Approximating Max-Cut on Bounded Degree Graphs

Jun-Ting (Tim) Hsieh, CMU

Joint work with

Pravesh K. Kothari

Max-Cut

Problem: Given a graph $G = (V, E)$, the max-cut is

$$\max_{S \subseteq V} E(S, \overline{S})$$

Max-Cut

Problem: Given a graph $G = (V, E)$, the max-cut is

$$\max_{S \subseteq V} E(S, \overline{S})$$

Goemans-Williamson (GW) [1994]: a poly-time algorithm that achieves **0.878** approximation ratio:

- Finds an $S \subseteq V$ such that $E(S, \overline{S}) \geq 0.878 \cdot \mathrm{OPT}$.

Max-Cut

Problem: Given a graph $G = (V, E)$, the max-cut is

$$\max_{S \subseteq V} E(S, \overline{S})$$

Goemans-Williamson (GW) [1994]: a poly-time algorithm that achieves **0.878** approximation ratio:

- Finds an $S \subseteq V$ such that $E(S, \overline{S}) \geq 0.878 \cdot \mathrm{OPT}$.

This is optimal under the Unique-Games Conjecture!

- $0.878 + \epsilon$ approximation is NP-hard [Khot-Kindler-Mossel-O'Donnell'07].

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

- Feige-Karpinski-Langberg [2002]: $0.878 + \Omega(1/d^4)$ approximation.

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

- Feige-Karpinski-Langberg [2002]: 0.878 + $\Omega(1/d^4)$ approximation.
- Florén [2016]: 0.878 + $\Omega(1/d^3)$ approximation.

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

- Feige-Karpinski-Langberg [2002]: $0.878 + \Omega(1/d^4)$ approximation.
- Florén [2016]: $0.878 + \Omega(1/d^3)$ approximation.

Our result: $0.878 + \tilde{\Omega}(1/d^2)$ approximation.

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

- Feige-Karpinski-Langberg [2002]: $0.878 + \Omega(1/d^4)$ approximation.
- Florén [2016]: $0.878 + \Omega(1/d^3)$ approximation.

Our result: $0.878 + \tilde{\Omega}(1/d^2)$ approximation.

- Similar algorithm via local updates, better and cleaner analysis.

Max-Cut on bounded degree graphs

Can we do better on graphs with maximum degree d?

- Feige-Karpinski-Langberg [2002]: $0.878 + \Omega(1/d^4)$ approximation.
- Florén [2016]: $0.878 + \Omega(1/d^3)$ approximation.

Our result: $0.878 + \tilde{\Omega}(1/d^2)$ approximation.

- Similar algorithm via local updates, better and cleaner analysis.
- Extend to weighted instances of Max-2LIN (generalization of Max-Cut).

Goemans-Williamson: hyperplane rounding

Goemans-Williamson: hyperplane rounding

Write the Max-Cut objective as

Goemans-Williamson: hyperplane rounding

Write the Max-Cut objective as

$$\mathrm{OPT} := \max_{x \in \{\pm 1\}^n} \sum_{(i,j) \in E} \frac{1}{2}(1 - x_i x_j)$$

Goemans-Williamson: hyperplane rounding

Write the Max-Cut objective as

$$\mathrm{OPT} := \max_{x \in \{\pm 1\}^n} \sum_{(i,j) \in E} \frac{1}{2}(1 - x_i x_j)$$

Semidefinite program (SDP) relaxation: find unit vectors $v_1, \ldots, v_n \in \mathcal{S}^{n-1}$:

Goemans-Williamson: hyperplane rounding

Write the Max-Cut objective as

$$\mathrm{OPT} := \max_{x \in \{\pm 1\}^n} \sum_{(i,j) \in E} \frac{1}{2}(1 - x_i x_j)$$

Semidefinite program (SDP) relaxation: find unit vectors $v_1, \ldots, v_n \in \mathcal{S}^{n-1}$:

$$\mathrm{SDP} := \max_{v_1, \ldots, v_n \in \mathcal{S}^{n-1}} \sum_{(i,j) \in E} \frac{1}{2}(1 - \langle v_i, v_j \rangle)$$

Goemans-Williamson: hyperplane rounding

Write the Max-Cut objective as

$$\mathrm{OPT} := \max_{x \in \{\pm 1\}^n} \sum_{(i,j) \in E} \frac{1}{2}(1 - x_i x_j)$$

Semidefinite program (SDP) relaxation: find unit vectors $v_1, \ldots, v_n \in \mathcal{S}^{n-1}$:

$$\mathrm{SDP} := \max_{v_1, \ldots, v_n \in \mathcal{S}^{n-1}} \sum_{(i,j) \in E} \frac{1}{2}(1 - \langle v_i, v_j \rangle)$$

Note: SDP $\geq$ OPT.

