 0.1$. We optimize for performance and set $\alpha = 0.95$.

Results. We run each sampling algorithm with cardinality constraint k on the full dataset to get a subset of examples that we then use to train a new ResNet-56 model. We report the average top-1 classification accuracy of these models in Table 1. GIST with margin or submodular utility is superior to all baselines. Interestingly, there is a cut-over value of k where the best algorithm switches from GIST-submod to GIST-margin. We also observe that GIST-submod and GIST-margin outperform submod and margin, respectively. This demonstrates how div(S) encourages diversity in the set of sampled points and improves downstream model quality. The running time of margin and submod is 3–4 minutes per run on average. GIST is similar to the margin or submodular algorithms for a given distance threshold d. The end-to-end running time is dominated by training ImageNet models, which takes more than a few hours even with several accelerators (e.g., GPU/TPU chips).

Conclusion

We introduce a novel subset selection problem called MDMS that combines the utility of the selected points (modeled as a monotone submodular function g) with the $\texttt{div}(S) = \min_{u,v \in S: u \neq v} \text{dist}(u, v)$ diversity objective. We design and analyze the GIST algorithm, which achieves a $1/2$-approximation guarantee by solving a series of maximal-weight independent set instances on intersection graphs with the GreedyIndependentSet bicriteria-approximation algorithm. We complement GIST with a 0.5584 hardness of approximation. It is an interesting open theory problem to close the gap between the 0.5 approximation ratio and 0.5584 inapproximability. For linear utilities, we show that GIST achieves a $2/3$-approximation and that it is NP-hard to find a $(2/3 + \varepsilon)$-approximation, for any $\varepsilon > 0$.

Our empirical study starts by comparing GIST to existing methods for MDMS on a simple synthetic task to show the shortcomings of baseline methods (in particular the greedy algorithm). Then we compare the top-1 image classification accuracy of GIST and state-of-the-art data sampling methods for ImageNet, demonstrating the benefit of optimizing the trade-off between a submodular utility and max-min diversity.

References

[1] Z. Abbassi, V. S. Mirrokni, and M. Thakur. Diversity maximization under matroid constraints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–40, 2013.

[2] R. Addanki, A. McGregor, A. Meliou, and Z. Moumoulidou. Improved approximation and scalability for fair max-min diversification. In *Proceedings of the 25th International Conference on Database Theory*, pages 7:1–7:21, 2022.

[3] A. Albalak, Y. Elazar, S. M. Xie, S. Longpre, N. Lambert, X. Wang, N. Muennighoff, B. Hou, L. Pan, H. Jeong, C. Raffel, S. Chang, T. Hashimoto, and W. Y. Wang. A survey on data selection for language models. *ArXiv*, abs/2402.16827, 2024.

[4] P. Alimonti and V. Kann. Some APX-completeness results for cubic graphs. *Theoretical Computer Science*, 237(1-2):123–134, 2000.

[5] G. Amanatidis, F. Fusco, P. Lazos, S. Leonardi, A. Marchetti-Spaccamela, and R. Reiffenhäuser. Submodular maximization subject to a knapsack constraint: Combinatorial algorithms with near-optimal adaptive complexity. In *International Conference on Machine Learning*, pages 231–242. PMLR, 2021.

[6] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[7] E. Balkanski and Y. Singer. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th annual ACM SIGACT Symposium on Theory of Computing*, pages 1138–1151, 2018.

[8] E. Balkanski, A. Rubinstein, and Y. Singer. An exponential speedup in parallel running time for submodular maximization without loss in approximation. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 283–302. SIAM, 2019.

[9] R. Barbosa, A. Ene, H. Nguyen, and J. Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *International Conference on Machine Learning*, pages 1236–1244. PMLR, 2015.

[10] R. d. P. Barbosa, A. Ene, H. L. Nguyen, and J. Ward. A new framework for distributed submodular maximization. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 645–654. Ieee, 2016.

[11] A. Bhaskara, M. Ghadiri, V. Mirrokni, and O. Svensson. Linear relaxations for finding diverse elements in metric spaces. *Advances in Neural Information Processing Systems*, 29:4098–4106, 2016.

[12] M. Borassi, A. Epasto, S. Lattanzi, S. Vassilvitskii, and M. Zadimoghaddam. Better sliding window algorithms to maximize subadditive and diversity objectives. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 254–268, 2019.

[13] A. Borodin, A. Jain, H. C. Lee, and Y. Ye. Max-sum diversification, monotone submodular functions, and dynamic updates. *ACM Transactions on Algorithms (TALG)*, 13(3):1–25, 2017.

[14] N. Buchbinder and M. Feldman. Constrained submodular maximization via new bounds for dr-submodular functions. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1820–1831, 2024.

[15] M. Böther, A. Sebastian, P. Awasthi, A. Klimovic, and S. Ramalingam. On distributed larger-than-memory subset selection with pairwise submodular functions. In *Proceedings of the 8th MLSys Conference*, 2025.

[16] Y. Chen and A. Kuhnle. Practical and parallelizable algorithms for non-monotone submodular maximization with size constraint. *Journal of Artificial Intelligence Research*, 79:599–637, 2024.

