From Theory to Practice

Efficient Influence Maximization in Modern Social Networks

COMP 3801: Algorithms for Modern Data Sets

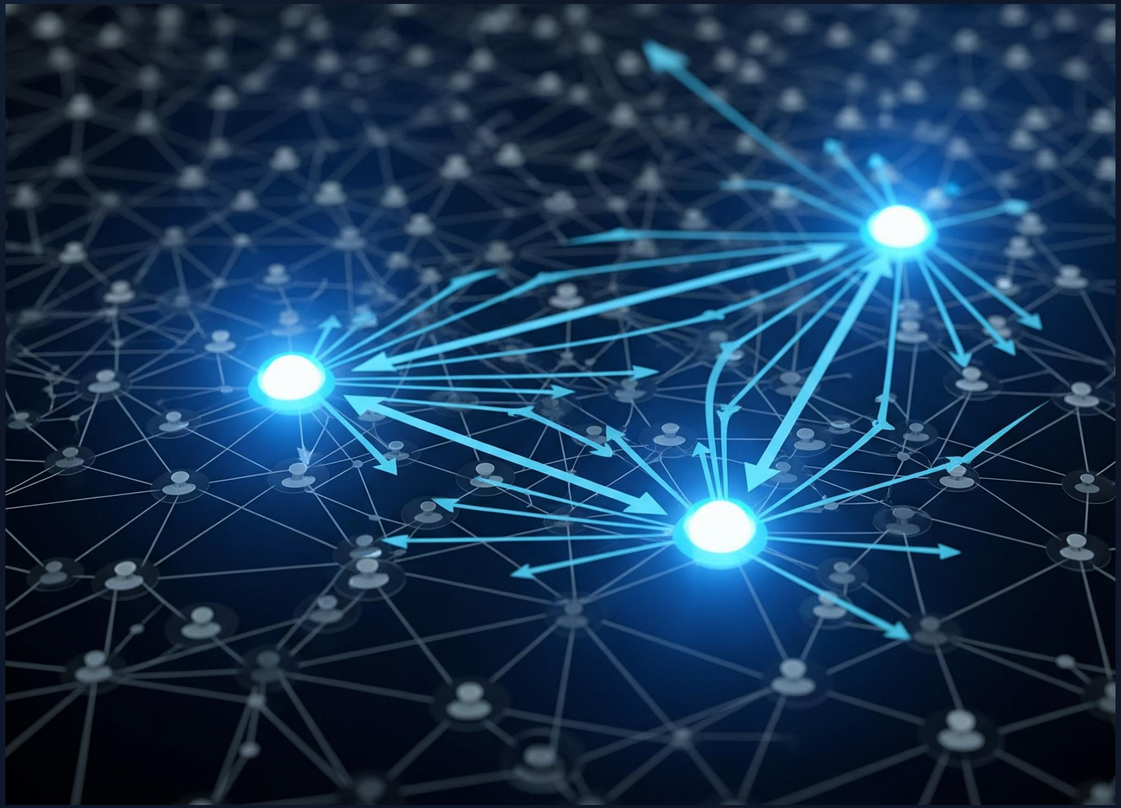
The Core Problem

The $1,000,000 Question

If you have a limited budget (e.g., 50 free samples)...

Who do you target to start the biggest "word-of-mouth" cascade?

- Viral Marketing
- Public Health Campaigns
- Stopping Misinformation



Formalizing the Problem

The Goal

Find a seed set S of size k that maximizes the influence σ(S).

$$S^* = \arg\max S \subseteq V, |S|=k \; \sigma(S)$$

The Bad News

This problem is

NP-Hard

We can't find the **perfect** solution efficiently. We must find a "good enough" approximation.

How Does Influence **Actually** Spread?

Independent Cascade (IC)

The "Coin Flip" Model

When a node becomes active, it gets **one chance** to "flip a coin" and activate each neighbor.

e.g., $p = 0.01$

Linear Threshold (LT)

The "Peer Pressure" Model

A node becomes active only when the **sum of influence** from its active friends passes a random threshold.

e.g., $\Sigma\, wi \geq \theta v$

"

The Key Insight (KKT, 2003)

$\sigma(S)$ is Submodular

"

This is just a formal name for...

