Stable Maximum Coverage Under a Cardinality Constraint

Themistoklis Haris
Boston University
tharis@bu.edu

Fabian Spaeh
Celonis
fspaeh@bu.edu

Nithin Varma
University of Cologne
varma@cs.uni-koeln.de

Yuichi Yoshida
National Institute of Informatics
yyoshida@nii.ac.jp

Abstract

We propose a stable algorithm for solving maximum coverage problems under a cardinality constraint of k in a record-level stability model, where adjacent instances $I \sim I'$ differ by the incidence of one universe element. Our algorithm computes a negative-entropy-regularized maximizer of the capped concave relaxation over the hypersimplex and rounds it with a common-seed rounding map from hypersimplex correlated sampling. The resulting sensitivity bound is $\min\{2k, \alpha_k k/\lambda\}$, where λ is the regularization parameter and $\alpha_k = O(\log k)$. For utility, the expected coverage has approximation factor $1 - \frac{1}{e}$ with an additive error of order $O(\lambda k \log \frac{m}{k})$, where m is the number of input sets. We also report experiments on synthetic and real-world instances.

1 Introduction

Maximum coverage asks for a small subfamily of sets whose union is as large as possible. It is one of the canonical examples of monotone submodular maximization under a cardinality constraint, and it also models many subset-selection tasks built from incidence data. In a recommendation system, for instance, the selectable sets may be candidate items or content bundles and a universe element may be a user request or user segment. In facility-location or service-placement problems, the sets are candidate facilities and the universe elements are demand points. In advertising, monitoring, and sensor-placement applications, the sets represent campaigns, probes, or sensors, and each universe element records an impression, event, or location that can be covered. In all of these examples, the algorithm must output a concrete list of selected actions, not only estimate the optimum value.

Stability under universe-element membership matters when the incidence data are edited or updated. A single record can be corrected, inserted, deleted, or reclassified; a single user may change the list of acceptable recommendations; or one demand point may gain or lose access to several facilities. If the output set changes substantially after such a one-record perturbation, then downstream systems see churn even though the underlying instance changed only locally. We therefore need algorithms that approximate the maximum coverage value while keeping the selected subset stable under record-level perturbations.

Given this, we initiate the study of stability for approximation algorithms for the cardinality-constrained maximum coverage problem. Formally, an input instance I consists of a universe U and a collection of sets $F_1, \ldots, F_m \subseteq U$. We refer to $[m]$ as the ground set. Two instances I and I' are neighbors if they differ only in the membership of a single universe element $e \in U$ in the sets $F_1, \ldots F_m$. An algorithm for the k-maximum coverage problem must, on input I, output exactly k sets from the collection $F_1, \ldots, F_m$. The value of an output is the cardinality of the union of the

selected sets. Sensitivity is measured by running the algorithm on two neighboring instances with the same random seed and bounding the expected symmetric difference of the two output subsets.

Our Results We provide the first stable algorithm for maximum coverage under a cardinality constraint. Its parameter λ gives an explicit tradeoff between output sensitivity and utility. Formally, for an instance I and seed ω, let $S_I(\omega)$ denote the algorithm's output and let $f_I(S_I(\omega))$ denote its coverage value on I. If $I \sim I'$ are adjacent instances, meaning that they differ only in the incidence pattern of one universe element, then running the algorithm on I and I' with the same random seed (sampled once) gives

$$\mathbb{E}_\omega[|S_I(\omega) \triangle S_{I'}(\omega)|] \leq \min\left\{2k, \alpha_k \frac{k}{\lambda}\right\},$$

where $\triangle$ denotes the symmetric difference, and

$$\mathbb{E}_\omega[f_I(S_I(\omega))] \geq (1 - e^{-1})\mathrm{OPT}_I - (1 - e^{-1})\lambda k \log(m/k).$$

Here OPT_I is the maximum possible coverage on instance I, $\lambda > 0$ is a tunable parameter that trades off sensitivity and utility, and $\alpha_k = O(\log k)$ is a factor depending only on k. The multiplicative factor $1 - e^{-1}$ matches the optimal worst-case approximation threshold: a better ratio is NP-hard [Feige, 1998]. The end-to-end guarantee is stated formally in Theorem 1.

Our Techniques Our proof has two main ingredients. First, we optimize the capped concave relaxation of the coverage function with *negative-entropy regularization*. We show that the resulting fractional maximizer is stable in the ℓ_1 norm. Second, we round this fractional point to an integral one. Classical rounding schemes such as pipage rounding do not preserve sensitivity, so our algorithm instead uses a *hypersimplex correlated sampling* algorithm (adapted from [Naor et al., 2026]) to obtain the final solution.

1.1 Related work and positioning

Maximum coverage and cardinality-constrained submodular maximization. Maximum coverage is a classical monotone submodular maximization problem under a cardinality constraint. The greedy algorithm achieves the optimal $(1 - 1/e)$ approximation for this setting in the standard worst-case sense [Nemhauser et al., 1978]. This factor is tight for maximum coverage under standard complexity assumptions and in the oracle model [Feige, 1998, Nemhauser and Wolsey, 1978]. The same modeling pattern underlies influence maximization and many other coverage-like objectives [Kempe et al., 2003]. Our work keeps this approximation benchmark, but changes the algorithmic requirement: the output set must also have low sensitivity to a one-record change in the incidence structure.

Stability, sensitivity, and replicability. Algorithmic stability has several meanings across theoretical computer science and learning theory. In learning, stability often controls the effect of changing one training example on a learned predictor and is used to prove generalization guarantees [Bousquet and Elisseeff, 2002]. In graph algorithms and approximation algorithms, sensitivity asks how much the output changes after a local update, and recent work gives both upper bounds and lower bounds for such notions [Varma and Yoshida, 2019, Fleming and Yoshida, 2024]. For monotone submodular maximization, sensitivity of the chosen argument under input deletions has also been studied [McMeel and Yoshida, 2020]. Replicability in learning asks for the same output, with high probability, when an algorithm is rerun with fixed randomness on fresh samples from the same distribution [Impagliazzo et al., 2022]. Our notion is closest in spirit to output sensitivity and replicability, but it is tailored to maximum coverage: adjacency is a change to one universe element's incidence pattern, and the distance between outputs is the symmetric difference of two size-k subsets under a common random seed.