Goemans-Williamson: hyperplane rounding

Given $v_1, \ldots, v_n \in \mathbb{R}^n$,

Goemans-Williamson: hyperplane rounding

Given $v_1, \ldots, v_n \in \mathbb{R}^n$,

- Sample standard Gaussian $g \sim \mathcal{N}(0, I_n)$.

Goemans-Williamson: hyperplane rounding

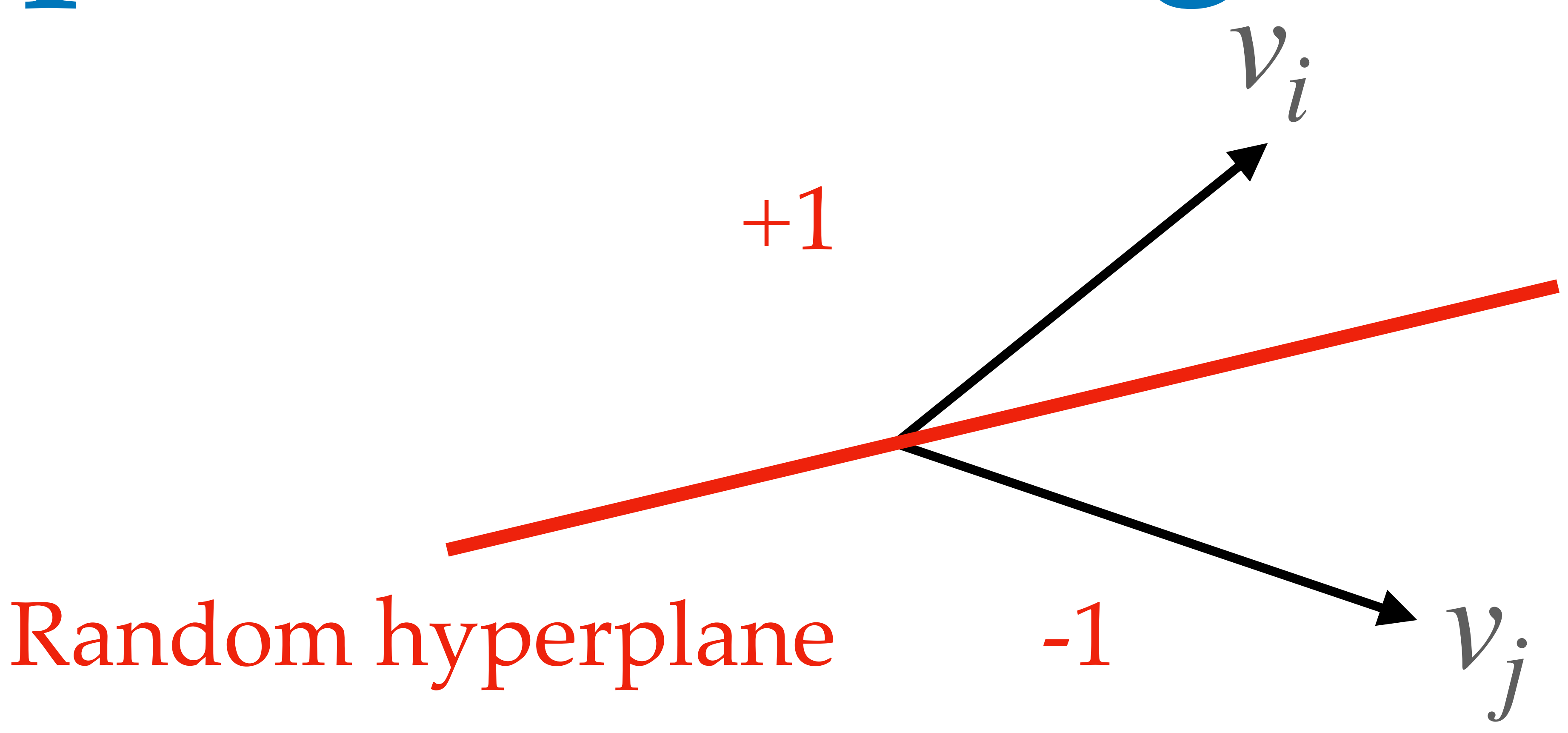
Given $v_1, \ldots, v_n \in \mathbb{R}^n$,

- Sample standard Gaussian $g \sim \mathcal{N}(0, I_n)$.
- Set $x_i = \text{sign}(\langle g, v_i \rangle)$.