"Diminishing Returns"

$\sigma(S \cup \{v\}) - \sigma(S) \geq \sigma(T \cup \{v\}) - \sigma(T)$ for $S \subseteq T$

What is Submodularity?

Diminishing Returns

The marginal gain of adding a node to a **small set** is greater than (or equal to) adding it to a **large set**.

Adding a seed to an empty set gives a HUGE gain. Adding the *same* seed to a set that already covers its neighbors gives a tiny gain.

Algorithm 1: The "Guaranteed" Greedy Algorithm

The Guarantee: Because σ(S) is submodular, a simple "hill-climbing" algorithm is provably good.

$$\sigma(S_{greedy}) \geq (1 - 1 / e) \cdot \sigma(S_{optimal}) \approx 0.63$$

The Algorithm: In each of k rounds, pick the one node that gives the biggest **additional** (marginal) gain.

The Problem: SLOW. To find the best node, you must run thousands of Monte Carlo simulations for **every** other node.

Complexity: $O(k \cdot n \cdot R \cdot (m + n))$ — "Could take days to complete."

Algorithm 2: The CELF Optimization (Chen et al. 2009)

The Goal: Get the **exact same 63% answer** as the Greedy algorithm, but do it fast.

The "Lazy" Idea: Uses submodularity again. A node's gain can **only go down**.

The Method:

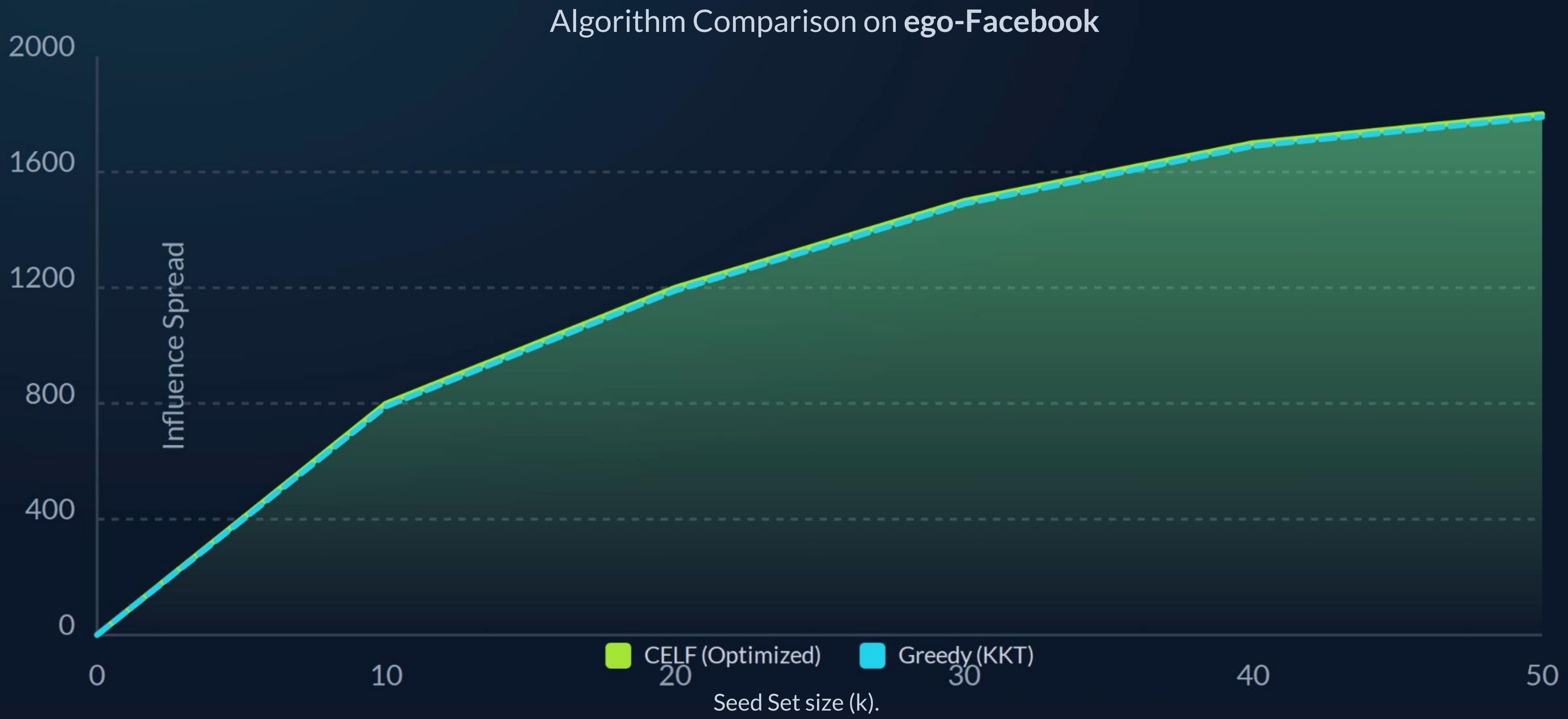
1. Calculate all ***n*** node gains once. Store them in a Priority Queue.