Difference from differential privacy. Differential privacy provides a distributional indistinguishability guarantee for outputs on neighboring databases, usually by ensuring that likelihood ratios are bounded after suitable randomization [Dwork et al., 2006, Dwork and Roth, 2014]. There are known conceptual connections among privacy, stability, adaptive generalization, and replicability [Bun et al., 2023]. Our guarantee is deliberately different: it is a coupling statement for two executions that use the same randomness. We seek to control the expected change in the set output, rather than the information revealed about the changed universe element.

Correlated sampling, dependent rounding, and hypersimplex rounding. Our rounding step sits at the intersection of dependent rounding and correlated sampling. Classical rounding methods, including pipage rounding, dependent rounding in bipartite graphs, and swap rounding, enforce hard combinatorial constraints while preserving marginals and/or objective guarantees, and have become standard tools in approximation algorithms [Ageev and Sviridenko, 2004, Gandhi et al., 2006, Chekuri et al., 2010]. Correlated sampling studies how to use shared randomness so that nearby input distributions produce nearby samples [Bavarian et al., 2020]. The hypersimplex setting combines these two requirements: the sample must be a size-k set with prescribed marginals, and two nearby fractional points should be rounded to nearby sets under a common seed. We use the dimension-free hypersimplex correlated sampling result of Naor et al. [2026] in Section 5.

Continuous greedy and multilinear-extension approaches. For general monotone submodular maximization under matroid constraints, the multilinear extension, continuous greedy, and contention-resolution framework gives $(1-1/e)$-type approximation guarantees [Calinescu et al., 2011, Chekuri et al., 2011]. In a stability analysis, however, these methods can introduce sensitivity parameters tied to the geometry of the multilinear process and to overlap-dependent quantities.

Additional extensions and comparisons are deferred to the appendix.

2 Problem Setup

Fix integers $m \geq 1$ and $1 \leq k \leq m$. Let U be a finite universe, and let $F_1, \ldots, F_m \subseteq U$ be the available sets. We write $I = (F_1, \ldots, F_m)$ for an instance and identify the family with the ground set $[m] = \{1, \ldots, m\}$ of selectable sets. For a universe element $u \in U$, let

$$N_I(u) := \{i \in [m] : u \in F_i\}$$

denote the sets that cover u. For a subset $S \subseteq [m]$, the maximum coverage objective is

$$f_I(S) := \left| \bigcup_{i \in S} F_i \right| = \sum_{u \in U} \mathbb{1}[S \cap N_I(u) \neq \varnothing].$$

We work under the cardinality constraint $|S| \leq k$. The optimal value is

$$\mathrm{OPT}_I := \max\{f_I(S) : S \subseteq [m],\ |S| \leq k\} = \max\{f_I(S) : S \subseteq [m],\ |S| = k\}.$$

Indeed, f_I is monotone, so any feasible set $S \subseteq [m]$ with $|S| < k$ can be extended to a superset of size exactly k without decreasing its value. From this point on, we therefore work with the exact-k formulation and its fractional hypersimplex relaxation.

We also use fractional points on the hypersimplex slice

$$B_{k,m} := \left\{ x \in [0,1]^m : \sum_{i=1}^{m} x_i = k \right\}.$$

For $x \in \mathbb{R}^m$ and $A \subseteq [m]$, we let $x(A) = \sum_{i \in A} x_i$.

2.1 Stable Maximum Coverage

We seek a stable algorithm $\mathcal{A}$ that outputs a size-k subset while maximizing f_I. We define the sensitivity of $\mathcal{A}$ as a global upper bound on the difference between the size-k solutions $\mathcal{A}$ finds on any two *adjacent* instances $I \sim I'$. Throughout, the ambient universe U is fixed; under this convention, insertions and deletions are represented by letting one designated element move between the empty incidence pattern and a nonempty incidence pattern, while reclassification changes one incidence pattern to another.

Definition 1 (Adjacency)**.** *Two instances I and I' are adjacent, written $I \sim I'$, if they differ only in the incidence pattern of one universe element: there exists $e \in U$ such that*

$$N_I(u) = N_{I'}(u) \qquad \textit{for every } u \in U \setminus \{e\}.$$

Stability is formally defined for randomized algorithms through a common-seed coupling.

Definition 2 (Stable Algorithm)**.** *A randomized algorithm $\mathcal{A}$ for maximum coverage is represented as $\mathcal{A}(I;\omega)$, where I is the input instance and ω is the random seed, including all internal randomness of the algorithm. We say that $\mathcal{A}$ is s-stable if*

$$\sup_{I\sim I'} \mathbb{E}_\omega \left[|\mathcal{A}(I;\omega)\triangle\mathcal{A}(I';\omega)|\right] \le s.$$

The two outputs in the symmetric difference are obtained by running $\mathcal{A}$ on the two adjacent instances with the same seed ω. Thus this is a common-seed coupling guarantee, not the expected distance between two independent executions using independent random seeds.

3 End-to-End Algorithm and Main Result

In this section we present our end-to-end algorithm and state our main theorem. While our algorithm follows the standard two-stage scheme of building a fractional solution and then rounding it, each stage has to guarantee solution quality and stability. A key contribution of our work is thus to apply an appropriate continuous relaxation and regularizer, and a compatible rounding scheme.

Stable Maximum Coverage Algorithm $\mathcal{A}$

Fractional stage. Compute a negative-entropy-regularized concave relaxation objective:

$$x_I^\lambda \in \arg\max_{x\in B_{k,m}} \{H_I(x) - \lambda R(x)\},$$

where $\lambda > 0$ is a regularization strength parameter and

$$H_I(x) := \sum_{u\in U} \min\{1, x(N_I(u))\} \quad \text{and} \quad R(x) := \sum_{i=1}^{m} x_i \log x_i$$

is the negative-entropy regularizer.

Rounding stage. Using the random seed ω, sample a deterministic common-seed rounding map Round_ω on the hypersimplex from Theorem 4 and output

$$S_I(\omega) := \mathrm{Round}_\omega(x_I^\lambda) \subseteq [m].$$

Our main result is the following:

Theorem 1. *For any instance I, let $S_I(\omega)$ denote the output of algorithm $\mathcal{A}$ on seed ω. Then $|S_I(\omega)| = k$ almost surely, and for every adjacent pair $I \sim I'$,*

$$\mathbb{E}_\omega \left[|S_I(\omega)\triangle S_{I'}(\omega)|\right] \le \min\left\{2k, \alpha_k \frac{k}{\lambda}\right\},$$

where $\alpha_k = O(1+\log k)$. Moreover,

$$\mathbb{E}_\omega[f_I(S_I(\omega))] \ge (1-e^{-1})\mathrm{OPT}_I - (1-e^{-1})\lambda k \log(m/k).$$

Choosing the parameter λ appropriately yields a clean sensitivity–utility tradeoff.