[17] Y. Chen, W. Chen, and A. Kuhnle. Breaking barriers: Combinatorial algorithms for non-monotone submodular maximization with sublinear adaptivity and $1/e$ approximation. *arXiv preprint arXiv:2502.07062*, 2025.

[18] G. Citovsky, G. DeSalvo, C. Gentile, L. Karydas, A. Rajagopalan, A. Rostamizadeh, and S. Kumar. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34:11933–11944, 2021.

[19] A. Ene and H. L. Nguyen. Submodular maximization with nearly-optimal approximation and adaptivity in nearly-linear time. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 274–282. SIAM, 2019.

[20] E. Erkut. The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48–60, 1990.

[21] M. Fahrbach, V. Mirrokni, and M. Zadimoghaddam. Non-monotone submodular maximization with nearly optimal adaptivity and query complexity. In *International Conference on Machine Learning*, pages 1833–1842. PMLR, 2019.

[22] M. Fahrbach, V. Mirrokni, and M. Zadimoghaddam. Submodular maximization with nearly optimal approximation, adaptivity and query complexity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 255–273. SIAM, 2019.

[23] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[24] U. Feige and J. Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6 (1):247–290, 2010.

[25] U. Feige, L. Lovász, and P. Tetali. Approximating min sum set cover. *Algorithmica*, 40:219–234, 2004.

[26] S. Gollapudi and A. Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th International Conference on World Wide Web*, pages 381–390, 2009.

[27] R. Guo, P. Sun, E. Lindgren, Q. Geng, D. Simcha, F. Chern, and S. Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896, 2020.

[28] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. In *Proceedings of 37th Conference on Foundations of Computer Science*, pages 627–636, 1996.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[30] P. Indyk, S. Mahabadi, M. Mahdian, and V. S. Mirrokni. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 100–108, 2014.

[31] R. Iyer, N. Khargoankar, J. Bilmes, and H. Asanani. Submodular combinatorial information measures with applications in machine learning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 2021.

[32] M. Kapralov, I. Post, and J. Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1216–1225. SIAM, 2013.

[33] E. Kazemi, S. Minaee, M. Feldman, and A. Karbasi. Regularized submodular maximization at scale. In *International Conference on Machine Learning*, pages 5356–5366. PMLR, 2021.

[34] F. Kazhamiaka, M. Zaharia, and P. Bailis. Challenges and opportunities for autonomous vehicle query systems. In *Proceedings of the Conference on Innovative Data Systems Research*, 2021.

[35] A. Kirsch, J. Van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] S. Kothawade, N. Beck, K. Killamsetty, and R. Iyer. SIMILAR: Submodular information measures based active learning in realistic scenarios. In *Advances in Neural Information Processing Systems*, 2021.

[37] J. Kudela. Social distancing as p-dispersion problem. *IEEE Access*, 8:149402–149411, 2020.

[38] I. Kumpulainen, F. Adriaens, and N. Tatti. Max-min diversification with asymmetric distances. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1440–1450, 2024.

[39] P. Liu and J. Vondrák. Submodular optimization in the MapReduce model. In *2nd Symposium on Simplicity in Algorithms*, volume 69, pages 18:1–18:10. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[40] I. Lozano-Osorio, A. Martínez-Gavara, R. Martí, and A. Duarte. Max–min dispersion with capacity and cost for a practical location problem. *Expert Systems with Applications*, 200:116899, 2022.

[41] S. Mahabadi and S. Trajanovski. Core-sets for fair and diverse data summarization. *Advances in Neural Information Processing Systems*, 2023.

[42] T. Meinl, C. Ostermann, and M. R. Berthold. Maximum-score diversity selection for early drug discovery. *Journal of Chemical Information and Modeling*, 51(2):237–247, 2011.

[43] V. Mirrokni and M. Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 153–162, 2015.

[44] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. *Advances in Neural Information Processing Systems*, 26, 2013.

[45] B. Mirzasoleiman, A. Badanidiyuru, and A. Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *International Conference on Machine Learning*, pages 1358–1367. PMLR, 2016.

[46] Z. Moumoulidou, A. McGregor, and A. Meliou. Diverse data selection under fairness constraints. In *Proceedings of the 24th International Conference on Database Theory*, pages 13:1–13:25, 2020.

[47] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.

[48] B. Qi. On maximizing sums of non-monotone submodular and linear functions. *Algorithmica*, 86(4): 1080–1134, 2024.

[49] S. Ramalingam, D. Glasner, K. Patel, R. Vemulapalli, S. Jayasumana, and S. Kumar. Balancing constraints and submodularity in data subset selection. *CoRR*, abs/2104.12835, 2021.

[50] S. Ramalingam, P. Awasthi, and S. Kumar. A weighted k-center algorithm for data subset selection, 2023. URL https://arxiv.org/abs/2312.10602.

[51] S. S. Ravi, D. J. Rosenkrantz, and G. K. Tayi. Heuristic and special case algorithms for dispersion problems. *Operations Research*, 42(2):299–310, 1994.

[52] M. G. Resende, R. Martí, M. Gallego, and A. Duarte. GRASP and path relinking for the max–min diversity problem. *Computers & Operations Research*, 37(3):498–508, 2010.