Goemans-Williamson: hyperplane rounding

Given $v_1, \ldots, v_n \in \mathbb{R}^n$,

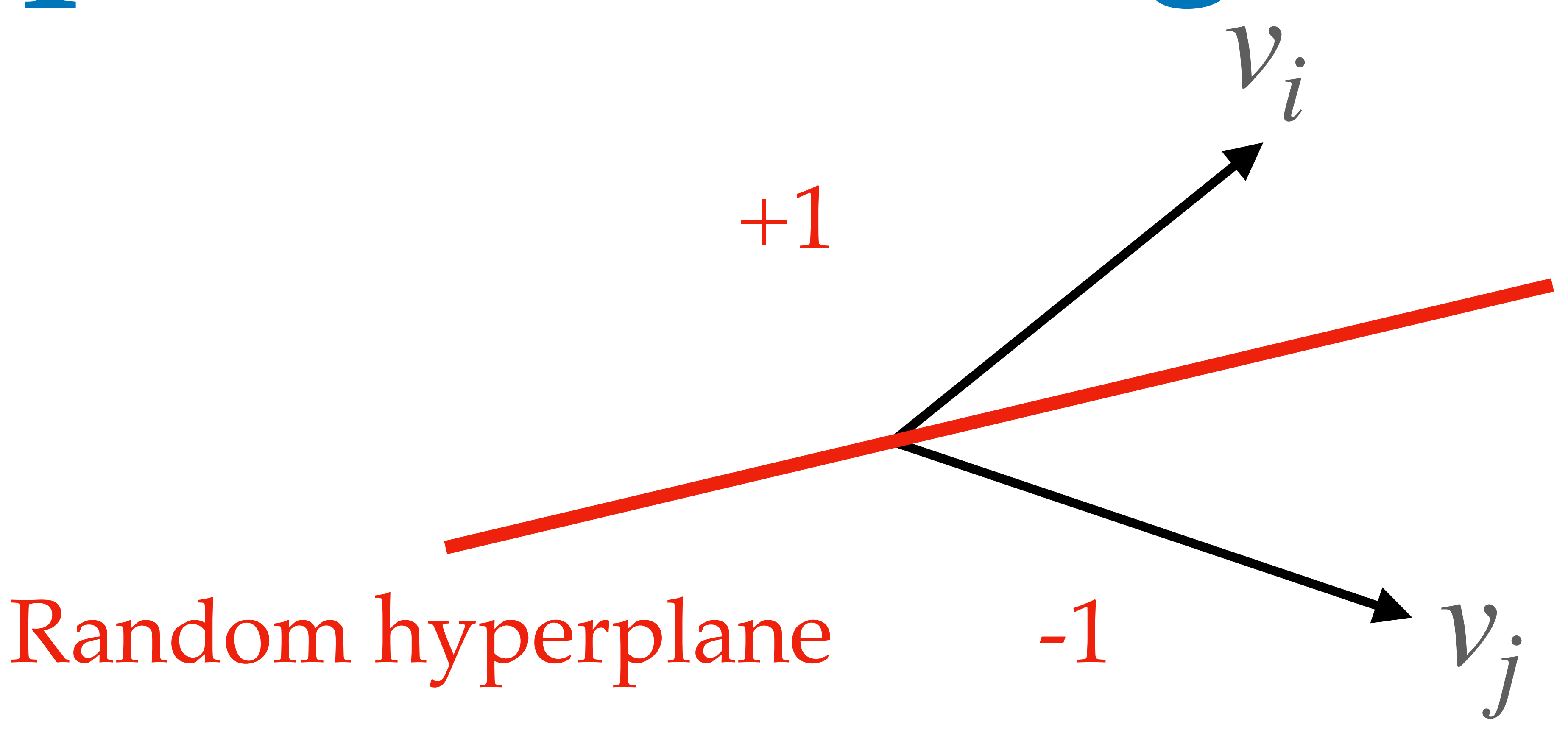
- Sample standard Gaussian $g \sim \mathcal{N}(0, I_n)$.
- Set $x_i = \text{sign}(\langle g, v_i \rangle)$.