Corollary 1. *Fix a target sensitivity budget $s > 0$ and choose $\lambda := \alpha_k k/s$. Then the resulting algorithm $\mathcal{A}$ is s-stable under Definition 2, and for every instance I,*

$$\mathbb{E}_\omega[f_I(S_I(\omega))] \ge (1-e^{-1})\mathrm{OPT}_I - (1-e^{-1})\alpha_k \frac{k^2 \log(m/k)}{s}.$$

Since the set output has cardinality exactly k, one always has $|S_I(\omega)\triangle S_{I'}(\omega)| \le 2k$. Thus the corollary remains true for any $s > 0$, but the meaningful nontrivial sensitivity-budget regime is $0 < s \le 2k$.

Analysis and Proof Strategy Sections 4–6 prove Theorem 1 by following a single route:

$$H_I \longrightarrow x_I^\lambda \longrightarrow \mathrm{Round}_\omega(x_I^\lambda) \longrightarrow S_I(\omega).$$

We first show that obtaining fractional solutions x_I^λ can be done in a stable fashion via negative-entropy-regularized concave relaxation. Then, we analyze the proposed common-seed rounding scheme from hypersimplex correlated sampling to obtain the final integral solution.

4 Stage 1: Stable Fractional Relaxation

This section develops the fractional half of the argument. We use the capped concave relaxation of maximum coverage because it keeps the sensitivity analysis on the hypersimplex. This analysis may also be useful for other coverage-type objectives.

The Concave and Multilinear Extensions The maximum coverage objective is a submodular function and so it is common to study it through its continuous extensions. We will make use of two commonly studied such extensions.

Definition 3 (Multilinear Extension of Maximum Coverage)**.** *The multilinear extension of f_I is*

$$F_I(x) := \sum_{u \in U} \left(1 - \prod_{i \in N_I(u)} (1 - x_i)\right), \qquad x \in [0,1]^m.$$

Similarly, f_I is also often studied through its concave relaxation.

Definition 4 (Concave Relaxation of Maximum Coverage)**.** *The concave relaxation of f_I is:*

$$H_I(x) := \sum_{u \in U} \min\{1, x(N_I(u))\}.$$

The two continuous extensions are related in terms of utility. We shall rely on the following properties of the concave relaxation:

Lemma 1. *For every instance I, the function H_I has the following properties.*

1. *H_I is concave on $[0,1]^m$.*
2. *For every $S \subseteq [m]$, $H_I(\mathbb{1}_S) = f_I(S)$.*
3. *For every $x \in [0,1]^m$, $F_I(x) \geq (1 - e^{-1}) H_I(x)$.*

Proof. See Appendix A. □

4.1 Analyzing Negative-Entropy-Regularized Concave Relaxation

The fractional part of Algorithm $\mathcal{A}$ solves a regularized program aimed at maximizing H_I with negative-entropy regularization on the hypersimplex.

$$x_I^\lambda \in \arg\max_{x \in B_{k,m}} \{H_I(x) - \lambda R(x)\}, \qquad R(x) := \sum_{i=1}^m x_i \log x_i. \tag{1}$$

Here R is the negative-entropy regularizer. As usual, we adopt the convention $0 \log 0 := 0$. To understand the properties of the resulting solution x_I^λ, we analyze both its utility and sensitivity.

4.1.1 Approximation Guarantee

First we analyze the approximation guarantee. By the second property in Lemma 1, the approximation error only appears through the regularization term, which is bounded by Jensen's inequality.

Theorem 2. *For every instance I, we have:*

$$H_I(x_I^\lambda) \geq \mathrm{OPT}_I - \lambda k \log(m/k),$$

Proof. Let $S^\star \subseteq [m]$ be an optimal size-k solution, which exists by the exact-k reformulation in Section 2. Then $\mathbb{1}_{S^\star} \in B_{k,m}$. By Lemma 1,

$$H_I(\mathbb{1}_{S^\star}) = f_I(S^\star) = \mathrm{OPT}_I \qquad \text{and} \qquad R(\mathbb{1}_{S^\star}) = 0.$$

Optimality of x_I^λ for (1) gives

$$H_I(x_I^\lambda) - \lambda R(x_I^\lambda) \geq H_I(\mathbb{1}_{S^\star}) - \lambda R(\mathbb{1}_{S^\star}) = \mathrm{OPT}_I,$$

so

$$H_I(x_I^\lambda) \geq \mathrm{OPT}_I + \lambda R(x_I^\lambda).$$

It remains to lower-bound $R(x_I^\lambda)$. Let $x_{I,i}^\lambda$ for $i \in [m]$ denote the i-th coordinate of x_I^λ. Since $\sum_i x_{I,i}^\lambda = k$, Jensen's inequality yields

$$R(x_I^\lambda) = \sum_{i=1}^{m} x_{I,i}^\lambda \log x_{I,i}^\lambda \geq k \log(k/m).$$

Hence

$$H_I(x_I^\lambda) \geq \mathrm{OPT}_I - \lambda k \log(m/k). \qquad \square$$

Applying the third property of Lemma 1 to Theorem 2, we get the following approximation guarantee for the multilinear extension:

Corollary 2. *It holds that:*

$$F_I(x_I^\lambda) \geq (1 - e^{-1})\mathrm{OPT}_I - (1 - e^{-1})\lambda k \log(m/k).$$

4.1.2 Sensitivity Analysis

We now analyze the sensitivity of the fractional solution in the $||\cdot||_1$-norm. The analysis follows the first-order conditions for maximizing a concave function over a convex set. Since H_I is concave, we write $\partial H_I(x)$ for its concave superdifferential at x, namely

$$\partial H_I(x) := \{g \in \mathbb{R}^m : H_I(y) \leq H_I(x) + \langle g, y - x\rangle \text{ for all } y \in [0,1]^m\}.$$

To show the desired sensitivity guarantee we need to control both the entropy curvature (Lemma 3), which can be bounded via Pinsker's inequality, and the perturbation of the supergradients between adjacent instances (Lemma 4). This analysis is provided in detail in Appendix B and enables us to show the following theorem:

Theorem 3. *For every adjacent pair $I \sim I'$,*

$$\|x_I^\lambda - x_{I'}^\lambda\|_1 \leq \frac{k}{\lambda}.$$

Proof. Write $x := x_I^\lambda$ and $x' := x_{I'}^\lambda$. Define

$$\Psi_I(z) := H_I(z) - \lambda R(z) \qquad \text{and} \qquad \Psi_{I'}(z) := H_{I'}(z) - \lambda R(z).$$

Since x maximizes the concave function Ψ_I over the convex set $B_{k,m}$, the first-order optimality condition for nonsmooth concave maximization is

$$0 \in -\partial\Psi_I(x) + N_{B_{k,m}}(x) = -\partial H_I(x) + \lambda\nabla R(x) + N_{B_{k,m}}(x),$$

where

$$N_{B_{k,m}}(x) := \{v \in \mathbb{R}^m : \langle v, u - x\rangle \leq 0 \text{ for all } u \in B_{k,m}\}$$

is the normal cone of $B_{k,m}$ at x. Equivalently, there exists $g \in \partial H_I(x)$ such that

$$\langle g - \lambda\nabla R(x), u - x\rangle \leq 0 \qquad \text{for all } u \in B_{k,m}.$$

Likewise, there exists $g' \in \partial H_{I'}(x')$ such that

$$\langle g' - \lambda\nabla R(x'), u - x'\rangle \leq 0 \qquad \text{for all } u \in B_{k,m}.$$

Applying the first variational inequality with $u = x'$ and the second with $u = x$, and then adding them, gives

$$\langle g - g', x - x'\rangle \geq \lambda\langle\nabla R(x) - \nabla R(x'), x - x'\rangle. \tag{2}$$

By Lemma 3 in the appendix,

$$\lambda\langle\nabla R(x) - \nabla R(x'), x - x'\rangle \geq \frac{\lambda}{k}\|x - x'\|_1^2.$$

Now apply Lemma 4 from the appendix at the common point x' to the supergradient $g' \in \partial H_{I'}(x')$. This yields a supergradient

$$\bar{g} \in \partial H_I(x')$$

such that

$$\|\bar{g} - g'\|_\infty \leq 1.$$

Since $g \in \partial H_I(x)$ and $\bar{g} \in \partial H_I(x')$, the superdifferential of the concave function H_I is anti-monotone:

$$\langle g - \bar{g}, x - x' \rangle \leq 0.$$

Indeed, concavity of H_I gives

$$H_I(x') \leq H_I(x) + \langle g, x' - x \rangle$$

and

$$H_I(x) \leq H_I(x') + \langle \bar{g}, x - x' \rangle.$$

Adding these two inequalities yields the displayed bound. Also,

$$\langle \bar{g} - g', x - x' \rangle \leq \|\bar{g} - g'\|_\infty \|x - x'\|_1 \leq \|x - x'\|_1.$$

Therefore

$$\langle g - g', x - x' \rangle = \langle g - \bar{g}, x - x' \rangle + \langle \bar{g} - g', x - x' \rangle \leq \|x - x'\|_1.$$

Combining this with (2) yields

$$\frac{\lambda}{k} \|x - x'\|_1^2 \leq \|x - x'\|_1.$$

If $x = x'$, there is nothing to prove. Otherwise we cancel $\|x - x'\|_1$ and obtain the stated bound. □

5 Stage 2: Common-Seed Rounding via Hypersimplex Correlated Sampling

We now pass from the stable fractional point $x_I^\lambda \in B_{k,m}$ to an integral size-k solution. We use the dimension-free hypersimplex correlated sampling result of Naor et al. [2026] as the rounding primitive. Let $\Delta_{k,m} := \{x \in [0,1]^m : \|x\|_1 \leq k\} \supset B_{k,m}$.

Theorem 4 (Hypersimplex correlated sampling, specialized from [Naor et al., 2026])**.** *For every $m, k \geq 1$, there exist a constant $\alpha_k = O(\log k)$ and a distribution $\mathcal{D}_{k,m}$ over deterministic maps*

$$\mathrm{Round} : \Delta_{k,m} \to 2^{[m]}$$

such that for all $x, y \in \Delta_{k,m}$ the following hold.

1. ***Cardinality**. For every* Round *in the support of $\mathcal{D}_{k,m}$, $|\mathrm{Round}(x)| \in \{\lfloor \|x\|_1 \rfloor, \lceil \|x\|_1 \rceil\}$.*
2. ***Marginals.** For every $i \in [m]$, we have $\Pr_{\mathrm{Round} \sim \mathcal{D}_{k,m}}[i \in \mathrm{Round}(x)] = x_i$.*
3. ***Stretch.** $\mathbb{E}_{\mathrm{Round} \sim \mathcal{D}_{k,m}}[|\mathrm{Round}(x) \triangle \mathrm{Round}(y)|] \leq \alpha_k \|x - y\|_1$.*
4. ***Submodular objective preservation.** For every submodular function $f : 2^{[m]} \to \mathbb{R}$ with multilinear extension F,*

$$\mathbb{E}_{\mathrm{Round} \sim \mathcal{D}_{k,m}}[f(\mathrm{Round}(x))] \geq F(x).$$

In particular, item 4 applies to every monotone submodular objective.

Remark 1. *Equivalently, the seed ω samples a deterministic map $\mathrm{Round}_\omega \sim \mathcal{D}_{k,m}$. In the common-seed convention of Definition 2, both fractional points are rounded using the same map Round_ω; hence the quantity $|\mathrm{Round}_\omega(x) \triangle \mathrm{Round}_\omega(y)|$ is measured under shared randomness, not independent samples of the rounding map.*

We apply Theorem 4 to the negative-entropy-regularized solutions from Section 4. Let the random seed ω sample a deterministic map

$$\mathrm{Round}_\omega \sim \mathcal{D}_{k,m}.$$

For each instance I, define

$$S_I(\omega) := \mathrm{Round}_\omega(x_I^\lambda).$$

The following theorem establishes our integral rounding guarantees:

Theorem 5. *For every instance I, the rounded solution $S_I(\omega)$ satisfies the following:*

1. $|S_I(\omega)| = k$ *almost surely.*
2. *For every* $i \in [m]$, *we have* $\Pr_\omega[i \in S_I(\omega)] = x^\lambda_{I,i}$.
3. $\mathbb{E}_\omega[f_I(S_I(\omega))] \geq F_I(x^\lambda_I)$.
4. *For every adjacent pair* $I \sim I'$, $\mathbb{E}_\omega[|S_I(\omega) \triangle S_{I'}(\omega)|] \leq \alpha_k \|x^\lambda_I - x^\lambda_{I'}\|_1$.