[53] D. Roth and K. Small. Margin-based active learning for structured output spaces. In *Proceedings of the 17th European Conference on Machine Learning*, 2006.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[55] F. Sayyady and Y. Fathi. An integer programming approach for solving the p-dispersion problem. *European Journal of Operational Research*, 253(1):216–225, 2016.

[56] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Proceedings of the Conference on Neural Information Processing Systems*, 2022.

[57] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[58] A. Tamir. Obnoxious facility location on graphs. *SIAM Journal on Discrete Mathematics*, 4(4):550–567, 1991.

[59] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[60] G. K. Tutunchi and Y. Fathi. Effective methods for solving the bi-criteria p-center and p-dispersion problem. *Computers & Operations Research*, 101:43–54, 2019.

[61] Y. Wang, M. Mathioudakis, J. Li, and F. Fabbri. Max-min diversification with fairness constraints: Exact and approximation algorithms. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 91–99. SIAM, 2023.

[62] K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963, 2015.

[63] F. Zhdanov. Diverse mini-batch active learning. *arXiv preprint arXiv:1901.05954*, 2019.

[64] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 681–690, 2006.

A $f(S)$ is not submodular

Consider the instance on four points $N = \{a, b, c, d\} \subseteq \mathbb{R}$ where $a = 0$, $b = 1$, and $c = d = 2$ with the (one-dimensional) Euclidean metric, i.e., all the points are collinear. Let $g(S) = 0$ for all $S \subseteq V$. For $f(S)$ to be submodular, it must hold that for every $S \subseteq T \subseteq N$ and $x \in N \setminus T$,

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T).$$

However, if $x = b$, $S = \{a, c\}$ and $T = \{a, c, d\}$, we have:

$$f(S \cup \{x\}) - f(S) = f(\{a, b, c\}) - f(\{a, c\}) = 1 - 2 = -1$$
$$f(T \cup \{x\}) - f(T) = f(\{a, b, c, d\}) - f(\{a, c, d\}) = 0 - 0 = 0,$$

so $f(S)$ is not submodular.

In this instance, $f(S)$ is not monotone. However, a similar monotone but still not submodular $f(S)$ can be defined by setting $g(S) = \sum_{v \in S} w(v)$ where $w(v) = 2$ for every $v \in N$.

B Greedy does not give a constant-factor approximation guarantee

The objective function f is highly non-monotone since the diversity term can decrease as we add items. Consequently, the standard greedy algorithm, which rejects any item with a negative marginal gain, can have an arbitrarily poor performance.

We demonstrate this with a hard instance. Let the submodular part of the objective be $g(S) = |S|$. For the diversity component, define the distance between a specific pair (u, v) to be $\text{dist}(u, v) = 2 + 2\varepsilon$, while for all other pairs $(x, y) \neq (u, v)$, we have $\text{dist}(x, y) = 1 + \varepsilon$. The greedy algorithm first selects the set $\{u, v\}$, achieving a value of $f(\{u, v\}) = g(\{u, v\}) + \text{dist}(u, v) = 4 + 2\varepsilon$. The algorithm then terminates because adding any subsequent item gives a submodular gain of 1 but causes a diversity loss of $1 + \varepsilon$, resulting in a negative marginal gain. An optimal solution of size k, however, can achieve a value of at least k from the submodular term alone. The resulting approximation ratio for greedy is at most $(4 + 2\varepsilon)/k$, which approaches 0 as k grows. Thus, greedy offers no constant-factor approximation guarantee. Furthermore, modifying greedy to accept items with negative marginal value does not solve this problem since one can construct instances where an optimal solution value is dominated by a diversity term such that selecting any set of size k reduces the diversity term to zero.

C Missing analysis for Section 3

C.1 Proof of Theorem 3.3

The following result is a tighter analysis of the bicriteria approximation of `GreedyIndependentSet` (Lemma 3.2) if $g(S)$ is a linear function. This is the key ingredient for improving the approximation ratio of `GIST` to $2/3 - \varepsilon$.

Lemma C.1. *Let $g(S) = \sum_{v \in S} w(v)$ be a linear function with nonnegative weights $w : V \to \mathbb{R}_{\geq 0}$. Let S_d^* be a max-weight independent set of the intersection graph $G_d(V)$ of size at most k. If T is the output of* `GreedyIndependentSet`(V, g, d', k)*, for $d' \leq d/2$, then $w(T) \geq w(S_d^*)$.*

Proof. Let $k' = |T|$ and $t_1, t_2, \ldots, t_{k'}$ be the points in T in the order that `GreedyIndependentSet` selected them. Let $B_i = \{v \in V : \text{dist}(t_i, v) < d'\}$ be the points in V contained in the radius-d' open ball around t_i. First, we show that each B_i contains at most one point in S_d^*. If this is not true, then some B_i contains two different points $u, v \in S_d^*$. Since $\text{dist}(\cdot, \cdot)$ is a metric, this means

$$\begin{aligned} \text{dist}(u, v) &\leq \text{dist}(u, t_i) + \text{dist}(t_i, v) \\ &< d' + d' \\ &\leq d/2 + d/2 \\ &= d, \end{aligned}$$

which contradicts the assumption that S_d^* is an independent set of $G_d(V)$. Note that it is possible to have $B_i \cap B_j \neq \varnothing$, for $i \neq j$, since these balls consider all points in V.