Goemans-Williamson: hyperplane rounding

Given $v_1, \ldots, v_n \in \mathbb{R}^n$,

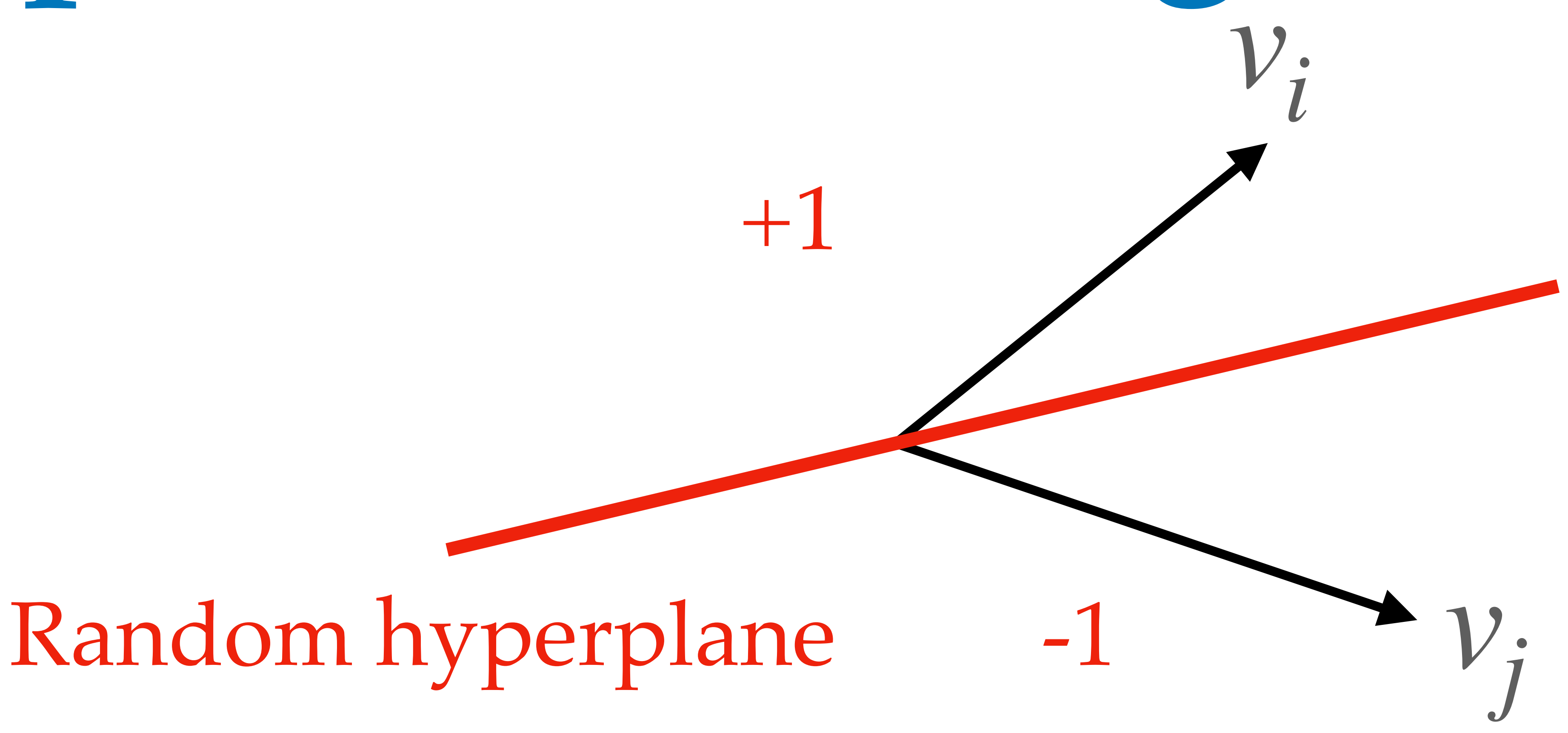
- Sample standard Gaussian $g \sim \mathcal{N}(0, I_n)$.
- Set $x_i = \text{sign}(\langle g, v_i \rangle)$.