Proof. Because $x^\lambda_I \in B_{k,m}$, we have $\|x^\lambda_I\|_1 = k$. The cardinality guarantee in Theorem 4 therefore gives

$$|S_I(\omega)| = |\operatorname{Round}_\omega(x^\lambda_I)| \in \{\lfloor k \rfloor, \lceil k \rceil\} = \{k\}$$

almost surely. The marginal, utility, and stretch statements are exactly the corresponding guarantees from Theorem 4 applied to $x = x^\lambda_I$ and $y = x^\lambda_{I'}$, with the randomness taken over the seed ω that samples $\operatorname{Round}_\omega$. □

6 Putting it All Together: The Utility–Stability Tradeoff

We now combine the fractional guarantees from Section 4 with the common-seed rounding theorem from Section 5.

Proof of Theorem 1. The exact-cardinality statement $|S_I(\omega)| = k$ follows from part (1) of Theorem 5. In particular, for every adjacent pair $I \sim I'$, both outputs have size k, so $|S_I(\omega) \triangle S_{I'}(\omega)| \leq 2k$ for every seed ω. On the other hand, part (4) of Theorem 5 and Theorem 3 give

$$\mathbb{E}_\omega\left[|S_I(\omega) \triangle S_{I'}(\omega)|\right] \leq \alpha_k \|x^\lambda_I - x^\lambda_{I'}\|_1 \leq \alpha_k \frac{k}{\lambda}.$$

Combining the two bounds yields

$$\mathbb{E}_\omega\left[|S_I(\omega) \triangle S_{I'}(\omega)|\right] \leq \min\left\{2k, \alpha_k \frac{k}{\lambda}\right\}.$$

For utility, part (3) of Theorem 5 and Corollary 2 yield

$$\mathbb{E}_\omega[f_I(S_I(\omega))] \geq F_I(x^\lambda_I) \geq (1 - e^{-1})\mathrm{OPT}_I - (1 - e^{-1})\lambda k \log(m/k). \qquad \square$$

7 Experimental Results

We evaluate our stable maximum coverage algorithm experimentally. The algorithm is implementable with standard numerical tools. The benchmark we use is the classic greedy algorithm that selectively picks the set with the highest marginal gain until the budget is exhausted. This algorithm does not offer stability guarantees and can in fact exhibit very high sensitivity on neighboring instances where changing one element's incidence pattern leads to different tie-breaking sets being used.

For details on our implementation, design choices and precise experimental setup, see Appendix C. We test our algorithm against three benchmark scenarios that capture common applications of maximum coverage: (1) *random inclusion*, (2) *facility location*, and (3) *influence maximization (IM)*. For the first, we include universe elements randomly in each set; for the second, we simulate a classic geometric facility environment in which coverage of a point is proximity-based. For the IM experiment, we use the live-edge model, treating graph reachability as a coverage function.

Remark 2. *It is important to note that to solve for the fractional point in the first part of our algorithm, our implementation utilizes a classic iterative technique in convex optimization (Frank-Wolfe iterations). Because the objective is convex, that algorithm converges within a polynomial number of steps. However, the error of the numerical solution does affect the final error and stability of the fractional answer, and therefore also the final answer.*

Analyzing the error propagation from the iterative solver is easy and does not change our analyses at all, with the exception of an additional additive term that needs to be accounted for. We omit this analysis here for brevity. Finally, the theoretical analysis does not consider issues of numerical stability that often arise in approximate Frank-Wolfe solvers like SCS.

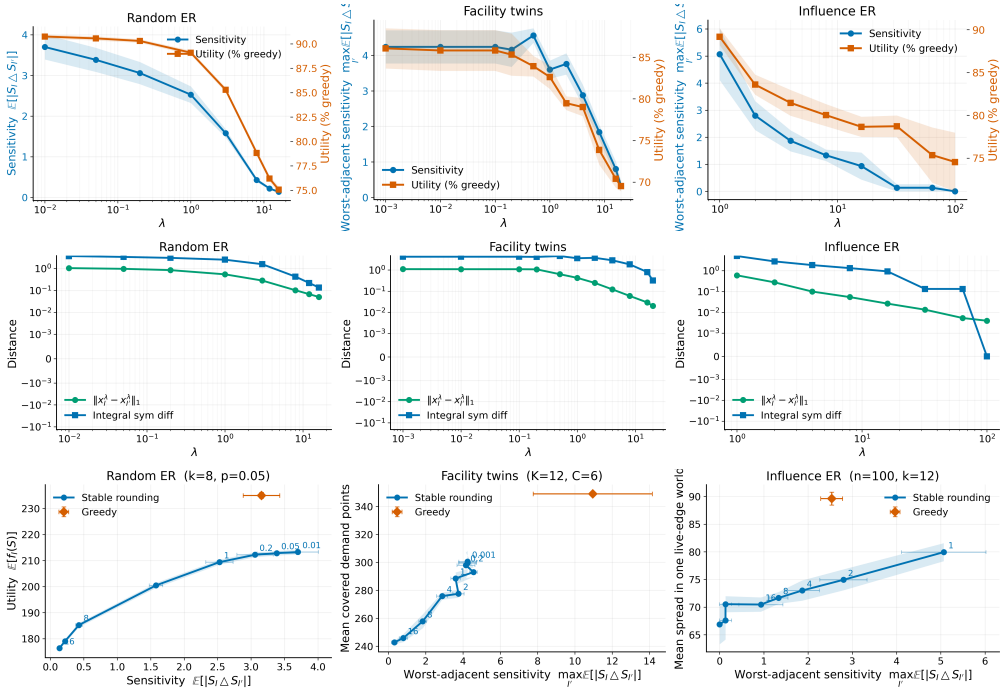

Figure 1: *Top*: Sensitivity–utility tradeoff for many values of λ. *Middle*: Sensitivity of the integral and fractional solutions. *Bottom*: Pareto frontier between sensitivity and utility. The performance of greedy in both metrics is highlighted in red.

Results Results on synthetic instances are in Figure 1 where we track several statistics (mean and standard deviation over 5 seeds). Results on real-world IM instances are in Figure 2.

A larger λ reduces the measured sensitivity, while a smaller λ gives higher utility, matching the tradeoff in Theorem 1. Across the instances we study, the algorithm is more stable than greedy while preserving at least $75\% - 80\%$ of the utility. The middle row shows that a larger λ decreases the ℓ_1 sensitivity of both the fractional and rounded solutions. The bottom row compares the sensitivity–utility Pareto frontier with the greedy algorithm. Greedy attains higher utility in these experiments but is much less stable; our algorithm reduces this instability. Figure 2 shows a similar trend for real-world coverage instances. Whether this Pareto frontier can be improved further remains open.