2. Pop the best node ***v***. Re-calculate its "true" (smaller) gain.

3 . Check: Is ***v***'s new gain still bigger than the **old** gain of the #2 node? I f yes, ***v*** is the winner. We just skipped ***n-1*** calculations.

The Result: Reported **up to 700x speedup**.

Experiment 1: Quality Check (Greedy vs. CELF)



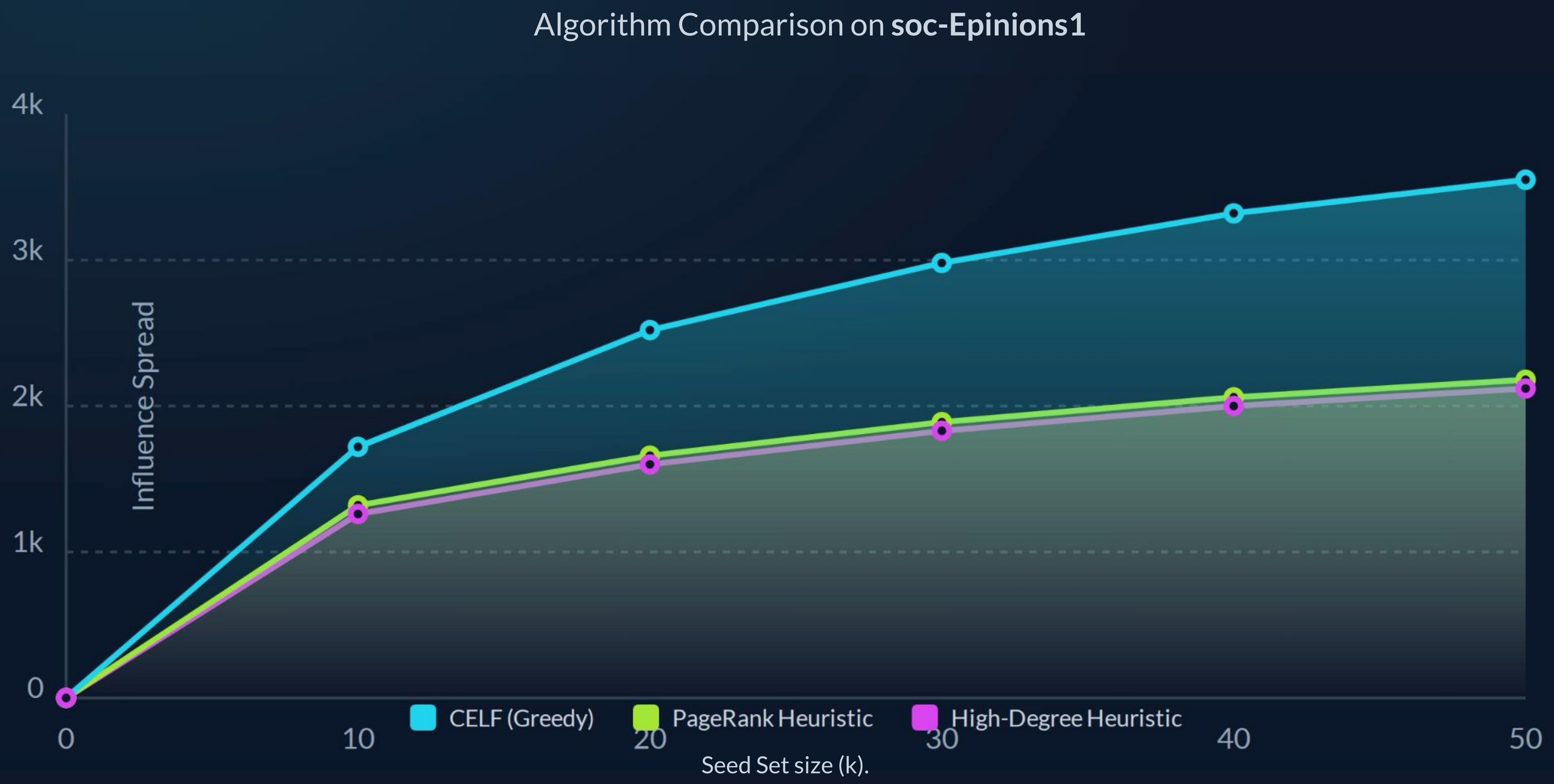
Result: The algorithms produce identical results. CELF is a true optimization.

