Guest lecture II: Amos Fiat's Social Networks class

Edith Cohen

TAU, December 2014

Today

Diffusion of information/contagion in networks:
Applications:

- Influence queries
- Influence maximization
- Influence similarity

Reachability-based diffusion:

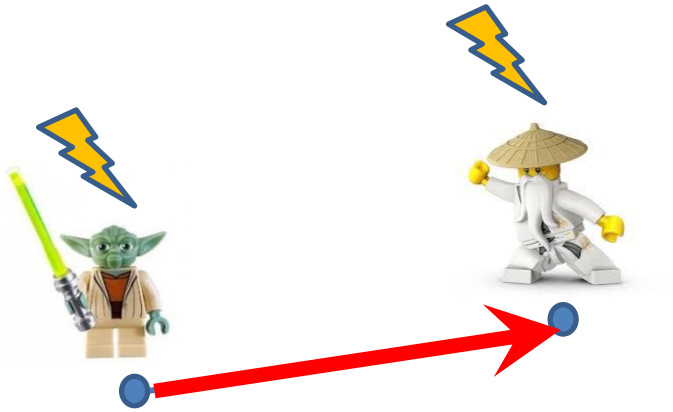
Models & Scalable computation

- Basic reachability
- IC model
- Set of instances

Diffusion in Networks

Contagion, information, news, opinions, …

spread over the network. When two nodes are connected, infection can pass from one to the other.



Diffusion in Networks

Model of how information/infection spreads

Applications:

- **Influence queries** $\mathrm{Inf}(S)$: The expected benefit/risk of recruiting/infecting the *seed* set S
- **Influence maximization**: With a given budget s, who should we recruit ? (viral marketing) $\arg\max_{|S|=s} \mathrm{Inf}(S)$
- **Influence similarity**: $J(u, v)$: similarity of u, v in terms of "correlation" of their influence sets

Challenges

- **Modeling**: Formulate a model that captures what we want
- **Scalability**: Very efficient computation of many queries on very large networks

Modeling Diffusion

Intuitions we may want our model to capture:

- Influence extends centrality from one node to multiple nodes
- The marginal influence of adding another seed node u to S is at most $\text{Inf}(u)$ (***submodularity)***

$$\text{Inf}(\boldsymbol{S_1} \cup \boldsymbol{S_2}) \leq \text{Inf}(\boldsymbol{S_1}) + \text{Inf}(\boldsymbol{S_2})$$

- Influence can only increase if we add nodes to S (***monotonicity)***

$$\text{Inf}(\boldsymbol{S_1} \cup \boldsymbol{S_2}) \geq \max\{\text{Inf}(S_1), \text{Inf}(S_2)\}$$

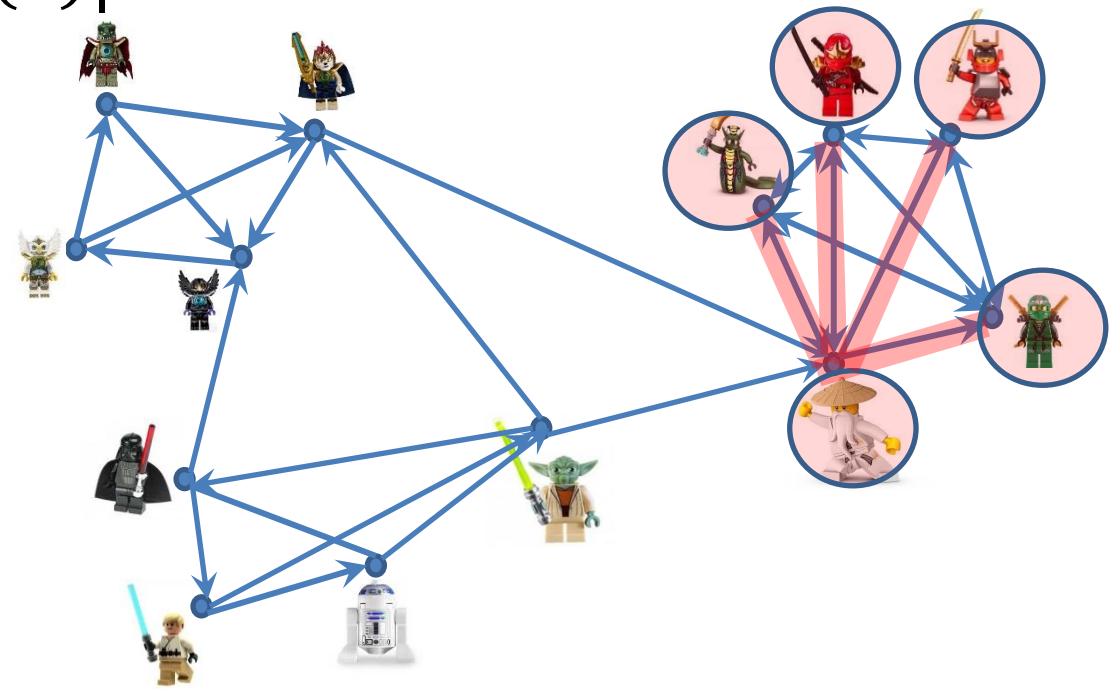
Simplest Model: Reachability

"You infect everyone you can reach"

For a seed set S of nodes: $R(S) = \{u | \exists v \in S, v \rightsquigarrow u\}$ are the nodes reachable from at least one node in S.

Influence$(S) = |R(S)|$.

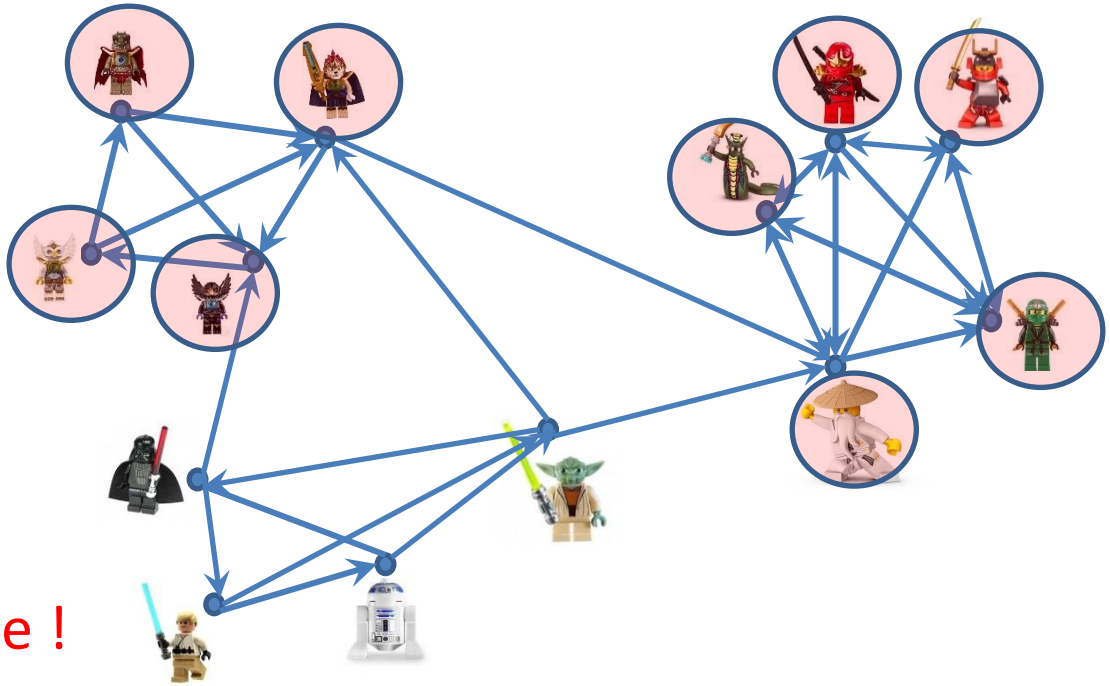
$Inf($  $) = 5$



Simplest Model: Reachability

"You infect everyone you can reach"

$Inf($   $) = 9$



Submodular and monotone !

Scalability: Node sketches

If we compute a MinHash sketch of $R(v)$ for each node v, we can efficiently estimate answers for

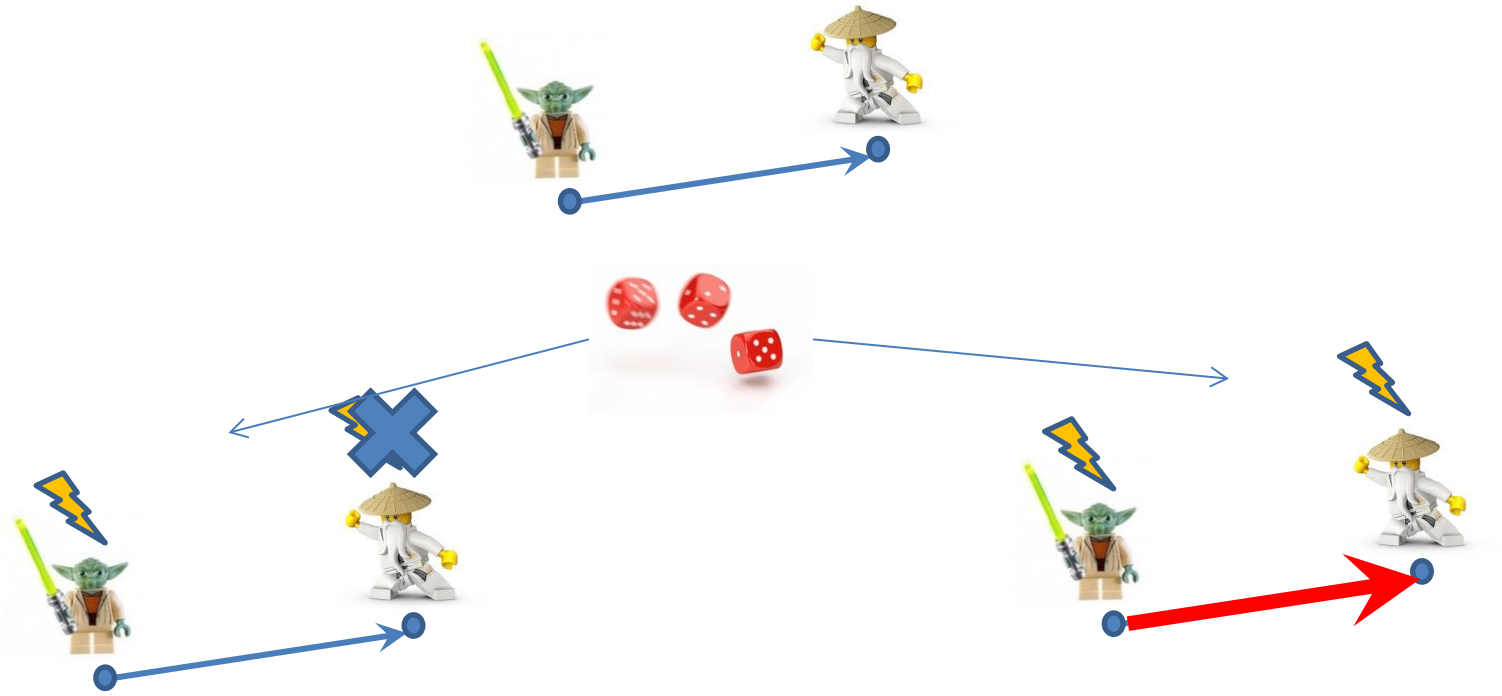
- Influence queries: For a set S of one or more seed nodes, estimate $Inf(S) = |\cup_{v \in S} R(v)|$ with a small relative error
- Jaccard similarity of "influence sets" of two nodes $J(u,v) = \frac{|R(v) \cap R(u)|}{|R(v) \cup R(u)|}$
- More queries supported by MinHash sketches.