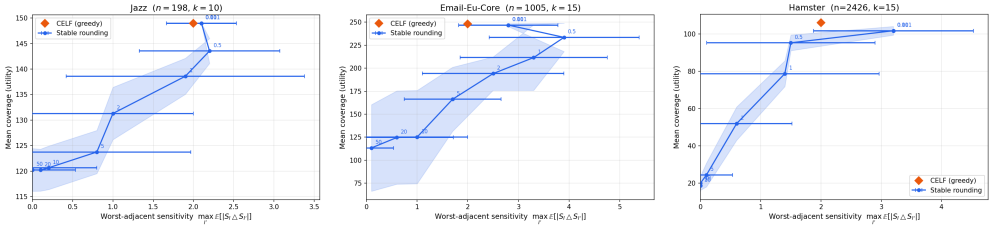

Figure 2: Influence maximization on three real-world datasets. The presented metrics are analogous to the Pareto frontier in Figure 1.

8 Limitations

Though our algorithm is provably stable, it does add an additive error to the answer, limiting it in situations where optimality is of the essence. Also, as we discussed previously, the analysis does not take into account the solver-dependent cost of the fractional stage and the numerical instability issues of Frank-Wolfe optimizers. Finally, there are additional experimental and empirical baselines that can be used to evaluate the algorithm that we did not examine, leaving additional empirical investigation

for future work. For instance, testing on different problems and with larger scale instances can reveal more insights about how the algorithm behaves in practice.

9 Conclusion

We propose the first stable algorithm for maximum coverage. Its parameter λ controls the sensitivity–utility tradeoff. Our analysis combines a negative-entropy-regularized concave relaxation with common-seed rounding via hypersimplex correlated sampling. Our experiments show that the algorithm is more stable than greedy at a moderate utility loss. We expect these techniques to be useful for other problems in constrained submodular optimization.

10 Acknowledgements

The authors thank Sofya Raskhodnikova for her insights during early discussions about this problem.

References

A. A. Ageev and M. I. Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004. doi: 10.1023/B:JOCO.0000038913.96607.c2.

M. Bavarian, B. Ghazi, E. Haramaty, P. Kamath, R. L. Rivest, and M. Sudan. Optimality of correlated sampling strategies. *Theory of Computing*, 16(12):1–18, 2020. doi: 10.4086/toc.2020.v016a012.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

M. Bun, M. Gaboardi, M. Hopkins, R. Impagliazzo, R. Lei, T. Pitassi, S. Sivakumar, and J. Sorrell. Stability is stable: Connections between replicability, privacy, and adaptive generalization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 520–527. ACM, 2023. doi: 10.1145/3564246.3585246.

G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011. doi: 10.1137/080733991.

C. Chekuri, J. Vondrák, and R. Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 575–584. IEEE Computer Society, 2010. doi: 10.1109/FOCS.2010.60.

C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 783–792, 2011.

S. Chen, D. Di Castro, Z. Karnin, L. Lewin-Eytan, J. S. Naor, and R. Schwartz. Correlated rounding of multiple uniform matroids and multi-label classification. In *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, pages 34–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279, 2008.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi: 10.1007/11681878_14.

U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998. doi: 10.1145/285055.285059.

N. Fleming and Y. Yoshida. Sensitivity lower bounds for approximation algorithms. *CoRR*, abs/2411.02744, 2024. doi: 10.48550/ARXIV.2411.02744. URL `https://doi.org/10.48550/arXiv.2411.02744`.

R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM*, 53(3):324–360, 2006. doi: 10.1145/1147954.1147956.

P. M. Gleiser and L. Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4):565–574, 2003.

R. Impagliazzo, R. Lei, T. Pitassi, and J. Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 818–831. ACM, 2022. doi: 10.1145/3519935.3519973.

D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146. ACM, 2003.

J. Kunegis. KONECT: The Koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web (WWW)*, pages 1343–1350, 2013.

J. Leskovec, J. M. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1):2, 2007.

C. McMeel and Y. Yoshida. Sensitivity analysis of submodular function maximization. *CoRR*, abs/2010.04281, 2020. doi: 10.48550/arXiv.2010.04281.

J. S. Naor, N. Raju, A. Shetty, A. Srinivasan, R. Valieva, and D. Wajc. Dimension-free correlated sampling for the hypersimplex. In S. Saraf, editor, *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 104:1–104:20, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi: 10.4230/LIPIcs.ITCS.2026.104. URL `https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2026.104`.

G. L. Nemhauser and L. A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978. doi: 10.1287/moor.3.3.177.

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. doi: 10.1007/BF01588971.

Y. Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

N. Varma and Y. Yoshida. Average sensitivity of graph algorithms. *CoRR*, abs/1904.03248, 2019.

A Proof of Lemma 1

Lemma. *For every instance I, the function H_I has the following properties.*

1. *H_I is concave on $[0,1]^m$.*
2. *For every $S \subseteq [m]$, $H_I(\mathbb{1}_S) = f_I(S)$.*
3. *For every $x \in [0,1]^m$, $F_I(x) \geq (1 - e^{-1})H_I(x)$.*

Proof. For each $u \in U$, the map $x \mapsto x(N_I(u))$ is linear and $t \mapsto \min\{1, t\}$ is concave on $\mathbb{R}_{\geq 0}$, so each summand of H_I is concave and therefore so is H_I.

If $x = \mathbb{1}_S$, then $x(N_I(u)) = |S \cap N_I(u)|$, so

$$\min\{1, x(N_I(u))\} = 1 \quad \Longleftrightarrow \quad S \cap N_I(u) \neq \varnothing.$$

Summing over u gives $H_I(\mathbb{1}_S) = f_I(S)$.

For the third claim, fix $u \in U$ and let $t_u := x(N_I(u))$. Then

$$1 - \prod_{i \in N_I(u)} (1 - x_i) \geq 1 - \exp\left(- \sum_{i \in N_I(u)} x_i \right) = 1 - e^{-t_u}.$$

If $t \in [0,1]$, then $1 - e^{-t} \geq (1 - e^{-1})t$, and if $t \geq 1$, then $1 - e^{-t} \geq 1 - e^{-1}$. Hence

$$1 - e^{-t_u} \geq (1 - e^{-1}) \min\{1, t_u\}.$$

Summing over u proves

$$F_I(x) \geq (1 - e^{-1}) H_I(x). \qquad \square$$

B Fractional Sensitivity Analysis

In this section we analyze the sensitivity of the fractional part of our algorithm.