Now let $C_i \subseteq V$ be the set of uncovered points (by the open balls) that become covered when `GreedyIndependentSet` selects t_i. Concretely, $C_i = B_i \setminus (B_1 \cup \cdots \cup B_{i-1})$. Each C_i contains at most one point in S_d^* since $|B_i \cap S_d^*| \leq 1$. Moreover, if $s \in C_i \cap S_d^*$, then $w(t_i) \geq w(s)$ because the points are sorted in non-increasing order and selected if uncovered.

Let $A = C_1 \cup \cdots \cup C_{k'}$ be the set of points covered by the algorithm. For each point $s \in S_d^* \cap A$, there is exactly one covering set C_i corresponding to s. It follows that

$$\sum_{s \in S_d^* \cap A} w(s) \leq \sum_{i \in [k']: S_d^* \cap C_i \neq \varnothing} w(t_i). \tag{6}$$

It remains to account for the points in $S_d^* \setminus A$. If we have any such points, then $|T| = k$ since the points in $S_d^* \setminus A$ are uncovered at the end of the algorithm. Further, for any $t_i \in T$ and $s \in S_d^* \setminus A$, we have $w(t_i) \geq w(s)$ since t_i was selected and the points are sorted by non-increasing weight. Therefore, we can assign each $s \in S_d^* \setminus A$ to a unique C_i such that $C_i \cap S_d^* = \varnothing$. It follows that

$$\sum_{s \in S_d^* \setminus A} w(s) \leq \sum_{i \in [k']: S_d^* \cap C_i = \varnothing} w(t_i). \tag{7}$$

Adding the two sums together in (6) and (7) completes the proof. □

Theorem 3.3. *Let $g(S) = \sum_{v \in S} w(v)$ be a linear function. For any $\varepsilon > 0$,* `GIST` *returns $S \subseteq V$ with $|S| \leq k$ and $f(S) \geq (2/3 - \varepsilon) \cdot \mathrm{OPT}$.*

Proof of Theorem 3.3. Let d^* be the minimum distance between two distinct points in S^*. There are two cases: $d^* \leq \varepsilon d_{\max}$ and $d^* > \varepsilon d_{\max}$. In the first case, outputting the k heaviest points (Line 2 of `GIST`) yields a $(1-\varepsilon)$-approximation. To see this, first observe that

$$\mathrm{OPT} \geq \lambda \cdot d_{\max} \geq \lambda \cdot \frac{d^*}{\varepsilon} \implies \varepsilon \cdot \mathrm{OPT} \geq \lambda \cdot d^*.$$

The sum of the k heaviest points upper bounds $g(S^*)$, so we have

$$\mathrm{ALG} \geq g(S^*) = \mathrm{OPT} - \lambda \cdot d^* \geq (1-\varepsilon) \cdot \mathrm{OPT}.$$

Now we consider the case where $d^* > \varepsilon d_{\max}$. `GIST` tries a threshold $d \in [d^*/(2(1+\varepsilon)), d^*/2)$, so Lemma C.1 implies that $\texttt{GreedyIndependentSet}(V, g, d, k)$ outputs a set T such that

$$f(T) \geq g(S^*) + \lambda \cdot d \geq g(S^*) + \lambda \cdot \frac{d^*}{2(1+\varepsilon)}. \tag{8}$$

The max-diameter check on Lines 3–6 give us the lower bound

$$\mathrm{ALG} \geq \lambda \cdot d_{\max} \geq \lambda \cdot d^*. \tag{9}$$

Combining (8) and (9), the following inequality holds for any $0 \leq p \leq 1$:

$$\begin{aligned} \mathrm{ALG} &\geq p \cdot \left[g(S^*) + \lambda \cdot \frac{d^*}{2(1+\varepsilon)} \right] + (1-p) \cdot \lambda \cdot d^* \\ &= p \cdot g(S^*) + \left(1 - p + \frac{p}{2(1+\varepsilon)} \right) \cdot \lambda \cdot d^*. \end{aligned}$$

To maximize the approximation ratio as $\varepsilon \to 0$, we set $p = 2/3$ by solving $p = 1 - p/2$. Therefore,

$$\begin{aligned} \mathrm{ALG} &\geq \frac{2}{3} \cdot g(S^*) + \left(1 - \frac{2}{3} + \frac{1}{3(1+\varepsilon)} \right) \cdot \lambda \cdot d^* \\ &= \frac{2}{3} \cdot g(S^*) + \frac{1}{3} \left(1 + \frac{1}{1+\varepsilon} \right) \cdot \lambda \cdot d^* \\ &\geq \frac{2}{3} \cdot g(S^*) + \frac{1}{3} (2 - \varepsilon) \cdot \lambda \cdot d^* \\ &\geq \left(\frac{2}{3} - \varepsilon \right) \cdot \mathrm{OPT}, \end{aligned} \tag{10}$$

which completes the proof. □

D Missing analysis for Section 4

D.1 Proof of Theorem 4.1

Theorem 4.1. *It is* NP*-hard to approximate* MDMS *within a factor of* $\frac{2(1-1/e)}{2(1-1/e)+1} + \delta < 0.55836 + \delta$*, for any* $\delta > 0$*.*

Proof. We construct our instance based on the collection of sets above as follows. We set $\varepsilon = \min\{\delta, \delta^2/6\}$ and $c = 1$. As mentioned above, we can assume that $k > 1/\varepsilon + 1$ since there is always a brute force polynomial time algorithm for constant k to distinguish between YES and NO instances.