Reachability diffusion model: Issues

"You infect everyone you can reach"

Reachability does not capture:

- Intuition that ***contagion is probabilistic*** in nature

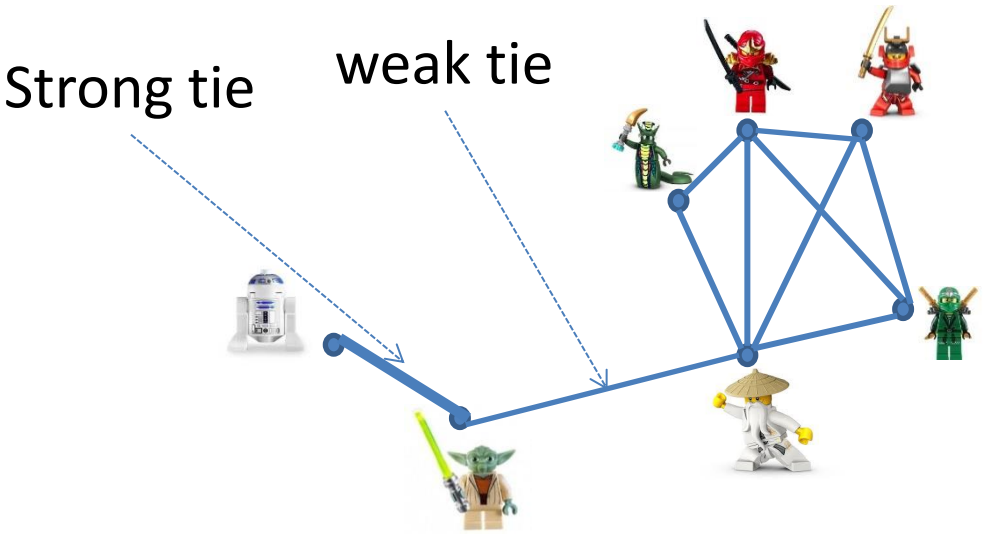


Reachability diffusion model: Issues

"You infect everyone you can reach"

Reachability does not capture:

- **Asymmetry**: Distinguish strong or weak connections. Even if network is undirected, influence is not (may depend, say, on how many friends you have)

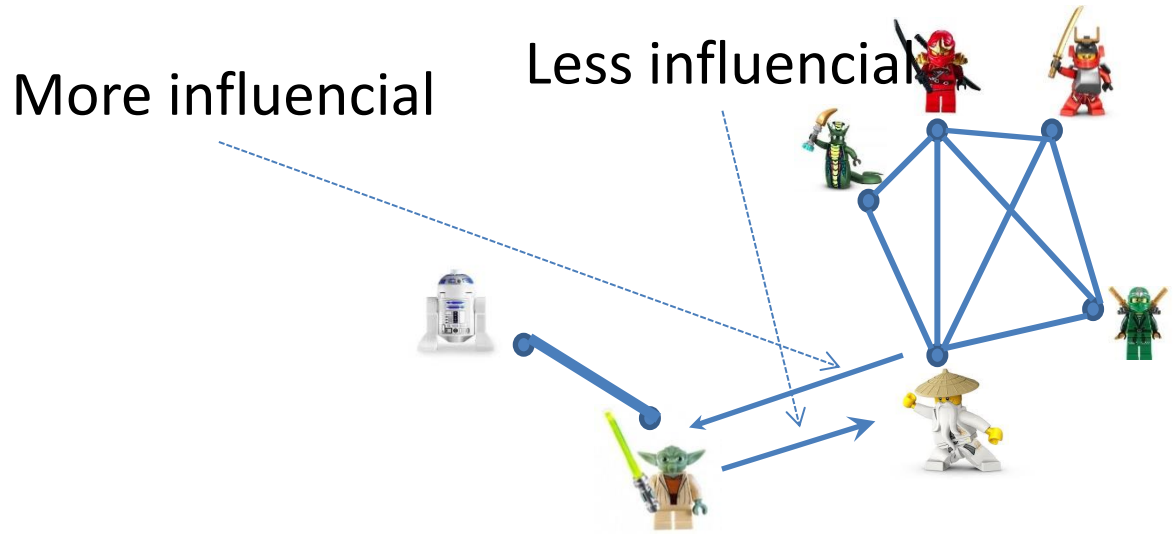


Reachability diffusion model: Issues

"You infect everyone you can reach"

Reachability does not capture:

- **Asymmetry**: Distinguish strong or weak connections. Even if network is undirected, influence is not (may depend, say, on how many friends you have)

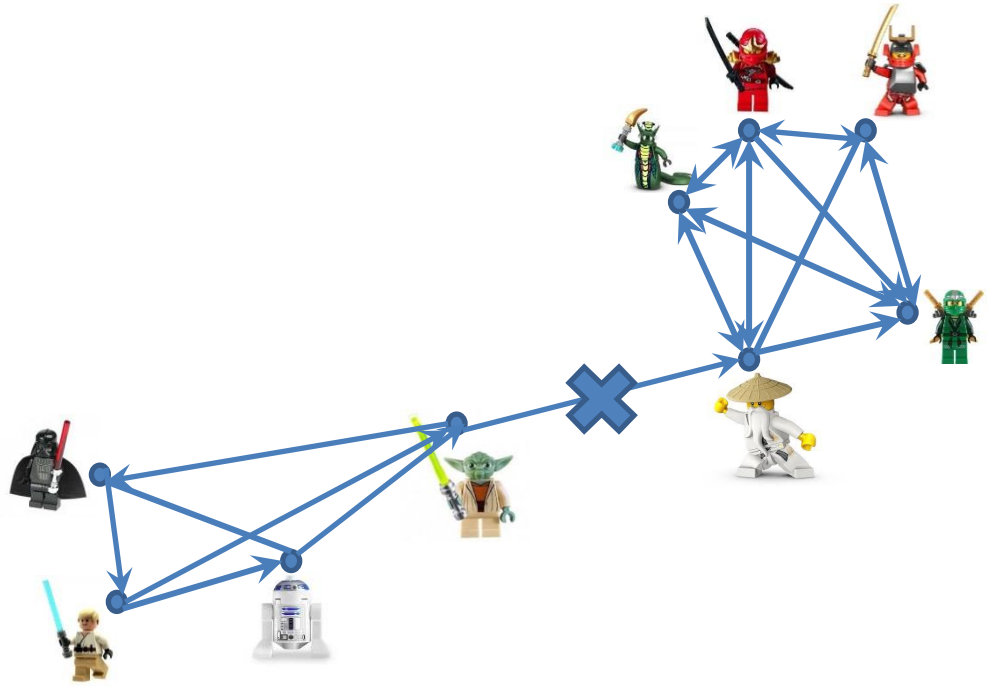


Reachability diffusion model: Issues

"You infect everyone you can reach"

Reachability does not capture:

- **Not robust**: Can be very sensitive to presence or deletions of one or few (weak) edges

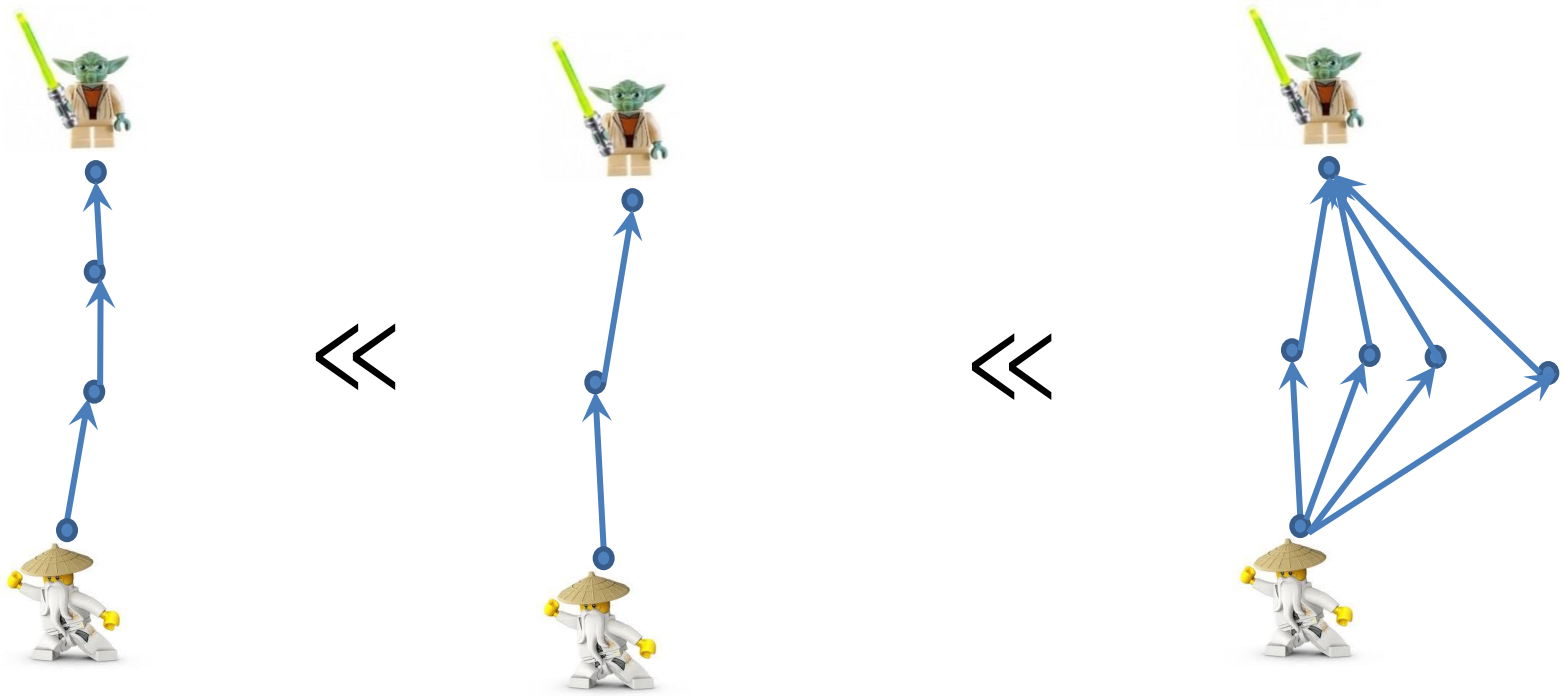


Reachability diffusion model: Issues

"You infect everyone you can reach"

Reachability does not capture:

- Infection probability should decrease with path length and increase with path multiplicity.



Reachability diffusion model: Issues

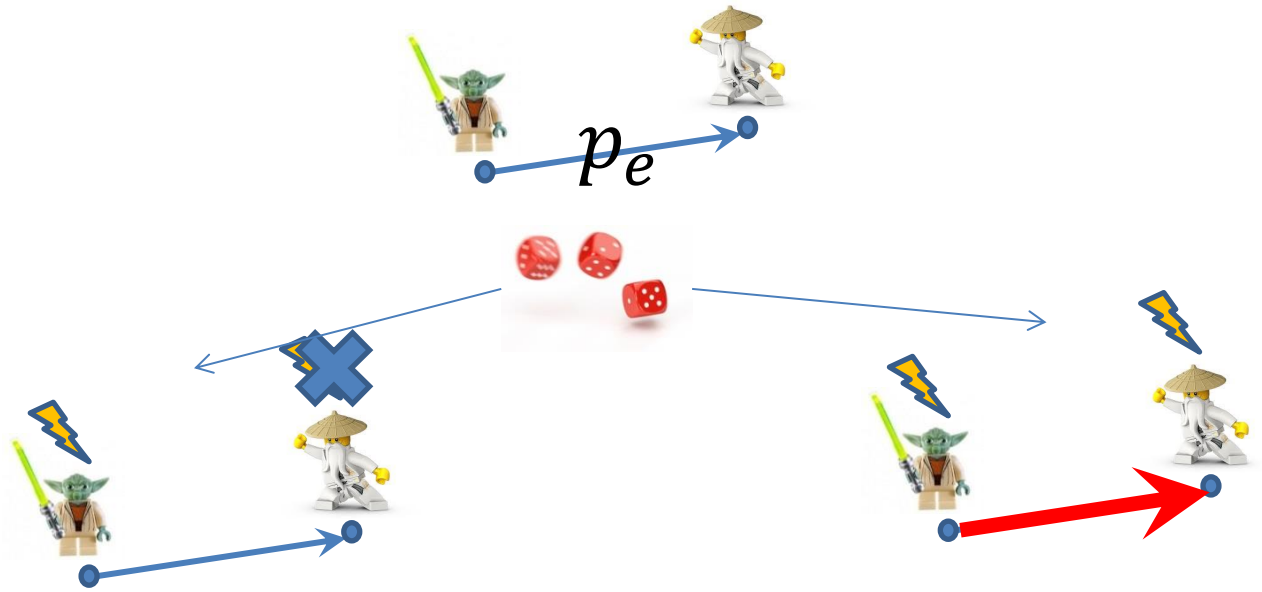
"You infect everyone you can reach"

Reachability does not capture:

- Intuition that *contagion is probabilistic* in nature
- Asymmetry: Distinguish strong or weak connections. Even if network is undirected, influence is not (may depend, say, on how many friends you have)
- Not robust: Can be very sensitive to presence or deletions of one or few (weak) edges
- Infection probability should decrease with path length and increase with path multiplicity.

Independent Cascade (IC) diffusion model [Kempe, Kleinberg, Tardos 2003]

- Each (directed) edge e has an independent probability p_e to be active
- Influence of S is the expected number of reachable nodes



Independent Cascade (IC)

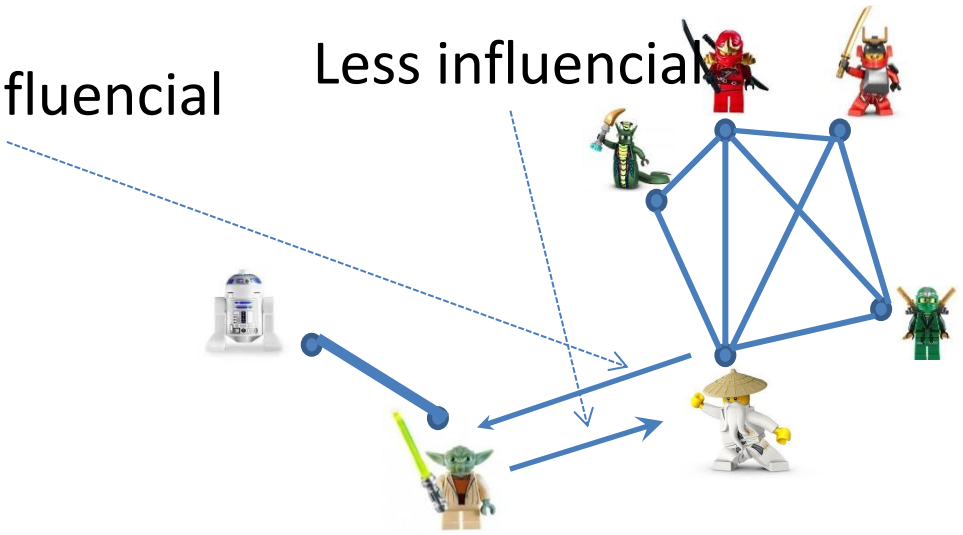
IC model does capture:

- ✓ Intuition that *contagion is probabilistic* in nature
- ✓ Asymmetry: Distinguish strong or weak connections. Even if network is undirected, influence is not (may depend, say, on how many friends you have)
- ✓ Not robust: Can be very sensitive to presence or deletions of one or few (weak) edges
- ✓ Infection probability should decrease with path length and increase with path multiplicity.