Lemma 2 (Uniqueness and positivity)**.** *For every instance I and every $\lambda > 0$, the program* (1) *has a unique maximizer x_I^λ. Moreover,*

$$x_{I,i}^\lambda > 0 \qquad \textit{for every } i \in [m].$$

Proof. The feasible region $B_{k,m}$ is compact and convex. The objective is strictly concave because H_I is concave and $-\lambda R$ is strictly concave on $B_{k,m}$. Hence the maximizer is unique.

It remains to prove positivity. Suppose $x = x_I^\lambda$ has $x_i = 0$ for some i. Since $\sum_j x_j = k > 0$, there exists j with $x_j > 0$. For sufficiently small $t > 0$, the point

$$x + t(e_i - e_j)$$

still lies in $B_{k,m}$. The change in H_I is $O(t)$ because H_I is Lipschitz on the compact cube $[0,1]^m$, while

$$R(x + t(e_i - e_j)) - R(x) = t \log t + (x_j - t) \log(x_j - t) - x_j \log x_j = t \log t + O(t).$$

Therefore

$$\big(H_I - \lambda R\big)\big(x + t(e_i - e_j)\big) - \big(H_I - \lambda R\big)(x) = O(t) - \lambda t \log t > 0$$

for all sufficiently small $t > 0$, contradicting optimality. Hence every coordinate is positive. $\square$

Lemma 3 (Entropy curvature)**.** *For any $x, y \in B_{k,m}$ with positive coordinates,*

$$\langle \nabla R(x) - \nabla R(y), x - y \rangle \geq \frac{1}{k} \|x - y\|_1^2.$$

Proof. Let $p := x/k$ and $q := y/k$. Then p, q belong to the probability simplex and

$$\langle \nabla R(x) - \nabla R(y), x - y \rangle = \sum_{i=1}^m (x_i - y_i)(\log x_i - \log y_i) = k \sum_{i=1}^m (p_i - q_i)(\log p_i - \log q_i).$$

The last quantity is the Jeffreys divergence

$$k\big(D_{\mathrm{KL}}(p \| q) + D_{\mathrm{KL}}(q \| p)\big),$$

which is at least $k\|p - q\|_1^2$ by Pinsker's inequality. Since $\|p - q\|_1 = \|x - y\|_1 / k$, the claim follows. $\square$

Lemma 4. *Let $I \sim I'$ be adjacent, and fix $x \in [0,1]^m$. For every $g' \in \partial H_{I'}(x)$ there exists $g \in \partial H_I(x)$ such that*

$$\|g - g'\|_\infty \leq 1.$$

Proof. Let $e \in U$ be the unique universe element whose incidence pattern may differ between I and I'. For each $u \in U$, the summand

$$h_u^I(x) := \min\{1, x(N_I(u))\}$$

has superdifferential

$$\partial h_u^I(x) = \left\{ a\, \mathbb{1}_{N_I(u)} : a \in \begin{cases} \{1\}, & x(N_I(u)) < 1, \\ [0,1], & x(N_I(u)) = 1, \\ \{0\}, & x(N_I(u)) > 1. \end{cases} \right\}.$$

By the finite-sum rule for concave superdifferentials,

$$\partial H_I(x) = \sum_{u \in U} \partial h_u^I(x) := \left\{ \sum_{u \in U} \gamma_u : \gamma_u \in \partial h_u^I(x) \text{ for every } u \in U \right\}.$$

In particular, every supergradient of H_I is the sum of one such contribution for each u. Choose a decomposition

$$g' = \sum_{u \in U} \gamma'_u \qquad \text{with} \qquad \gamma'_u \in \partial h_u^{I'}(x) \quad \text{for every } u \in U.$$

For every unchanged element $u \neq e$, we have $N_I(u) = N_{I'}(u)$ and hence

$$\partial h_u^I(x) = \partial h_u^{I'}(x).$$

We therefore set $\gamma_u := \gamma'_u$ for all $u \neq e$. For the changed element e, choose any

$$\gamma_e \in \partial h_e^I(x),$$

and define

$$g := \sum_{u \in U} \gamma_u.$$

Then $g \in \partial H_I(x)$ and

$$g - g' = \gamma_e - \gamma'_e.$$

Each coordinate of γ_e and γ'_e lies in $[0,1]$, so each coordinate of $\gamma_e - \gamma'_e$ lies in $[-1,1]$. Therefore

$$\|g - g'\|_\infty = \|\gamma_e - \gamma'_e\|_\infty \leq 1.$$

□

C Implementation Details

In this section we provide details about the implementation and deployment of the algorithm.

C.1 The Fractional Stage

The fractional stage maximizes $f(x) = H_I(x) - \lambda R(x)$ subject to $x \in B_{k,m}$. Since f is a non-smooth concave function and $B_{k,m}$ is convex, standard convex optimization methods apply. For instance, the *projected subgradient method* [Nesterov et al., 2018] guarantees a $O(1/\sqrt{T})$ convergence rate. It proceeds as follows:

1. Initialize $x^0 \in B_{k,m}$ and let $\eta_t = \Theta(1/\sqrt{t})$.
2. For $t = 0, 1, 2, ..., T$, let $g^t \in \partial f(x^t)$ and update:

$$y^{t+1} = x^t + \eta_t\, g^t$$
$$x^{t+1} = \Pi_{B_{k,m}}(y^{t+1}) = \arg\min_{x \in B_{k,m}} \|x - y^{t+1}\|^2$$

We return $x^* = x^{t^*}$ and $t^* = \arg\max_{t \leq T} f(x^t)$.

Each iteration requires computing $g^t \in \partial f(x^t)$ and taking the ℓ_2-projection back onto $B_{k,m}$. The former requires computing the superdifferential $\partial f(x^t)$ and can be done in time $O(m + E)$ where E is the total number of element-to-set inclusions. The latter is a well-studied problem with efficient algorithms that can run in $O(m)$ time [Duchi et al., 2008].

In our implementation, this convex optimization problem is formulated using CVXPY and solved via the SCS solver, followed by a Euclidean projection step to correct for minor numerical solver tolerances and ensure the solution lies exactly on the hypersimplex.