Suppose there are n sets $S_1, S_2, \cdots, S_n$ in the collection $\mathcal{F}$. We represent these n sets with n points in the MDMS instance. We overload the notation to show the corresponding point with S_i too. For any subset of sets/points T, the submodular value $g(T)$ is defined as the cardinality of the union of corresponding sets, i.e., the *coverage submodular function*. In other words, $g(T) = |\bigcup_{S_i \in T} S_i|$.

The distances between points/sets are either d or $2d$. If two sets are disjoint and belong to two separate partitions $\mathcal{F}_r$ and $\mathcal{F}_{r'}$, their distance is $2d$. Otherwise, we set their distance to d. So for any two sets S, S' belonging to the same group $\mathcal{F}_r$, their distance is set to d. Also, for any two sets with nonempty intersection, we set their distance to d too. Any other pair of points will have distance $2d$.

We set d to be $(1 - 1/e)U$ where U is the cardinality of the union of all sets, i.e., $U = |\bigcup_{S \in \mathcal{F}} S|$. Finally, we set $\lambda = 1$ to complete the construction of the MDMS instance.

In the YES case, the optimum solution is the family of k disjoint sets from different groups that cover the entire universe. The value of optimum in this case is $2d + U = (2(1 - 1/e) + 1)U$.

The algorithm has two possibilities: (case a) the algorithm gives up on the diversity objective and selects k sets with minimum distance d, or (case b) it aims for a diversity term of $2d$. In case (a), the submodular value is at most $(1 - (1 - 1/k)^k + \varepsilon)U$ because of the property of the NO case. We note that if the algorithm finds a set of k sets with union size above this threshold, we can conclude that we have a YES instance, which contradicts the hardness result.

The limit of the upper bound for the submodular value as k goes to infinity is $1 - 1/e + \varepsilon$. We use its expansion series to derive:

$$\left(1 - \frac{1}{k}\right)^k \geq \frac{1}{e} - \frac{1}{2ek} - \frac{5}{24ek^2} - \cdots$$

The sequence of negative terms declines in absolute value with a rate of at least $1/k$. Thus, the absolute value of their total sum is at most the first deductive term $1/2ek$ times $1/(1 - 1/k) = k/(k-1)$, which gives the simpler bound:

$$\left(1 - \frac{1}{k}\right)^k \geq \frac{1}{e} - \frac{1}{2e(k-1)}.$$

Since $k - 1$ is at least $1/\varepsilon$, the submodular value in case (a) does not exceed:

$$\left(1 - \frac{1}{e} + \frac{1}{2e(k-1)} + \varepsilon\right)U \leq \left(1 - \frac{1}{e} + 2\varepsilon\right)U.$$

Recall that $d = (1 - 1/e)U$, so the ratio of what the algorithm achieves in case (a) and the optimum solution of YES case is at most:

$$\begin{aligned} \frac{(1 - 1/e + 2\varepsilon)U + d}{U + 2d} &= \frac{2(1 - 1/e) + 2\varepsilon}{2(1 - 1/e) + 1} \\ &\leq \frac{2(1 - 1/e)}{2(1 - 1/e) + 1} + \delta. \end{aligned}$$

In case (b), the algorithm is forced to pick only disjoint sets to maintain a minimum distance of $2d$. This means if the algorithm picks ℓ sets, their union has size $\ell \cdot U/k$. This is true because all sets in $\mathcal{F}$ have the same size and we know in the YES case, the union of k disjoint sets covers the entire

universe hence each set has size U/k. For the special case of $\ell = 1$, we know that only a $1/k < \varepsilon$ fraction of universe is covered. We upper bound the covered fraction in terms of ε for the other cases. The property of the NO case implies the following upper bound on ℓ:

$$\frac{\ell}{k} \leq 1 - \left(1 - \frac{1}{k}\right)^{\ell} + \varepsilon.$$

We use the binomial expansion of $(1 - 1/k)^{\ell}$ and note that each negative term exceeds its following positive term in absolute value. This is true because of the cardinality constraint $\ell \leq k$. Therefore,

$$\begin{aligned}\left(1 - \frac{1}{k}\right)^{\ell} &\geq 1 - \frac{\ell}{k} + \frac{\ell(\ell-1)}{2k^2} - \frac{\ell(\ell-1)(\ell-2)}{6k^3} \\ &\geq 1 - \frac{\ell}{k} + \frac{\ell(\ell-1)}{3k^2} \\ &\geq 1 - \frac{\ell}{k} + \frac{\ell^2}{6k^2},\end{aligned}$$

where the second to last inequality holds since $\ell - 2 < k$ and the last inequality holds because $\ell \geq 2$ and consequently $\ell - 1 \geq \ell/2$. We can now revise the initial inequality:

$$\begin{aligned}\frac{\ell}{k} &\leq (1 - (1 - 1/k)^{\ell} + \varepsilon) \\ &\leq 1 - 1 + \frac{\ell}{k} - \frac{\ell^2}{6k^2} + \varepsilon \implies \frac{\ell^2}{6k^2} \leq \varepsilon.\end{aligned}$$

Thus, the fraction of the universe that the algorithm covers, namely ℓ/k, is at most $\sqrt{6\varepsilon} \leq \delta$.