Independent Cascade (IC)

✓ Asymmetry: Distinguish strong or weak connections. Even if network is undirected, influence is not (may depend, say, on how many friends you have)

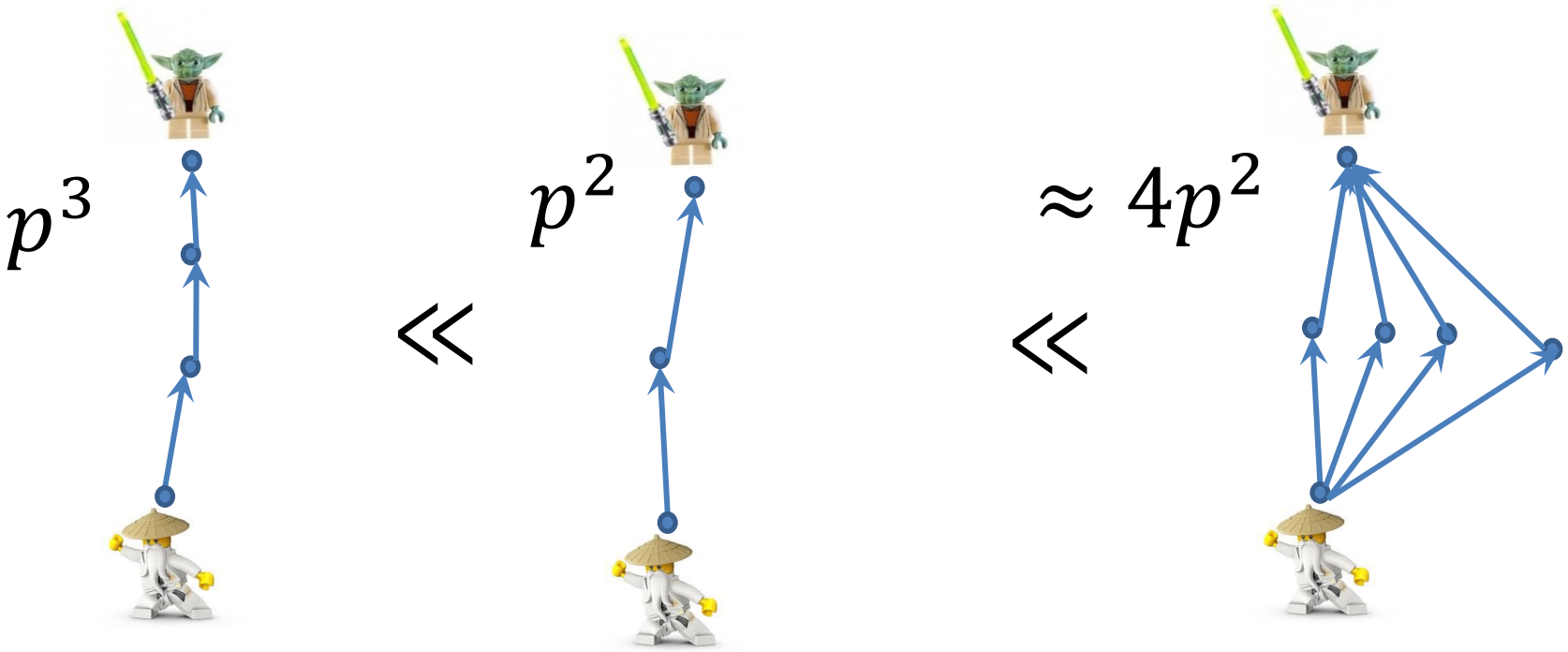
Can model stronger ties with higher $p(u, v)$



Can do $p(u, v) \propto 1/\deg(v)$

Independent Cascade (IC)

✓ Infection probability should decrease with path length and increase with path multiplicity.



Scalability: Sketches for an IC model

To work with an IC model:

We would like to compute a MinHash sketch of the "influence set" of each node. This would allow us to answer efficiently influence queries $Inf(S)$ and similarity queries $J(u, v)$.

- What are the sets that we sketch ?

Sketches for a fixed set of instances

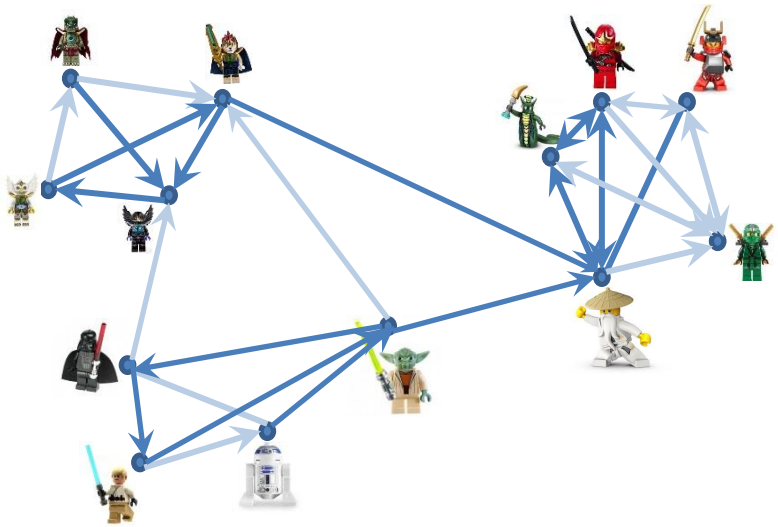
- We first consider a fixed (arbitrary) set of *instances* (edge sets): Influence is the average (or sum) of reachable set sizes, over instances.

Motivation:

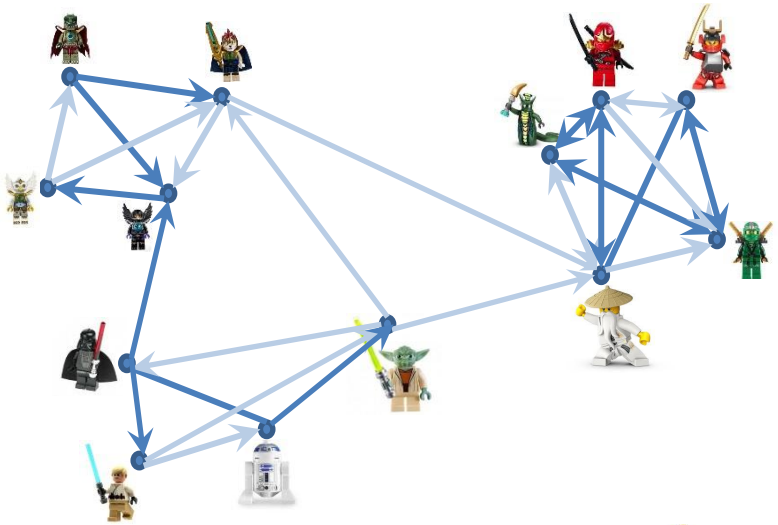
- When the instances come from "enough" Monte Carlo simulations of an IC model, the sketches capture the model.
- Capture "median" behavior of IC model
- Can capture relations beyond IC model (edges not independent)