C.1.1 Subdifferential Calculation

Let $v_u \in \{0,1\}^m$ be the indicator vector for the set $N_I(u)$, such that $(v_u)_i = 1$ if $i \in N_I(u)$ and 0 otherwise. We can write the non-smooth term as $h_u(x) = \min\{1, v_u^\top x\}$.

Because $h_u(x)$ is concave, we compute its superdifferential $\partial h_u(x)$:

$$\partial h_u(x) = \begin{cases} \{v_u\} & \text{if } v_u^\top x < 1 \\ \{\mathbf{0}\} & \text{if } v_u^\top x > 1 \\ \{\alpha v_u \mid \alpha \in [0,1]\} & \text{if } v_u^\top x = 1 \end{cases}$$

For the negative-entropy regularizer $R(x) = \sum_{i=1}^m x_i \log x_i$, the gradient with respect to coordinate i is:

$$\nabla R(x)_i = 1 + \log x_i$$

The full objective is $f(x) = \sum_u h_u(x) - \lambda R(x)$. A supergradient $g \in \partial f(x)$ has components given by:

$$g_i = \sum_{u: i \in N_I(u)} (\tilde{g}_u)_i - \lambda(1 + \log x_i)$$

where $\tilde{g}_u \in \partial h_u(x)$ is chosen according to the piecewise definition above.

C.2 The Rounding Stage

The rounding stage requires us to implement the hypersimplex correlated sampling algorithm from [Naor et al., 2026]. By Theorem 3.13 of [Naor et al., 2026], the $O(\log k)$-stretch hypersimplex correlated sampling algorithm for $\Delta_{n,k}$ takes time $O(nnz \cdot \log^*(m))$ to round a point $x \in \Delta_{k,m}$, where $nnz := |\{i \in [m] : x_i \neq 0\}|$. Since $nnz \leq m$, this step also takes linear time.

The implementation first partitions the fractional variables into "small" and "large" items based on a randomized threshold. Small items are uniformly hashed into buckets, and each bucket is compressed into a single representative item using an exponential clock mechanism (Algorithm 4 from [Naor et al., 2026]). This compression step reduces the dimensionality of the problem to $O(k^5 \log^5 n)$. The compressed vector is then rounded using the Coordinate Tree Rounding algorithm of Chen et al. [2017]. To ensure the process runs in input-sparsity time, our implementation evaluates the coordinate tree bottom-up, lazily generating the required uniform random variables θ_z and completely bypassing zero-valued nodes.

C.3 Experimental Setup

Our algorithms are implemented in Python. We ran the experiments on a machine with 16GB RAM and an M2 CPU.

Facility Location We construct a geometric maximum coverage instance in $[0,1]^2$ with K well-separated clusters. Around each cluster center c_a we place C nearly co-located "twin" facilities

$$f_{a,j} = c_a + \rho_f(\cos(2\pi j/C), \sin(2\pi j/C)), \qquad j = 0, \ldots, C-1,$$

with $\rho_f \ll r$, where r is the coverage radius. Each cluster also contains many core demand points near c_a, which are covered by several twins and therefore make the twins nearly interchangeable. To create instability, we add boundary demand points

$$b_{a,j} = c_a + \rho_b(\cos(2\pi j/C), \sin(2\pi j/C)),$$

with ρ_b chosen so that $b_{a,j}$ is covered by the jth twin but not by most other twins. An adjacent instance is obtained by moving one boundary demand from direction j to direction $j+1$, thereby shifting a small marginal advantage from one twin to a neighboring twin.

Influence Maximization We use the standard live-edge model. For each sampled live-edge world, the universe consists of target nodes and the candidate sets are seed nodes: seed v covers target u iff u is reachable from v in the live-edge graph. An adjacent instance may change the incidence pattern of a single target node, while leaving all other target rows unchanged. To estimate worst-case

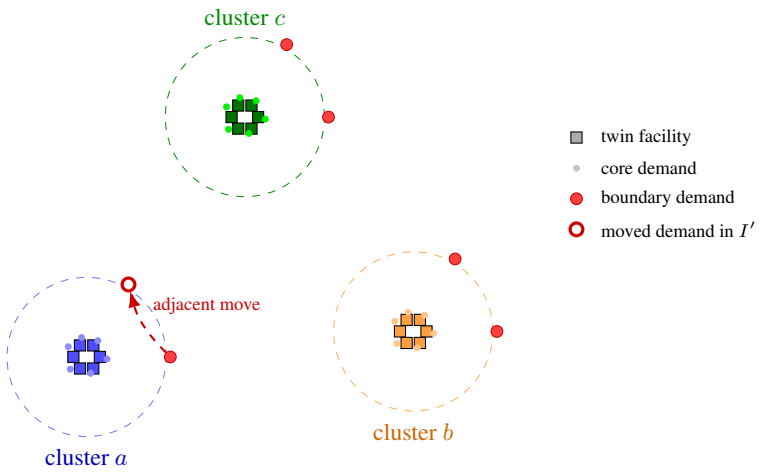

Figure 3: Schematic of the facility-twins construction.

sensitivity, for each live-edge world we generate candidate replacements for several target rows, including deletion and all-seed reachability. We report the maximum sensitivity averaged across live-edge worlds.

For our real-world experiments, we evaluate our algorithm on the *Jazz* network ($n = 198$), a collaboration network of jazz musicians [Gleiser and Danon, 2003]; *Email-Eu-Core* ($n = 1{,}005$), an email communication network from a European research institution [Leskovec et al., 2007]; and *Hamsterster* ($n = 2{,}426$), an online social network [Kunegis, 2013].

Random Inclusion : Fix $|U| = 500$, $m = 200$, $k = 8$ and $p = 0.05$. Each universe element is included with probability p to each set. We randomly sample a collection of neighboring instance pairs and record the maximum sensitivity over those pairs for each algorithm.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: We provide an algorithm for stable maximum coverage and analyze it.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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 analyze the sensitivity–utility tradeoff.

Guidelines:

- The answer [N/A] 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: Our analysis is found in the main body and appendix.

   Guidelines:

   - The answer [N/A] 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: All our experiments and setup are outlined in detail.

   Guidelines:

   - The answer [N/A] 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: [No]

   Justification: Our codebase is available upon request.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are outlined in the Appendix.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: We report error bars by averaging over multiple seeds.

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: This information is provided in the Appendix.

   Guidelines:

   - The answer [N/A] 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:

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: Our paper is mostly theoretical in nature and we do not foresee immediate societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification:

Guidelines:

- The answer [N/A] 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: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.