In case (b), the ratio of what the algorithm achieves, and the optimum solution of YES case is at most:

$$\frac{\delta \cdot U + 2d}{U + 2d} = \frac{2(1 - 1/e) + \delta}{2(1 - 1/e) + 1} \leq \frac{2(1 - 1/e)}{2(1 - 1/e) + 1} + \delta.$$

This concludes the proof in both cases (a) and (b). □

D.2 Proof of Theorem 4.2

Theorem 4.2. *For any $\varepsilon > 0$, there is no polynomial-time $(2/3 + \varepsilon)$-approximation algorithm for the* MDMS *problem if g is a linear function, unless* P = NP.

Proof. First, recall that a clique is a subset of vertices in an undirected graph such that there is an edge between every pair of its vertices. Håstad [28] and Zuckerman [64] showed that the maximum clique problem does not admit an $n^{1-\theta}$-approximation for any constant $\theta > 0$, unless NP = P. This implies that there is no constant-factor approximation algorithm for maximum clique. In other words, for any constant $0 < \delta \leq 1$, there exists a graph G and a threshold integer value k such that it is NP-hard to distinguish between the following two cases:

- YES instance: graph G has a clique of size k.
- NO instance: graph G does not have a clique of size greater than δk.

We reduce this instance of the maximum clique decision problem to MDMS with objective function (2) as follows. Represent each vertex of graph G with a point in our ground set. The distance between a pair of points is 2 if there is an edge between their corresponding vertices in G, and it is 1 otherwise.

Use the same threshold value of k (in the YES and NO instance above) for the cardinality constraint on set S, and set each weight $w(v) = \alpha/k$ for some parameter α that we set later in the proof. We also set $\lambda = 1 - \alpha$. In a YES instance, selecting a clique of size k as set S results in the maximum possible value of the objective:

$$\text{OPT} = \alpha \cdot \frac{1}{k} \cdot k + (1 - \alpha) \cdot 2 = 2 - \alpha. \tag{11}$$

In a NO instance, the best objective value that can be achieved in polynomial-time is the maximum of the following two scenarios: (a) selecting k points with minimum distance 1, or (b) selecting at most δk vertices forming a clique with minimum distance 2. The maximum value obtained by any polynomial-time algorithm is then

$$\begin{aligned}\text{ALG} &= \max\{\alpha + (1-\alpha)\cdot 1, \alpha\cdot\delta + (1-\alpha)\cdot 2\} \\ &= \max\{1, 2-(2-\delta)\alpha\}.\end{aligned}$$

We make these two terms equal by setting $\alpha = 1/(2-\delta)$. Thus, the gap between the maximum value any algorithm can achieve in the NO case and the optimum value in the YES case is

$$\frac{1}{2-\alpha} = \frac{1}{2-{}^{1}/_{(2-\delta)}} = \frac{2-\delta}{3-2\delta}.$$

To complete the proof, it suffices to show that the ratio above is at most ${}^{2}/_{3} + \varepsilon$. We separate the ${}^{2}/_{3}$ term as follows:

$$\frac{2-\delta}{3-2\delta} = \frac{{}^{2}/_{3}\cdot(3-2\delta) + {}^{\delta}/_{3}}{3-2\delta} = \frac{2}{3} + \frac{\delta}{9-6\delta}.$$

Therefore, we must choose a value of δ satisfying ${}^{\delta}/_{(9-6\delta)} \le \varepsilon$. Since $\delta \le 1$, the denominator $9-6\delta$ is positive. Equivalently, we want to satisfy:

$$\frac{9-6\delta}{\delta} = \frac{9}{\delta} - 6 \ge \frac{1}{\varepsilon}.$$

By setting $\delta < 9\varepsilon/(1+6\varepsilon)$, we satisfy the required inequality and achieve the inapproximability gap in the theorem statement. □

D.3 Proof of Lemma 4.4

Let $G = (V, E)$ be a simple undirected graph. Our goal is to embed the vertices of V into $\mathbb{R}^d$, for some $d \ge 1$, in a way that encodes the adjacency structure of G. Concretely, we want to construct a function $h_G : V \to \mathbb{R}^d$ such that:

- $\|h_G(u) - h_G(v)\|_2 = 1$ if $\{u, v\} \notin E$, and
- $\|h_G(u) - h_G(v)\|_2 \le 1 - \varepsilon_G$ if $\{u, v\} \in E$,

for the largest possible value of $\varepsilon_G \in (0, 1]$.

Construction. Let $n = |V|$ and $m = |E|$. Augment G by adding a self-loop to each node to get $G' = (V, E')$. We embed V using G' since each node now has positive degree. Let $\deg'(v)$ be the degree of v in G' and $N'(v)$ be the neighborhood of v in G'.