Fixed set of instances

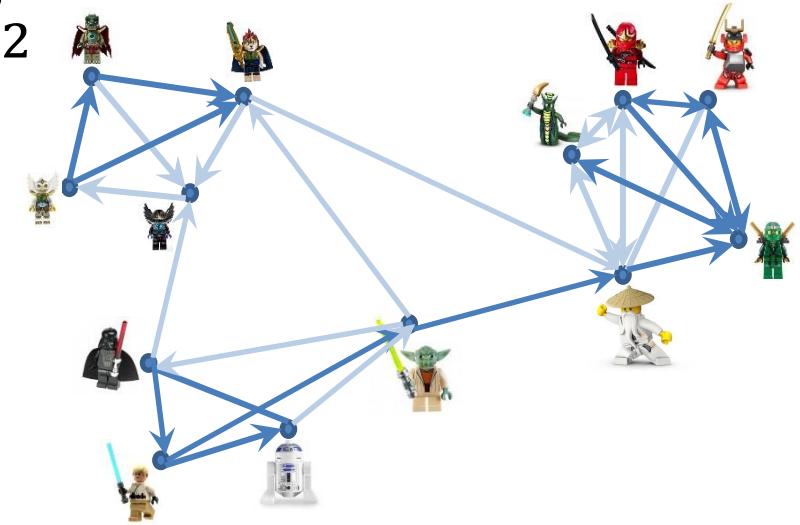
E_1



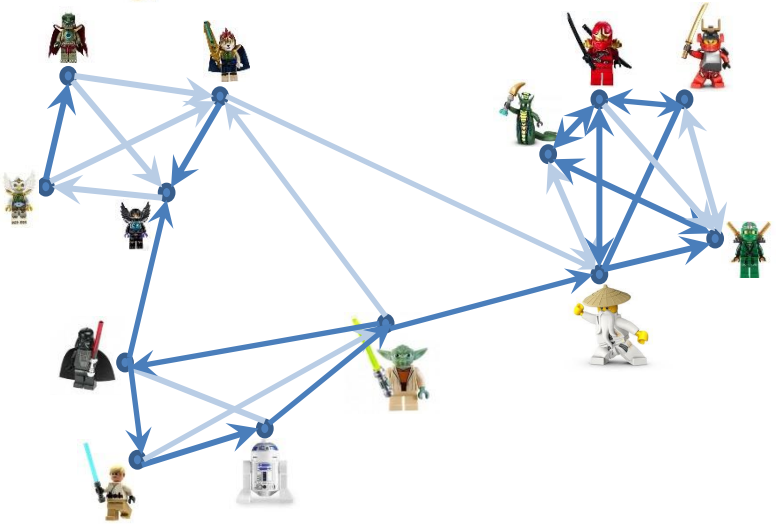
E_3



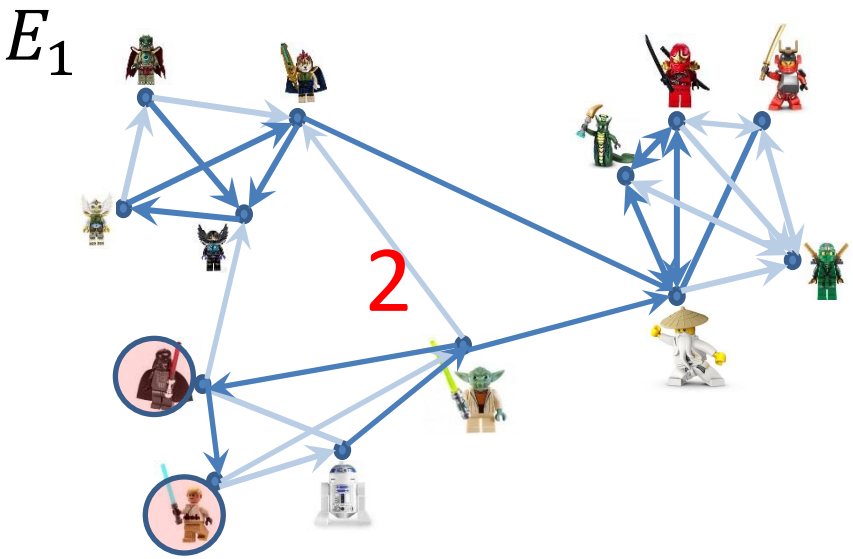
E_2

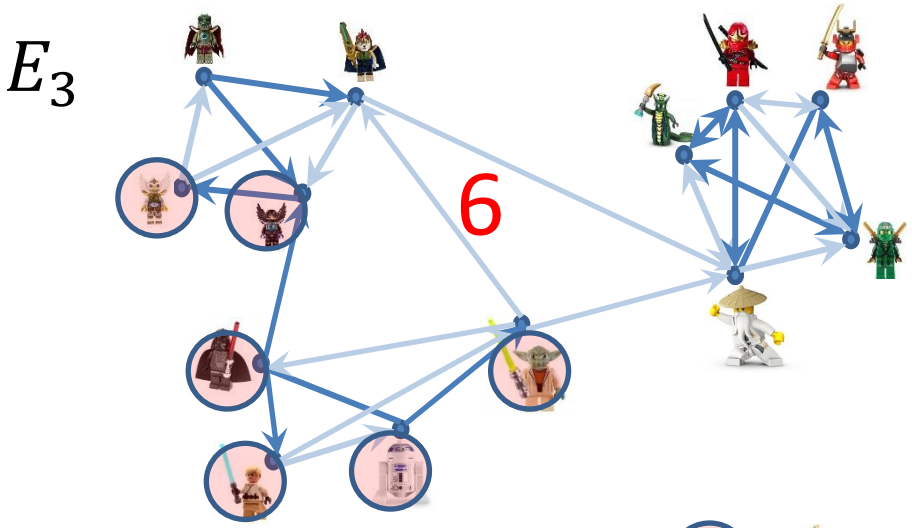


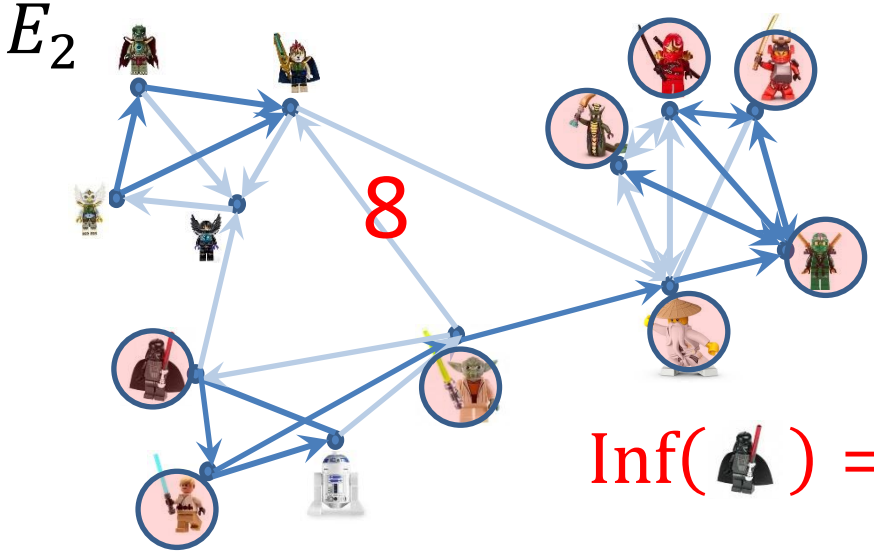
E_4

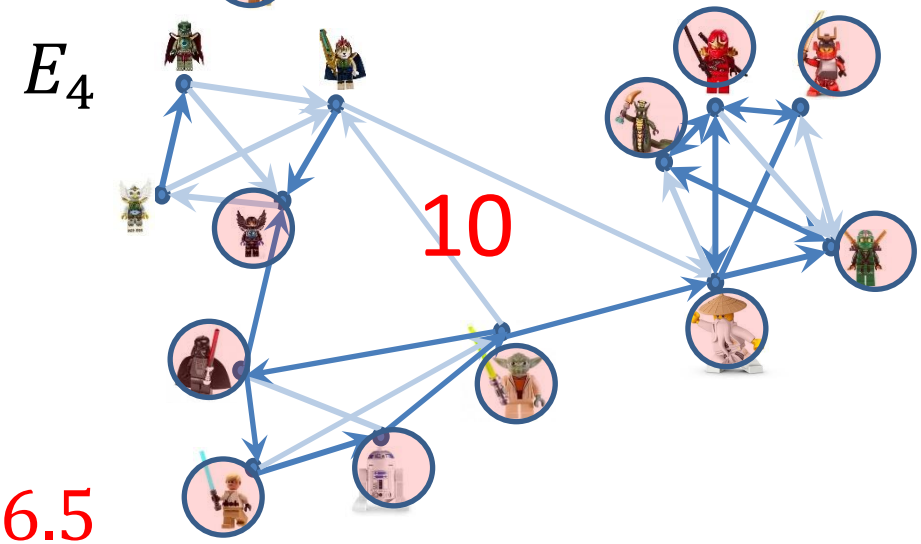


Fixed set of instances: Inf( )









Inf( ) = 6.5

Sketches for a fixed set of instances

Approach I [CWY KDD 2009]

- Compute a set of Reachability MinHash sketches for each instance. Keep and work with all sets.
- For a query $\mathrm{Inf}(S)$: Estimate from sketches the reachability of S in each instance and then average.

But with ℓ instances, we need, ℓk storage per node!

Sketches for a set of ℓ instances
Better Approach [CDPW 2014]: Combined reachability sets

- Elements are (node,instance) pairs.
- The *combined reachability set* of v:

$$R_v = \{(u, i) | v \rightsquigarrow u \text{ in } E_i\}$$

- $\text{Inf}(S) = \frac{|\cup_{v \in S} R_v|}{\ell}$
- $J(u, v) = \frac{|R_u \cap R_v|}{|R_u \cup R_v|}$

Combined reachability sketches

We compute MinHash sketches for the combined reachability sets: $R_v = \{(u, i) | v \rightsquigarrow u \; in \; E_i\}$

- Each node-instance pair gets a rank $\sim U[0,1]$. The bottom-k sketch includes the k smallest ranks of pairs in R_v
- We can sketch R_v by first computing a set of sketches in each instance, and then computing the union sketch over instances (k smallest hash values across)

- Computation grows linearly with the number of instances
- Sketch size is $O(k)$

Sketches for an IC model

- Simulate working with infinite number of instances. $O(nk)$ instances are always enough
- Estimation accuracy of influence and similarity is with respect to the expectation in the IC model
- Computation of IC sketches can be expensive

Open problem: Can we **compute** IC model sketches more efficiently ?

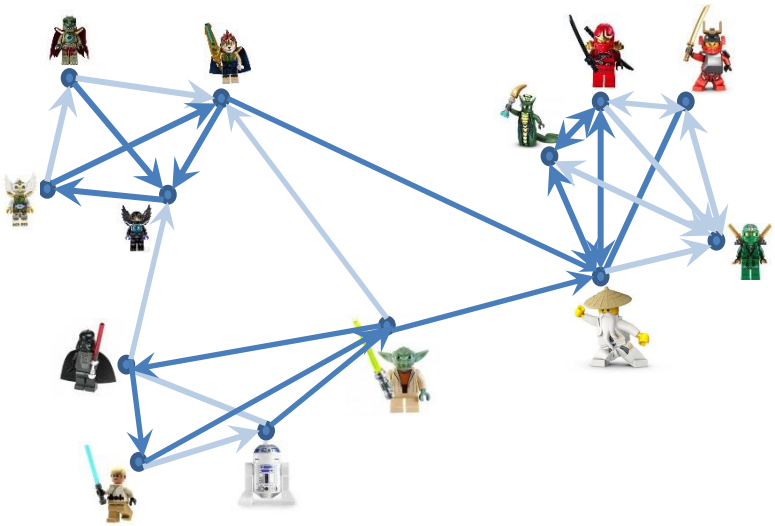
IC model sketching

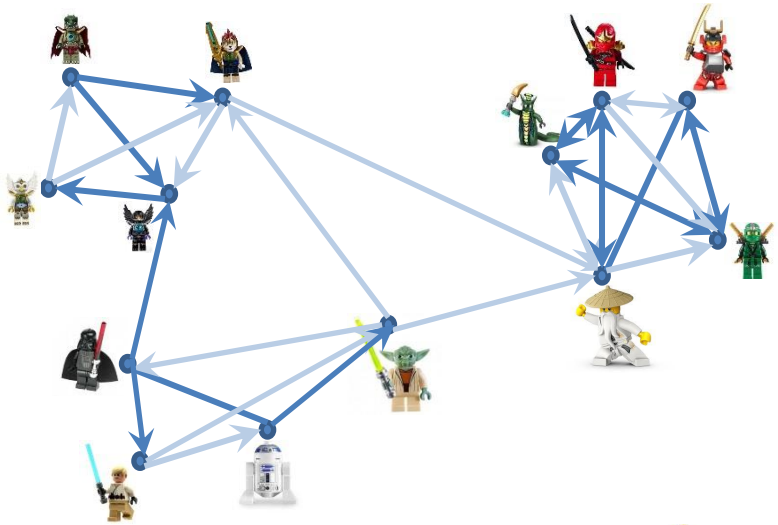
$j \leftarrow 0$

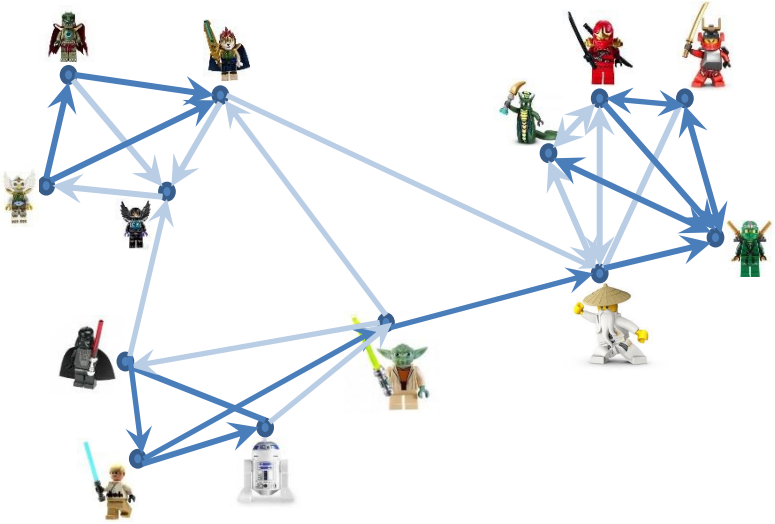
Repeat until $\forall v, |S(v)| = k$

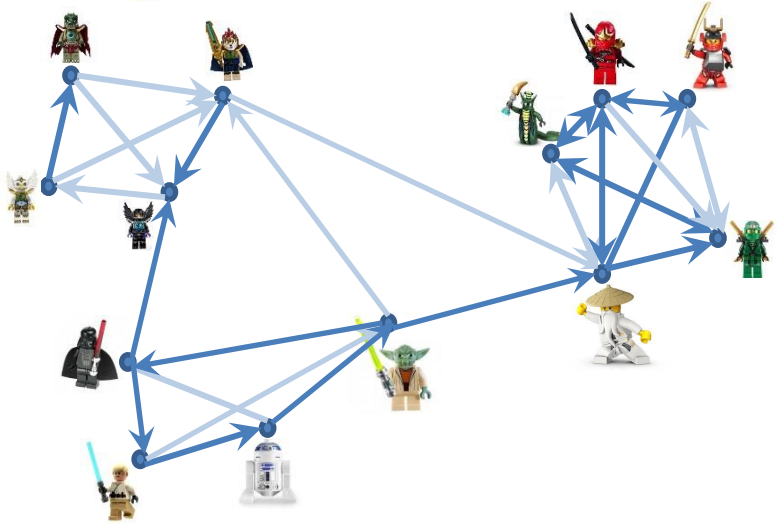
- j++; Select a node uniformly at random.
- Perform a reverse search from the node, instantiating edges along the way.
- $\forall$ visited u with $|S(u)| < k$, add j to $S(u)$

- A node selected k times always has a full sketch.
- kn iterations suffice: Can stack k random permutations of the n nodes









Influence Maximization

$$\arg\max_{|S|=s} \mathrm{Inf}(S)$$

For a given s, find a set of seed nodes S of size s that has maximum influence

We can consider influence maximization:

- On a single instance ("static" graph)
- A set of instances
- An IC model

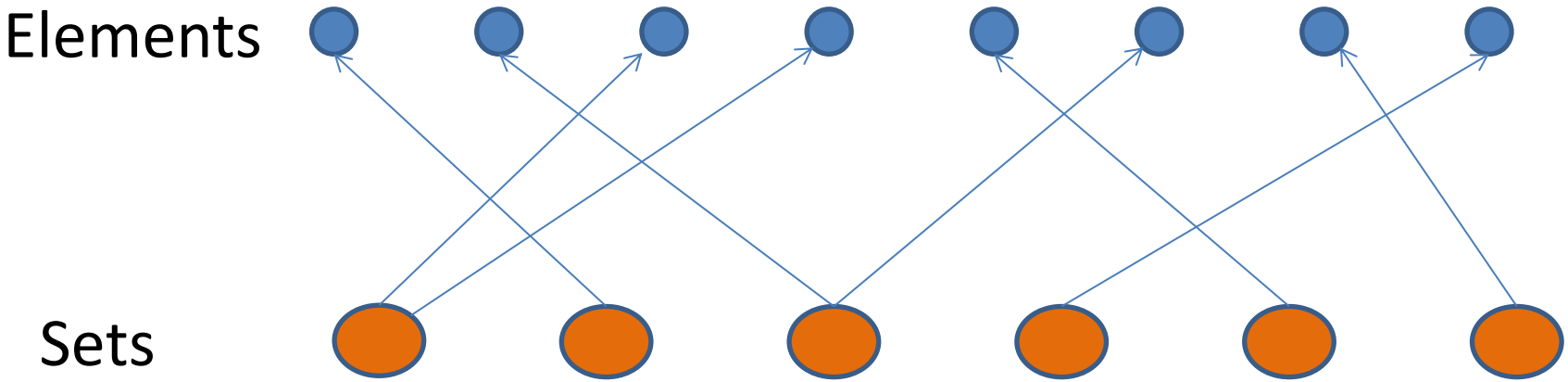
Single instance captures the basic scalability challenges

Influence Maximization

$$\arg\max_{|S|=s} \mathrm{Inf}(S)$$

- Bad news: Problem is NP-hard even for a single instance (one "static" graph)

Reduction to max/set cover:



Arc (u, v) $\Leftrightarrow$ element v is in set u

Influence Maximization

$$\arg\max_{|S|=s} \mathrm{Inf}(S)$$

- Good news: Monotone and submodular

The greedy algorithm gives approximation ratio:

$\geq 1 - \left(1 - \frac{1}{s}\right)^s > 1 - \frac{1}{e}$ of opt [NWF '78]

- Practice: Greedy is extensively used in very many applications.
- Theory: Approximation ratio is the best we can hope for in $\ll n^s$ time [Feige '98]

Greedy Influence Maximization

Initialize: $S \leftarrow \emptyset$

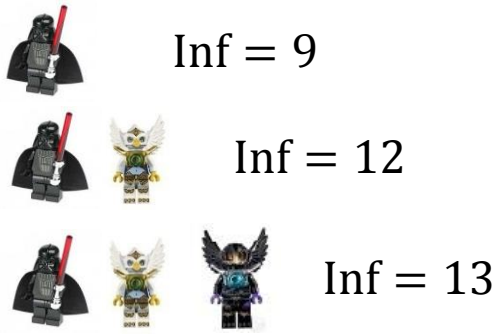
Repeat:

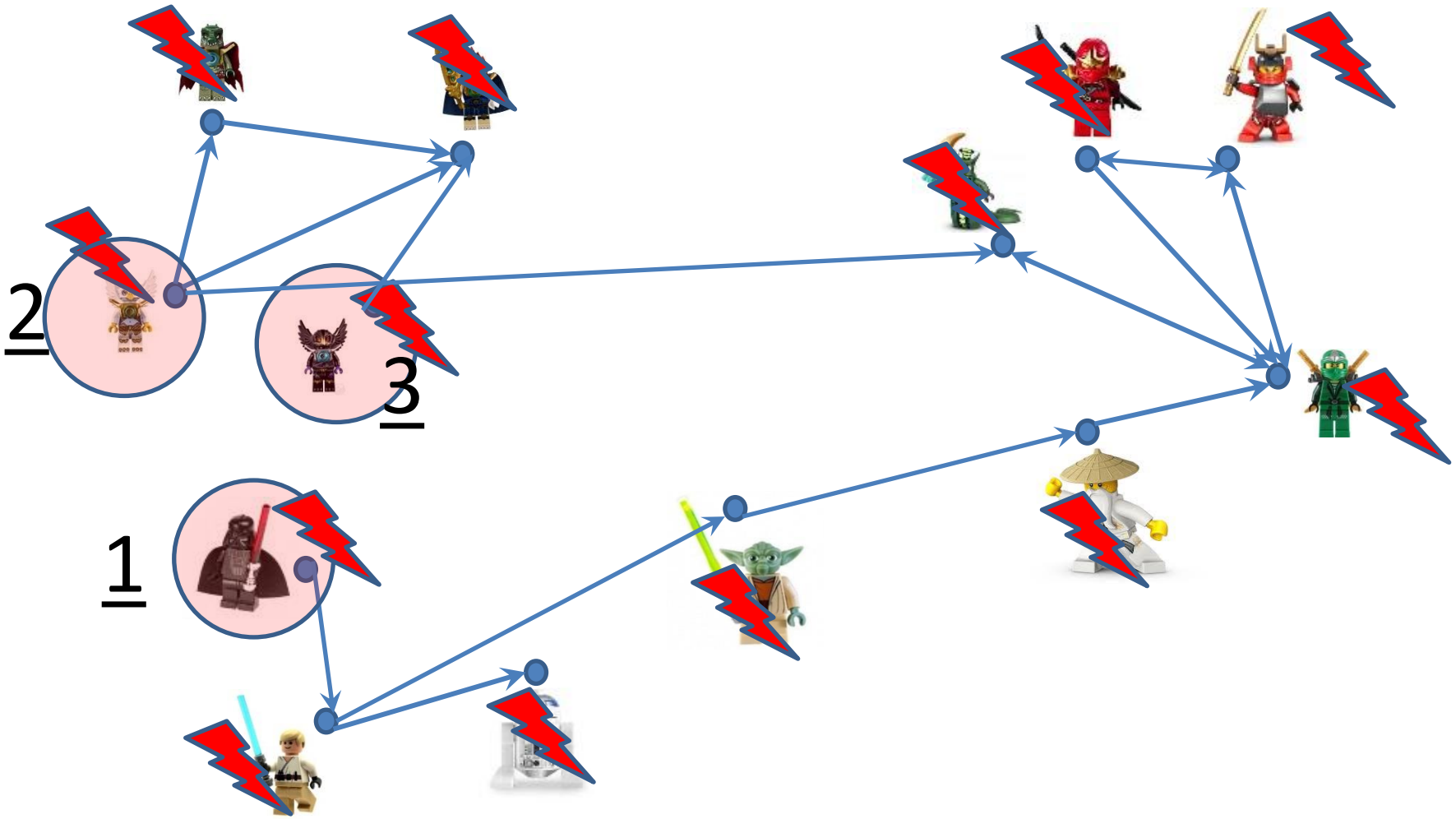
- $u \leftarrow \arg\max_v \mathrm{Inf}(S \cup \{v\})$
- $S \leftarrow S \cup \{u\}$

Until $|S| = s$

- Greedy generates a sequence of nodes
- The approximation guarantee is for each prefix

Greedy Sequence





Scalability

Greedy does not scale well even on a single "static" graph – We can not afford much more than linear time on very large networks

- In each step we need to determine the node with maximum marginal gain.
- Exact computation of the cardinality for each node is costly (search from each node)

Scalability

Settle for *approximate maximum* in each step!

Relative error affects approximation ratio only by a little: $\geq 1 - \left(1 - \frac{1}{s}\right)^s - O(\epsilon)$ times Opt

- We can use reachability sketches to determine the *approximate* maximum in each step.
- But… still $O(\#edges)$ per step.

SkIM: Sketch Based Influence Maximization [CDPW CIKM 2014]

SkIM Iteration:

- Compute "sketches" but only to the point of determining the node u with (approximate) maximum influence.
- Update a residual problem which has selected and covered nodes removed. Other nodes have partial sketches that include entries due to remaining nodes.

We show SkIM for one instance (similar for multiple instances)

SkIM: Sketch Based Influence Maximization [CDPW CIKM 2014]

SkIM Iteration (detailed): Use $k = O(\epsilon^{-2} \log \mathrm{n})$

Sketch building:

Select new node v uniformly, do a reverse search from v .

For each visited node u:

- Increment $sk[u]$
- $L_v \leftarrow L_v \cup \{u\}$
- If $sk[u] == k$: break; select u

If all nodes processed: $u \leftarrow \arg\max_u sk[u]$

Residual problem update after selecting u:

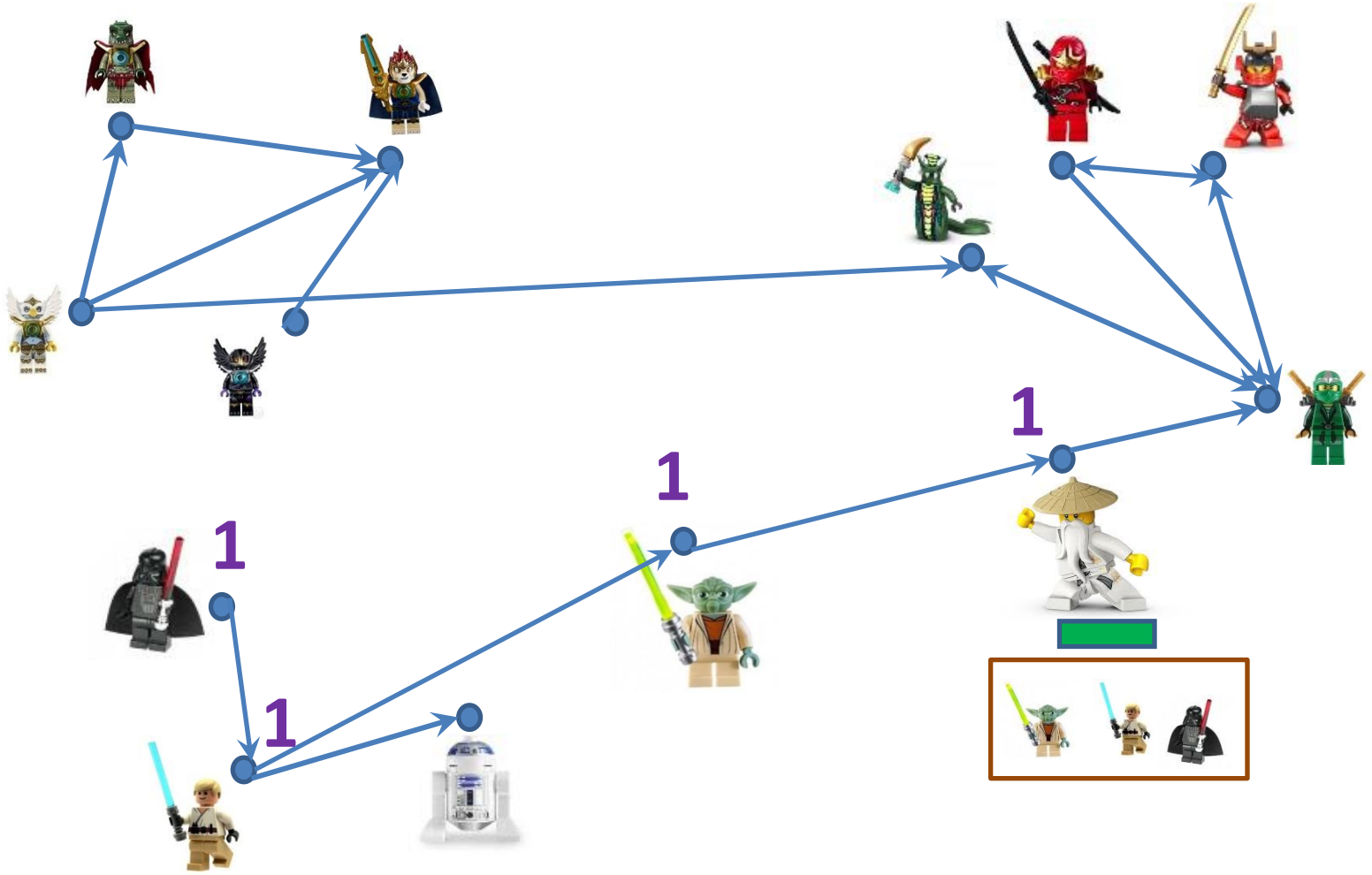
- Do a forward search from u.
- Remove all reachable edges and nodes z.
- For all $u \in L_z$, decrement $sk[u]$

SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

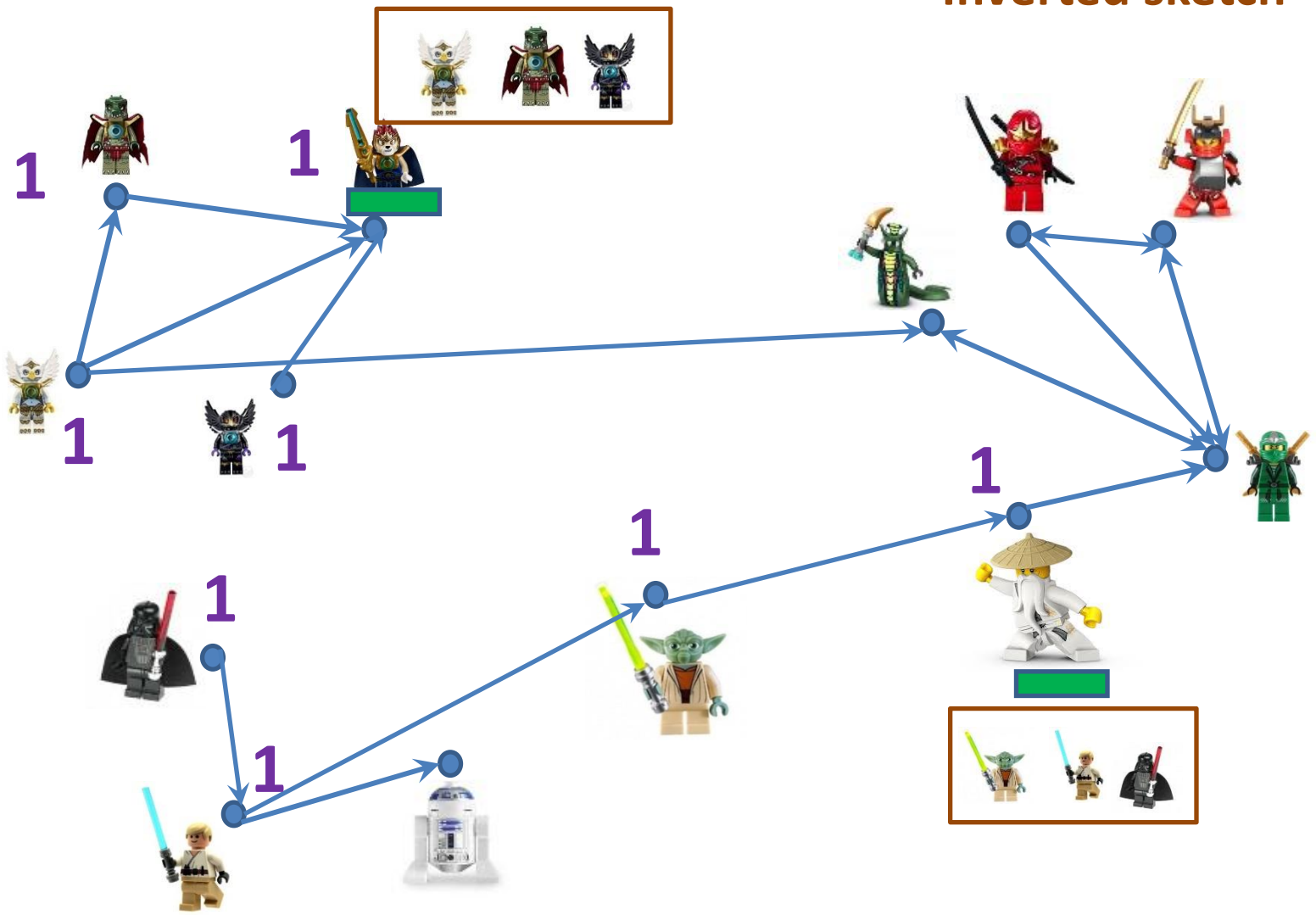


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

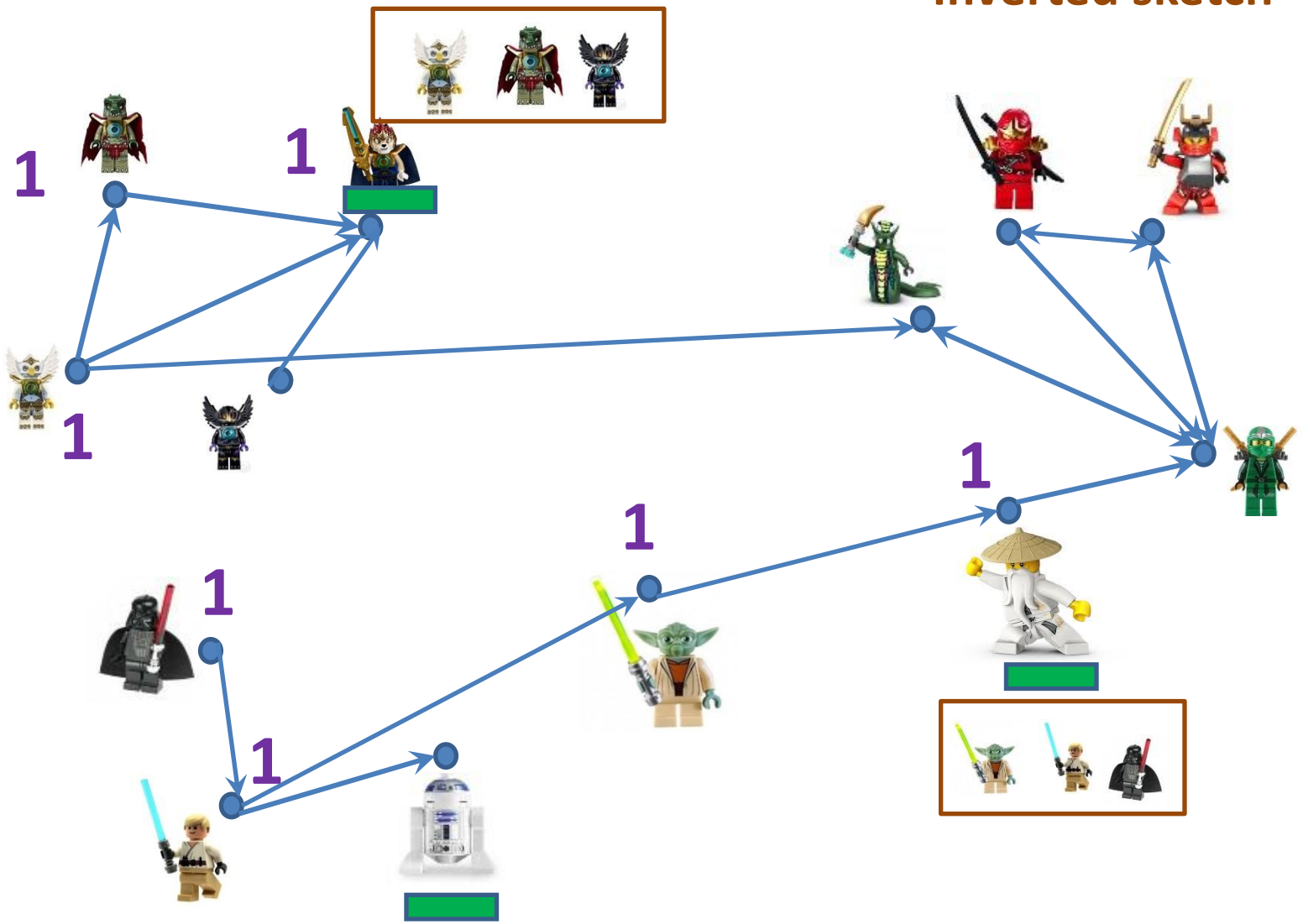


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

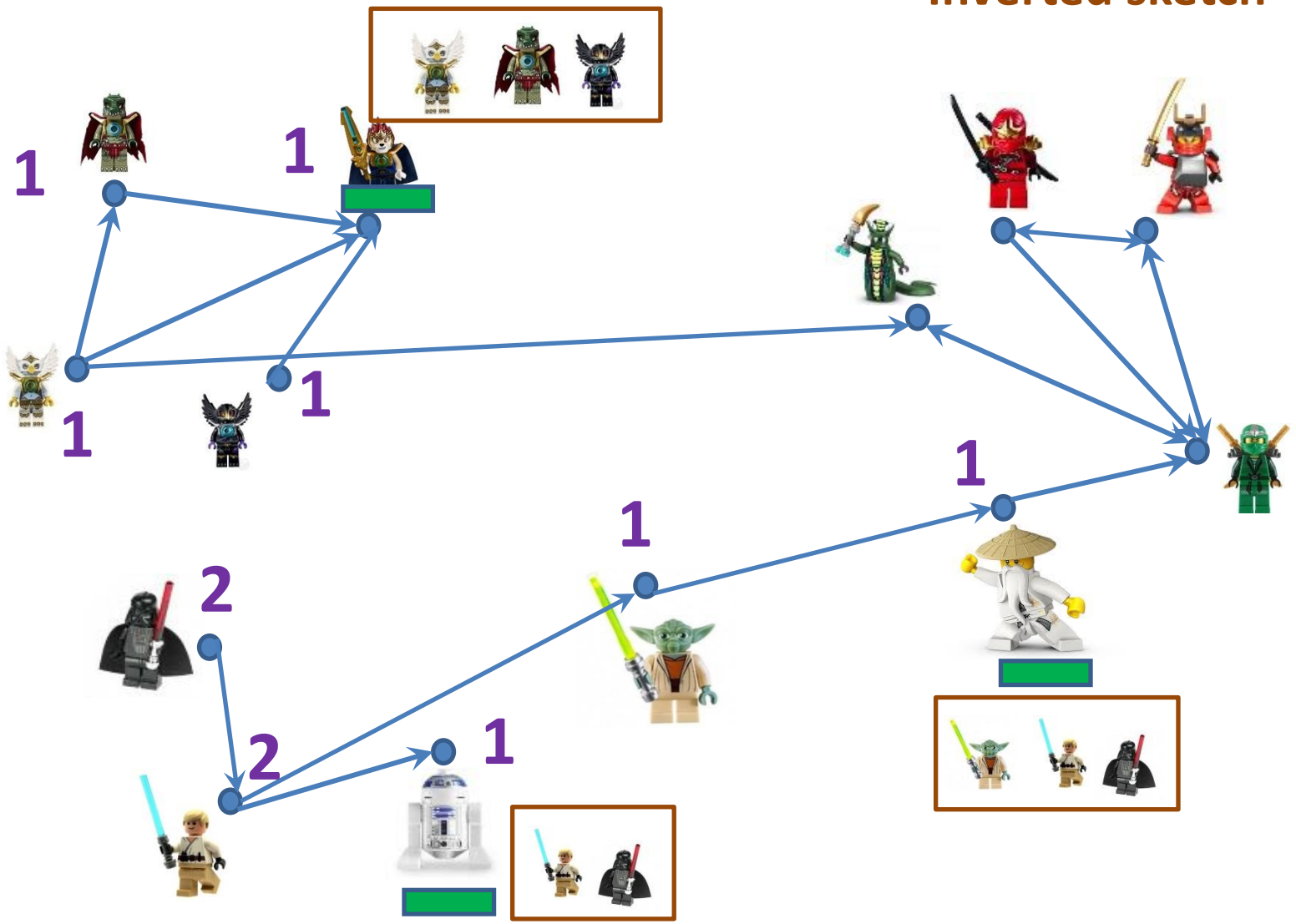


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

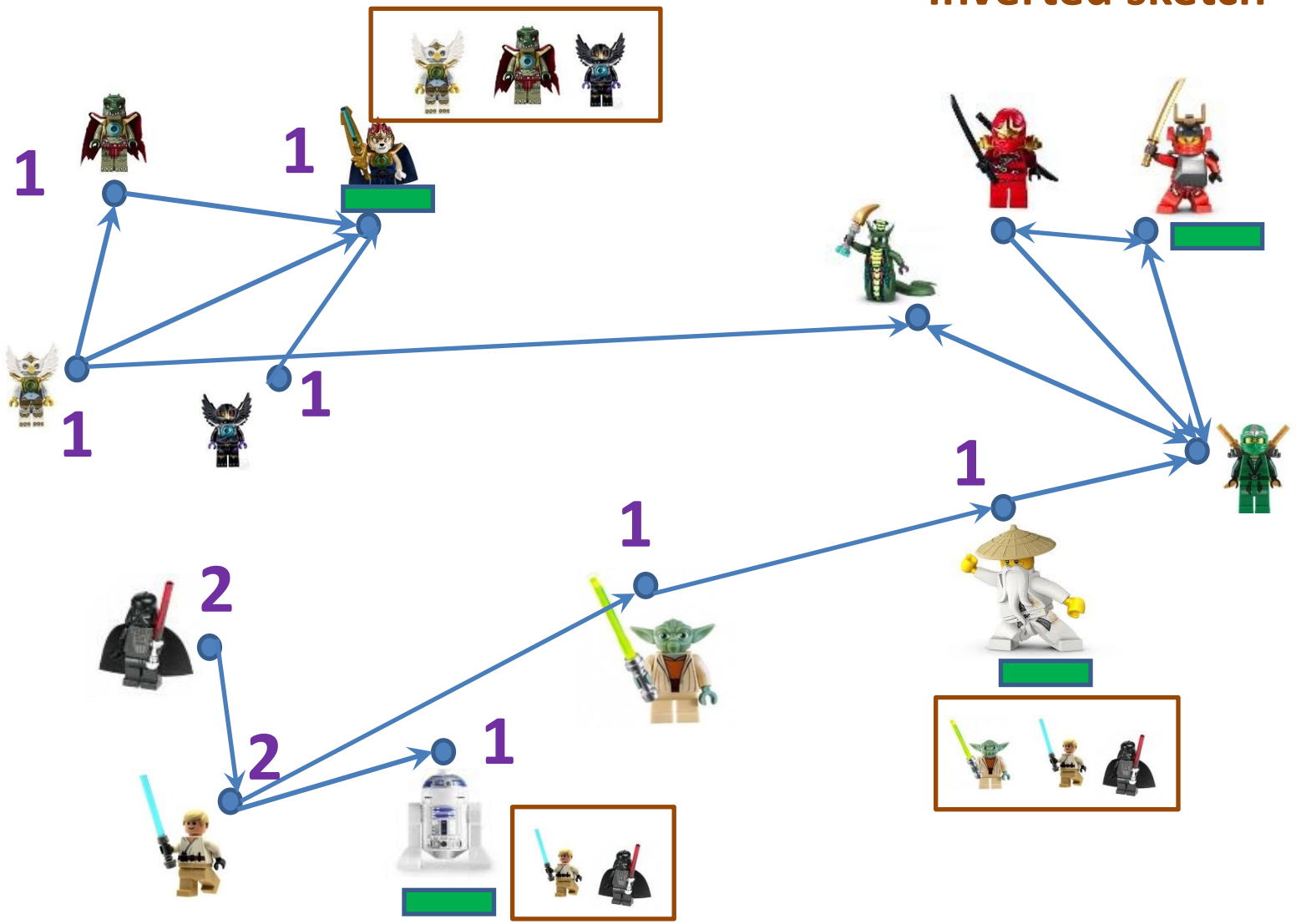


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

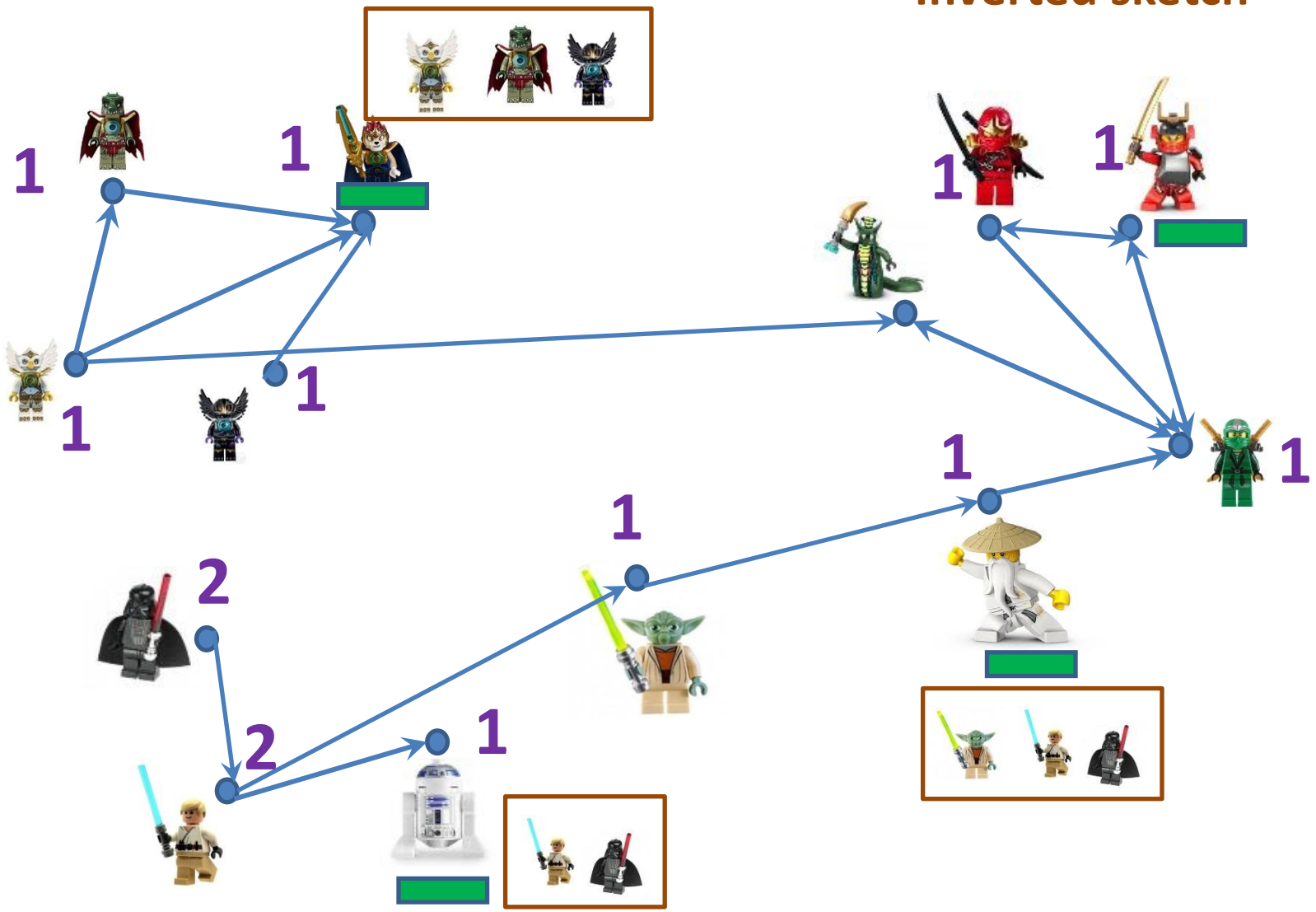


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

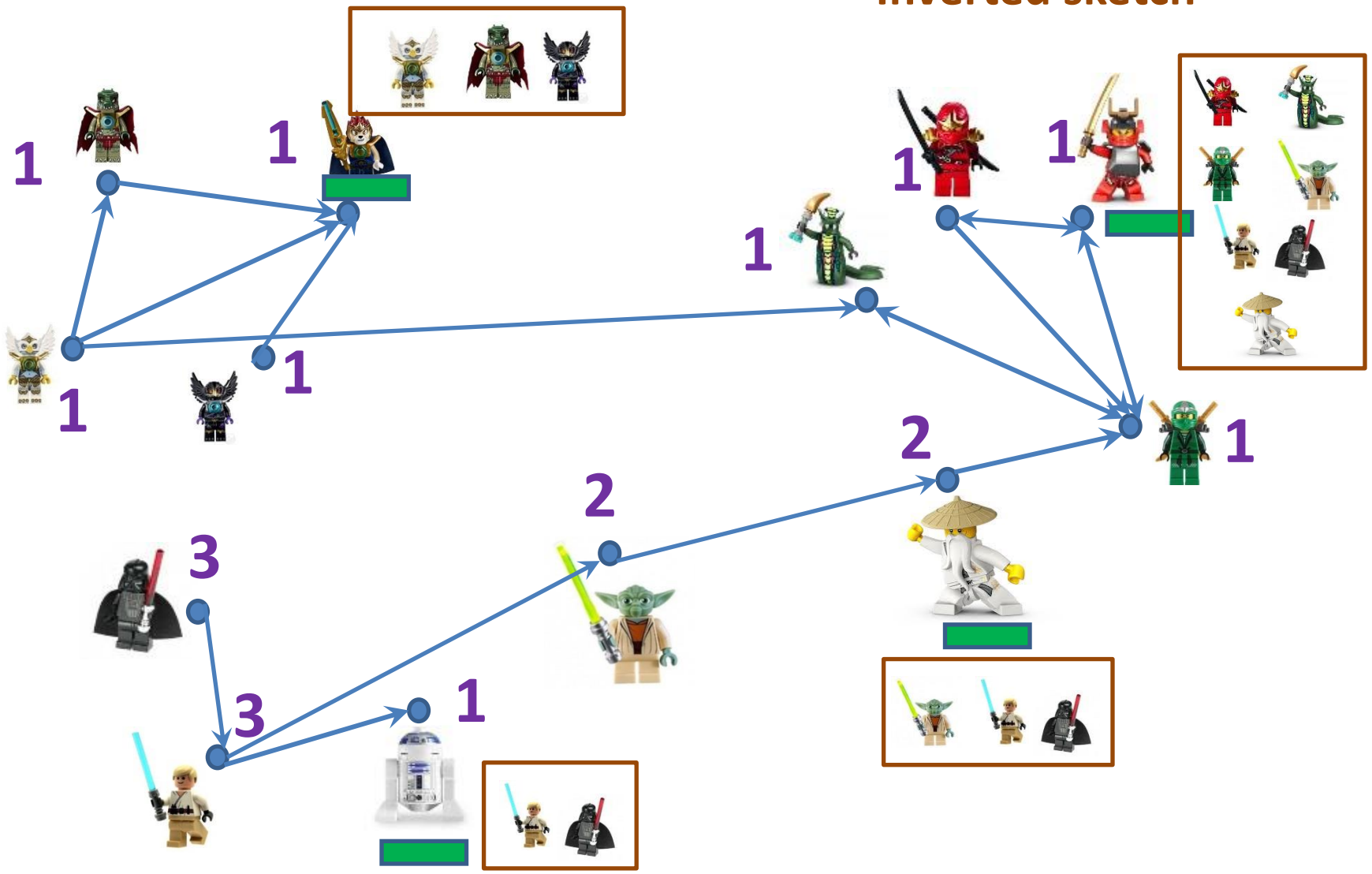


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch

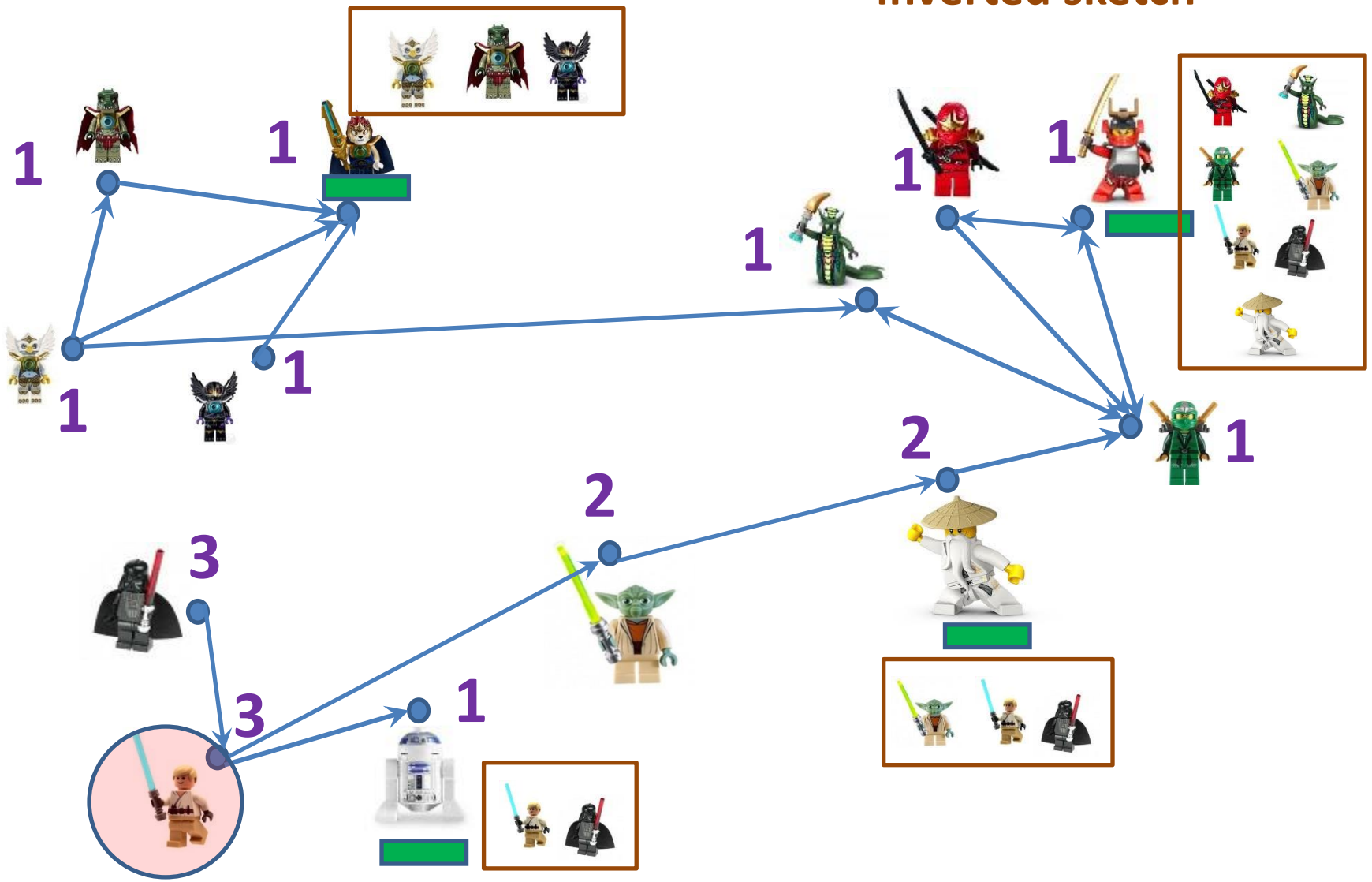


SkIM with $k = 3$

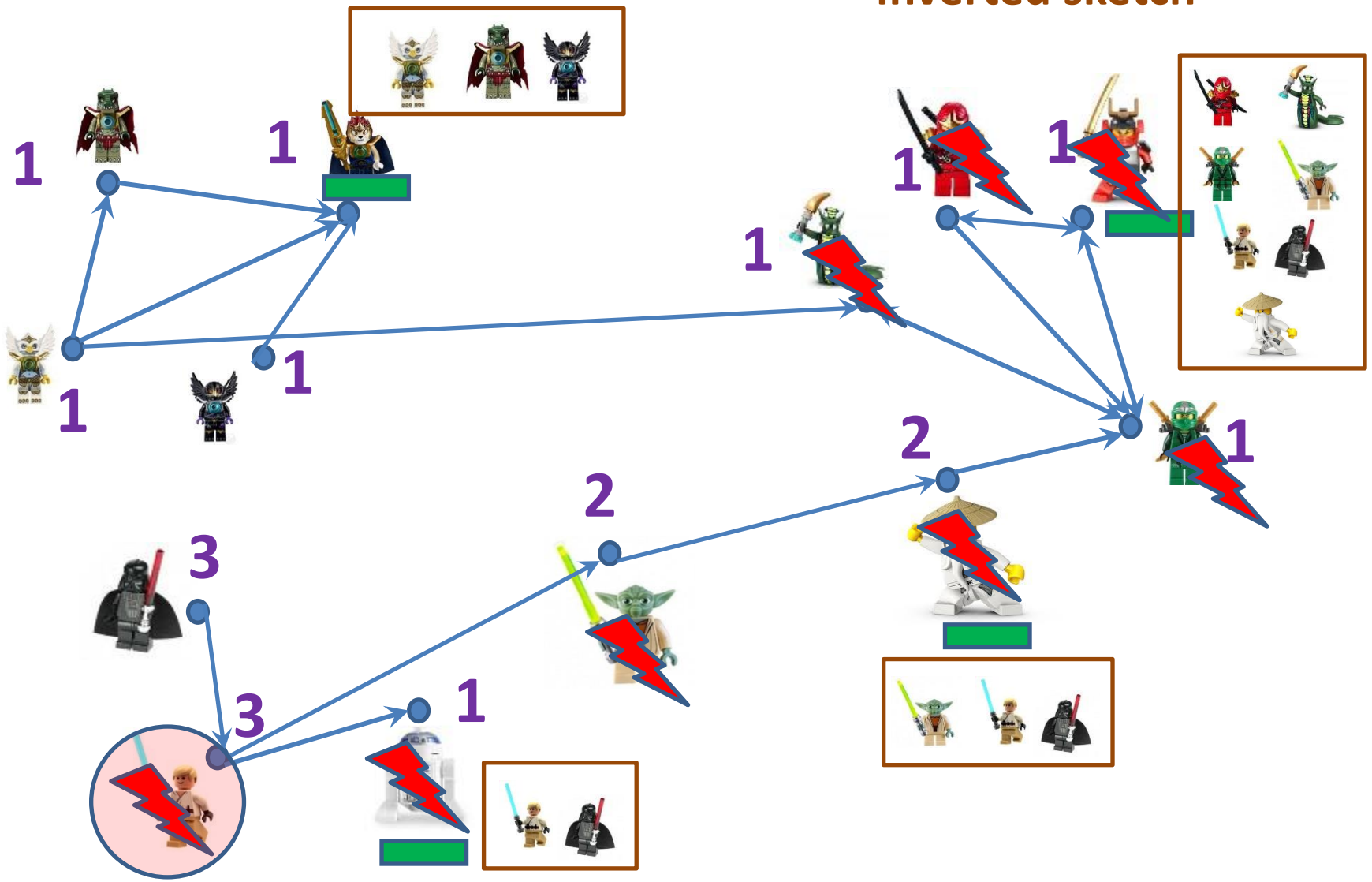
Sampled

Sketch size

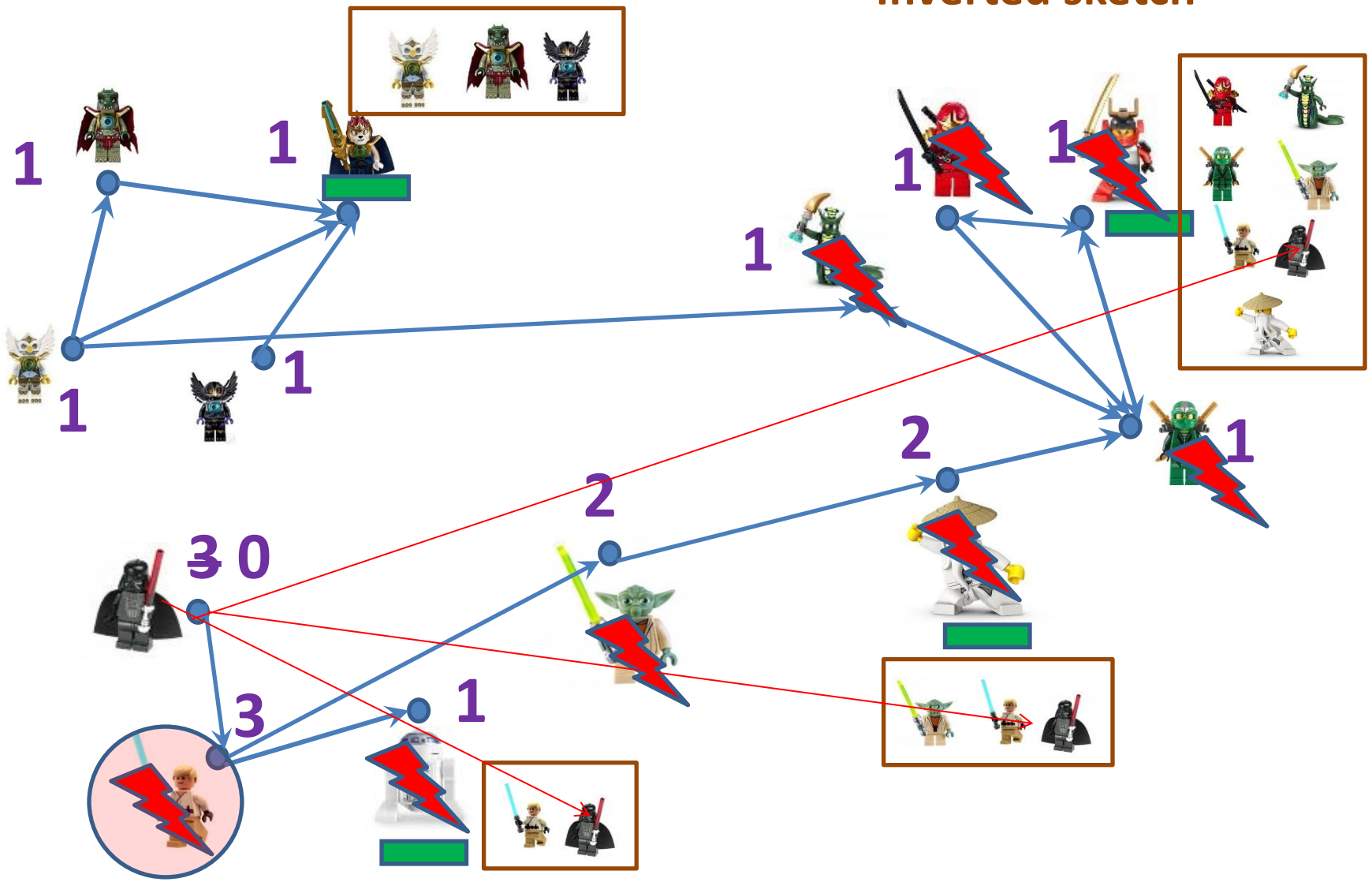
Inverted sketch



SkIM with $k = 3$



SkIM with $k = 3$

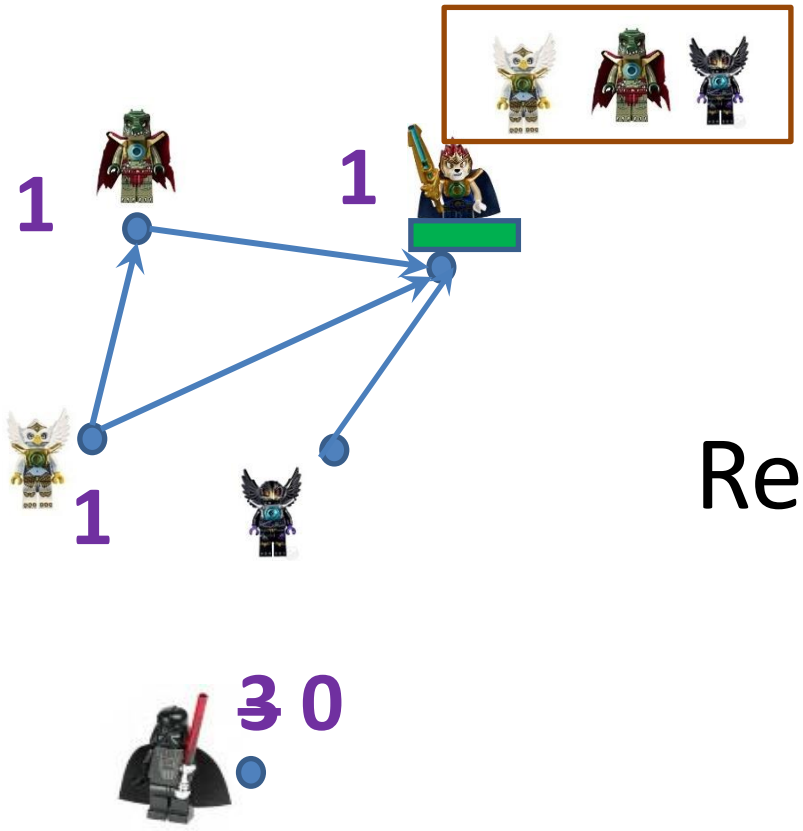


SkIM with $k = 3$

Sampled

Sketch size

Inverted sketch



Residual problem

SkIM correctness

- Bottom-k estimator depends only on largest value in sketch (threshold value): Highest estimate for node with smallest threshold. SkIM computes sketches only to the point that the one that would have the highest estimate (lowest threshold) is determined.
- From concentration, $k = O(\epsilon^{-2} \log \mathrm{n})$ suffices for relative error $1 \pm \epsilon$ WHP for *all* nodes in *all* iterations.
- Can verify that retained sketch entries correspond to residual problem.

SkIM running time (1 instance)

- Forward searches to remove selected and "covered" nodes are linear $O(m)$. Sketch decrements are "charged" to decrements
- Backward searches for sketch building:
  - Each node visit (and scan of in-edges) is charged to a new entry in sketch.
  - There are at most k entries at any particular time.