Experiment 2: Performance (Is CELF **Fast**?)

Total Runtime to Select 50 Seeds on **ego-Facebook** (Log Scale)

*Result: CELF is **~74x faster**, achieving the same 63% guarantee in a fraction of the time.*

Experiment 3: Quality (Is CELF **Smart**?)



Conclusion: Why Did CELF Win?

The "Clustering" Problem

Simple heuristics (like "High-Degree" or "PageRank") make a basic mistake:

They pick popular nodes that are all in the **same community**.

Their influence **overlaps**, wasting the budget.

The Submodular Solution

CELF (Greedy) is "smarter." It optimizes for **marginal gain**.

It naturally finds a **diverse set of seeds** from different clusters to "cover" the network more effectively.

Conclusion & Future Work

- **Conclusion 1: Submodularity** is the key theoretical property that makes this problem solvable.
- **Conclusion 2:** This property gives the Greedy algorithm a **provable 63% optimal guarantee**.
- **Conclusion 3:** The **CELF optimization** uses this **same property** to make the algorithm 10s or 100s of times faster.
- **Future Work:**
  - Scalability: Implement near-linear time algorithms (like IMM) for billion-node graphs.
  - Extensions: Explore competitive, topic-aware, or dynamic network models.

Paper (Authors, Year)	Core Problem Addressed	Simple Summary	Key Concepts
KKT (Kempe, Kleinberg, Tardos, 2003)	The fundamental question: Is influence maximization even solvable?	They proved it's **NP-Hard**, but showed that because the spread function has the property of **submodularity** ('diminishing returns'), a simple **greedy algorithm** is guaranteed to give a 63% approximation.	**Submodularity, Greedy Algorithm, (1-1/e) Guarantee.**
Chen et al. (Chen, Wang, Yang, 2009)	The problem with KKT: The greedy algorithm is too slow to run in practice.	They introduced the **CELF** optimization. It uses the same submodularity principle but adds a clever **'lazy' check** using a priority queue to skip thousands of unnecessary simulations, making the algorithm practically feasible.	**CELF Optimization, Efficiency, Priority Queue.**
Tang et al. (Tang, Shi, Xiao, 2015)	The problem with KKT/CELF: Even CELF is too slow for networks with billions of edges.	They moved away from costly direct simulation entirely. They introduced a mathematically complex method using **Reverse Reachable (RR) Sets** and martingale bounds to *estimate* influence in near-linear time, O((k+l)(n+m)).	**Near-Linear Time, Reverse Reachable (RR) Sets, Scalability.**

Thank You

Questions?