Define a total ordering on E' (e.g., lexicographically by sorted endpoints $\{u, v\}$). Each edge $e \in E'$ corresponds to an index in the embedding dimension $d := |E'| = m + n$. We consider the embedding function that acts as a degree-normalized adjacency vector:

$$h_G(v)_e = \begin{cases} \sqrt{\frac{1}{2\deg'(v)}} & \text{if } v \in e, \\ 0 & \text{if } v \notin e. \end{cases} \tag{12}$$

Lemma 4.4. *Let $G = (V, E)$ be a simple undirected graph with $n = |V|$, $m = |E|$, and max degree Δ. There exists an embedding $h_G : V \to \mathbb{R}^{n+m}$ such that if $\{u, v\} \in E$ then*

$$\|h_G(u) - h_G(v)\|_2 \le 1 - \frac{1}{2(\Delta+1)},$$

and if $\{u, v\} \notin E$ then $\|h_G(u) - h_G(v)\|_2 = 1$.

Proof. If $\{u, v\} \notin E$, then we have

$$\begin{aligned}
\|h_G(u) - h_G(v)\|_2^2 &= \sum_{e \in N'(u)} \left(\sqrt{\frac{1}{2\deg'(u)}} - 0 \right)^2 + \sum_{e \in N'(v)} \left(\sqrt{\frac{1}{2\deg'(v)}} - 0 \right)^2 \\
&= \left(\frac{1}{2} \sum_{e \in N'(u)} \frac{1}{\deg'(u)} \right) + \left(\frac{1}{2} \sum_{e \in N'(v)} \frac{1}{\deg'(v)} \right) \\
&= \frac{1}{2} + \frac{1}{2} \\
&= 1.
\end{aligned}$$

This follows because the only index where both embeddings can be nonzero is $\{u, v\}$, if it exists.

Now suppose that $\{u, v\} \in E$. It follows that

$$\begin{aligned}
&\|h_G(u) - h_G(v)\|_2^2 \\
&= \sum_{e \in N'(u) \setminus \{v\}} \frac{1}{2\deg'(u)} + \sum_{e \in N'(v) \setminus \{u\}} \frac{1}{2\deg'(v)} + \left(\sqrt{\frac{1}{2\deg'(u)}} - \sqrt{\frac{1}{2\deg'(v)}} \right)^2 \\
&= \frac{\deg'(u) - 1}{2\deg'(u)} + \frac{\deg'(v) - 1}{2\deg'(v)} + \left(\frac{1}{2\deg'(u)} + \frac{1}{2\deg'(v)} - 2\sqrt{\frac{1}{4\deg'(u)\deg'(v)}} \right) \\
&= \frac{1}{2} + \frac{1}{2} - \sqrt{\frac{1}{\deg'(u)\deg'(v)}} \\
&\leq 1 - \frac{1}{\Delta + 1}.
\end{aligned}$$

The previous inequality follows from $\deg'(v) = \deg(v) + 1 \leq \Delta + 1$. For any $x \in [0, 1]$, we have

$$\sqrt{1 - x} \leq 1 - \frac{x}{2},$$

so it follows that

$$\|h_G(u) - h_G(v)\|_2 \leq \sqrt{1 - \frac{1}{\Delta + 1}} \leq 1 - \frac{1}{2(\Delta + 1)},$$

which completes the proof. □

D.4 Proof of Theorem 4.5

Theorem 4.5. MDMS *is* APX*-complete for the Euclidean metric if g is a linear function.*

Proof. We build on the hardness of approximation for the maximum independent set problem for graphs with maximum degree $\Delta = 3$. Alimonti and Kann [4, Theorem 3.2] showed that this problem is APX-complete, so there exists an $\varepsilon_0 > 0$ such that there is no polynomial-time $(1 - \varepsilon_0)$-approximation algorithm unless NP = P. Hence, there exists a graph G with max degree $\Delta = 3$ and a threshold integer value k such that it is NP-hard to distinguish between the following two cases:

- YES instance: graph G has an independent set of size k.
- NO instance: graph G does not have an independent set of size greater than $(1 - \varepsilon_0)k$.

We reduce this instance of bounded-degree maximum independent set to MDMS with objective function (2) as follows. Embed each node of the graph G into Euclidean space using the function $h_G(v)$ in Lemma 4.4. We use the same threshold value of k (between YES and NO instances above) for the cardinality constraint on set S, and we set each weight $w(v) = \alpha/k$ for some parameter α that we set later in the proof. We also set $\lambda = 1 - \alpha$.

In a YES instance, selecting an independent set of size k as the set S results in the maximum value of objective (2):

$$\text{OPT} = \alpha \cdot \frac{1}{k} \cdot k + (1-\alpha) \cdot 1 = 1,$$

since $\|h_G(u) - h_G(v)\|_2 = 1$ for any two distinct points $u, v \in S$ since there is no edge between u and v in graph G.