  - Entries get removed: but removals mean that statistically "marginal influence" decreases in expectation by $1 - \frac{1}{k}$. This can happen at most $k \log n$ times per node.

SkIM: Sketch Based Influence Maximization [CDPW CIKM 2014]

One instance: $O(\epsilon^{-2} m \log^2 n)$.

- We use $k = O(\epsilon^{-2} \log n)$
- In expectation, each node is visited $O(k \log n)$ times (total number of sketch entries)
- So we have $O(mk \log n) = O(m\epsilon^{-2} \log^2 n)$ edge traversals in total for sketch building
- We have $O(nk \log n) = O(n\epsilon^{-2} \log^2 n)$ total entries in sketches

Engineering SkIM

$\epsilon^{-2}\ \log^2\ n$ is a costly ! Is it really expressed in running time? Can we reduce it in practice and retain estimation guarantees (confidence) ?

- One $\log n$ factor is due to sketch entries analysis. In practice, it does not show up.
- The rest is due to working with $k = O(\epsilon^{-2}\ \log\ n)$. We can engineer around it.

…Engineering SkIM

Instead of using a "worst-case" $k = O(\epsilon^{-2} \log n)$, we adaptively estimate the error on the maximum and increase k only as needed. To estimate, we use:

- Computed exact marginal gain
- Other partial sketches to determine separation

We gain when:

- Max node is unique and separated from rest (can reduce the "ϵ^{-2}" dependence)
- Influence distribution is skewed (eliminate "union bound" $\log n$)
- When aiming for specific s, can increase ϵ on the go

SkIM on multiple instances

- Sketch building: “Elements” are node-instance pairs. Select randomly a remaining node-instance pair (v, i). Do a reverse search from v in instance i. Maintain $L_{(v,i)}$ of visited nodes.
- Residual problem: Forward search from u in each instance. If v is reached in instance i and $L_{(v,i)}$ exists. Decrement $sk[z]$ for all $z \in L_{(v,i)}$

SkIM: Sketch Based Influence Maximization [CDPW CIKM 2014]

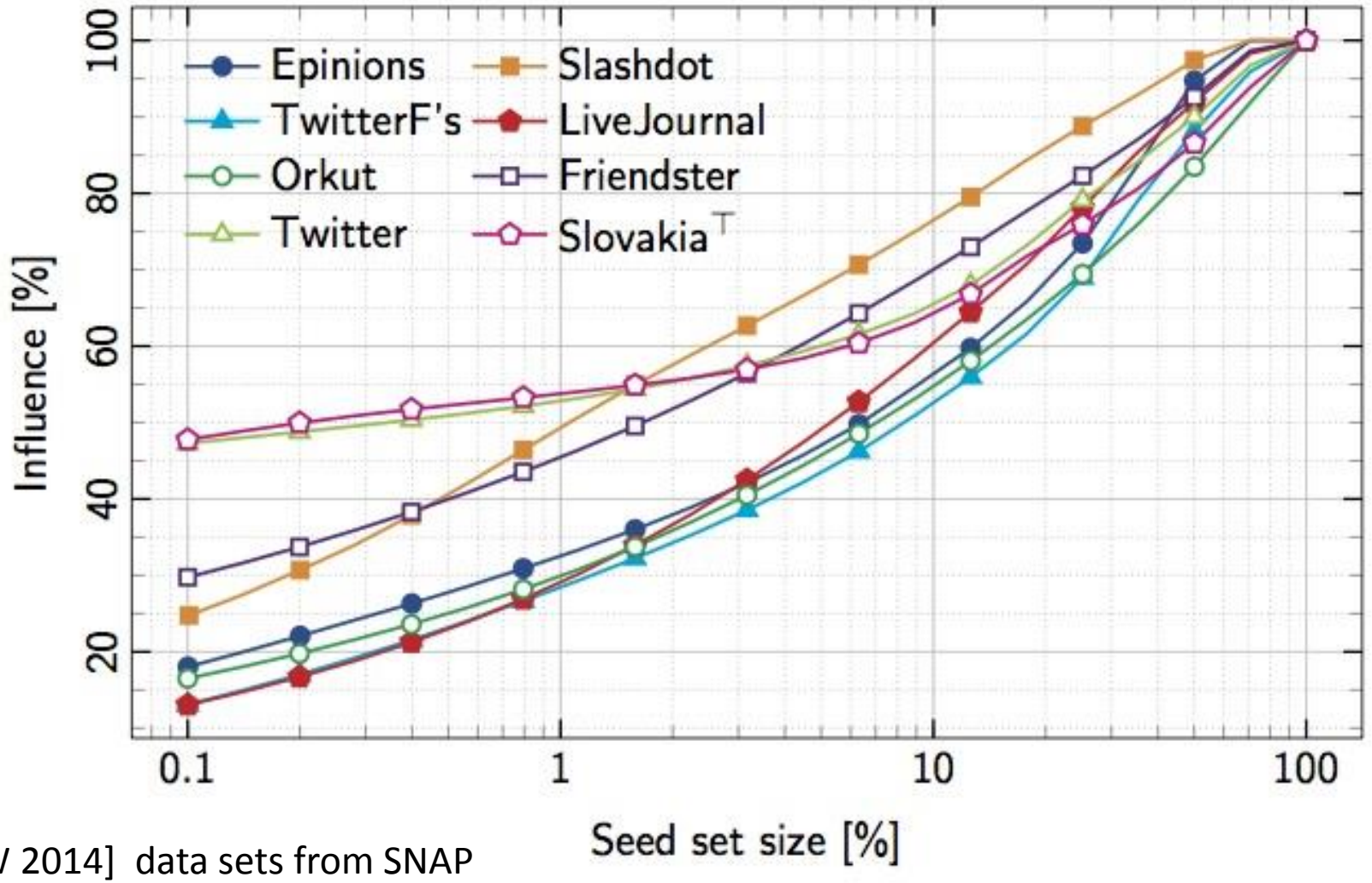
One instance: $O(\epsilon^{-2} m \log^2 n)$.

ℓ instances:

$O(\sum_{i=1}^{\ell} |E_i| + \epsilon^{-2} m \log^2 n)$. ($m$ is sum over nodes of max indegree in an instance)

IC model: ?? Conjecture that a "small number," perhaps $O(\epsilon^{-2} \log n)$, instances suffice

Full Influence Permutations: Influence



[CDPW 2014] data sets from SNAP

Bibliography: Reachability-based diffusion in networks

- Reachability sketches: E. Cohen "Size-Estimation Framework with Applications to Transitive Closure and Reachability" JCSS 1997
- IC model: Kempe, Kleinberg, Tardos "Maximizing the spread of influence through a social networks" KDD 2003

Use of reachability sketches for influence:

- Chen, Wang, Young. Efficient Influence Maximization in Social Networks. KDD 2009

Combined reachability sketches and scalable influence maximization:

- Cohen, Delling, Pajor, Werneck. Sketch-based Influence Maximization and Computation: Scaling up with Guarantees. CIKM 2014

Greedy algorithm for monotone submodular functions:

- Nemhauser, Wolsey, Fisher. "An analysis of the approximations of maximizing submodular set functions" 1978

❖ KDD 2012 tutorial: Castillo, Chen, Lakshmanan "information and influence spread in social networks" http://research.microsoft.com/en-us/people/weic/kdd12tutorial_inf.aspx

❖ There is a huge literature on scalable IM implementations (without guarantees…)

Further Modelling flexibility: “timed” influence
Distance-based diffusion in networks

N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*. 2013

M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, 2011.

Enhanced model and scalable algorithms:

- Cohen, Delling, Pajor, Werneck. Timed-influence: Computation and Maximization. http://arxiv.org/abs/1410.6976

All-Distances Sketches:

- E. Cohen “Size-Estimation Framework with Applications to Transitive Closure and Reachability” JCSS 1997
- All-Distances skethces, revisited. PODS 2014 http://arxiv.org/abs/1306.3284

Thank you