In a NO instance, the best objective value that can be achieved in polynomial-time is the maximum of the following two scenarios: (a) selecting k points with minimum distance at most $1-1/(2(\Delta+1)) = 1-1/8$, or (b) selecting at most $(1-\varepsilon_0)k$ vertices forming an independent set with minimum distance equal to 1. The maximum value obtained by any polynomial-time algorithm is then

$$\begin{aligned}\text{ALG} &= \max\{\alpha(1-\varepsilon_0) + (1-\alpha)\cdot 1, \alpha + (1-\alpha)(1-1/8)\} \\ &= \max\{1 - \varepsilon_0 \cdot \alpha, (7+\alpha)/8\}.\end{aligned}$$

We make these two terms equal by setting $\alpha = 1/(1+8\varepsilon_0)$. Therefore, the gap between the maximum value any algorithm can achieve in the NO case and the optimum value in the YES case is upper bounded by

$$1 - \varepsilon_0 \cdot \alpha = 1 - \frac{\varepsilon_0}{1/\varepsilon_0 + 8} = 1 - \varepsilon_1.$$

Since $\varepsilon_0 > 0$ is a constant, $\varepsilon_1 \coloneqq \varepsilon_0/(1/\varepsilon_0 + 8) > 0$ is also a constant. This completes the proof of APX-completeness. □

E Additional details for Section 5

E.1 Synthetic dataset

We extend our comparison of baseline algorithms in Section 5.1 by sweeping over values of α, β in the objective function.

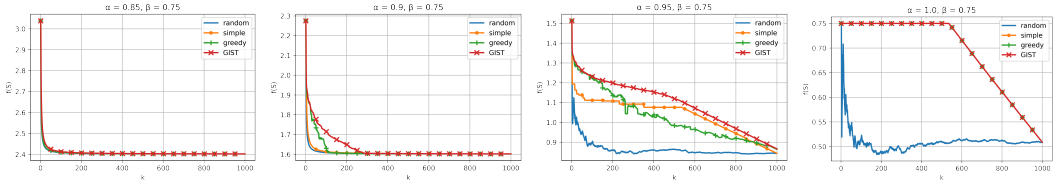

Figure 2: Baseline comparison with $\alpha \in (0.85, 0.90, 0.95, 1.00)$ and $\beta = 0.75$.

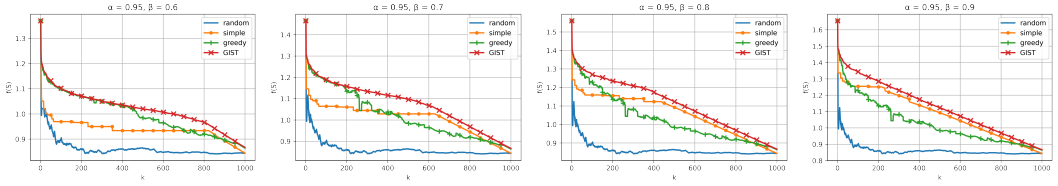

Figure 3: Baseline comparison with $\alpha = 0.95$ and $\beta \in (0.60, 0.70, 0.80, 0.90)$.

E.2 Image classification

Hyperparameters for ImageNet classification. We generate predictions and embeddings for all points using a coarsely-trained ResNet-56 model [29] trained on a random 10% subset of ImageNet [54]. We use SGD with Nesterov momentum 0.9 with 450/90 epochs. The base learning rate is 0.1, and is reduced by a tenth at 5, 30, 69, and 80. We extract the penultimate layer features to produce 2048-dimensional embeddings of each image. We use the same hyperparameters as the original ResNet paper [29] with budgets and one-shot subset selection experiments designed in the same manner as [49].

Running times. The end-to-end running time is dominated by ImageNet model training, which takes more than a few hours even with accelerators (e.g., GPU/TPU chips). The subset selection algorithms that use margin and submodular sampling range between 3–4 minutes per run on an average. `GIST` subset selection is similar to the margin or submodular algorithms for a given distance threshold d. By using parallelism for different d values, we can keep the `GIST-submod` subset selection runtime the same as the `submod` algorithm. In summary, the actual subset selection algorithm step is extremely fast (nearly negligible) compared to the ImageNet training time. Furthermore, we can exploit distributed submodular subset selection algorithms that can even handle billions of data points efficiently [15].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction are short summaries of the results in our paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We state that the theoretical analysis of the approximation guarantees are for the well-studied worst-case analysis framework. For the hardness results, we state the underlying computational complexity assumptions. For empirical results, we describe our findings about the specific experiments we ran and do not generalize claims beyond those.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide the complete proofs of the approximation guarantee of our algorithm and the hardness results, including the underlying computational complexity assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The pseudocode for `GIST` is provided in Algorithm 1 along with the necessary experimental settings/hyperparameters in Section 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experimental setup and hyperparameters are provided in Section 5 and Appendix E.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes, we provide exact dataset, model architecture, and hyperparameters for model training in Section 5 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The plot of $f(S)$ for the `random` algorithm in Figure 1 is just for one trial to highlight the non-monotonicity of the objective function (as opposed to plotting the mean). The error bars for top-1 classification accuracies are provided in Table 1 for all the algorithms considered: `random`, `margin`, k-`center`, `submod`, `GIST-margin`, and `GIST-submod`.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The experiments we present are medium size and can run a personal machine.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The paper conforms with the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: To the best of our knowledge, our work does not have negative societal impacts. The newly designed algorithms are for data summarization and data curation steps of the learning pipeline and are orthogonal to further tasks involved in societal impacts.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The only data introduced in this paper is a set of Gaussian points with uniform random weights.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The owners of ImageNet [54] and ResNet-56 [29] are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only asset introduced is a set of Gaussian points with uniform random weights. We use a standard NumPy setup to generate these points and weights with $\texttt{seed} = 0$. Moreover, we expect the results to hold for any seed by concentration.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this paper does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.