$$\Pr[i, j \text{ cut}] = \frac{\arccos(\langle v_i, v_j \rangle)}{\pi}$$

Goemans-Williamson: hyperplane rounding

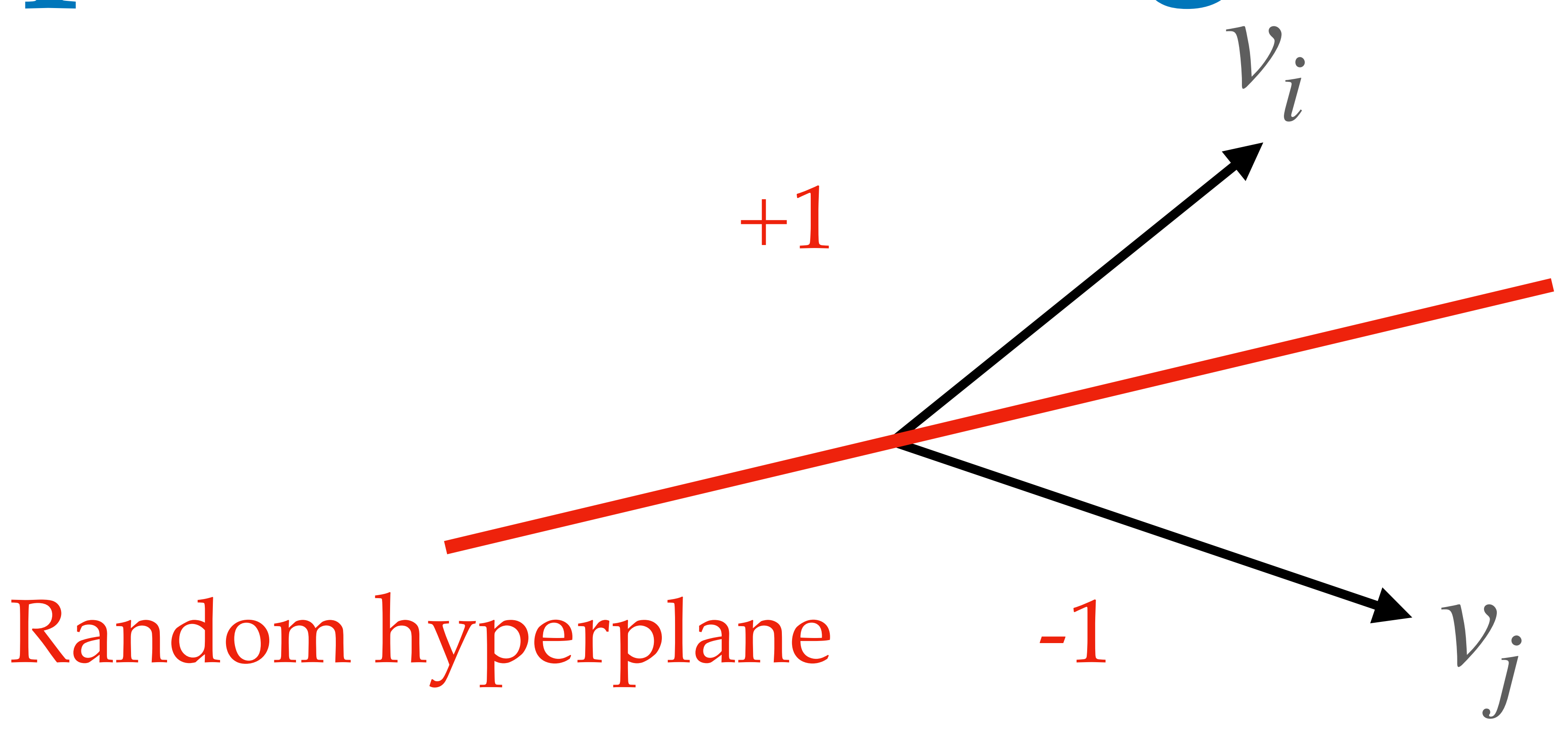
$$\mathbb{E}[\text{Cut}] = \sum_{(i,j)\in E} \frac{\arccos(\langle v_i, v_j \rangle)}{\pi}$$



Goemans-Williamson: hyperplane rounding

$$\mathbb{E}[\text{Cut}] = \sum_{(i,j)\in E} \frac{\arccos(\langle v_i, v_j \rangle)}{\pi}$$

$$\text{OPT} \leq \text{SDP} = \sum_{(i,j)\in E} \frac{1}{2}(1 - \langle v_i, v_j \rangle)$$

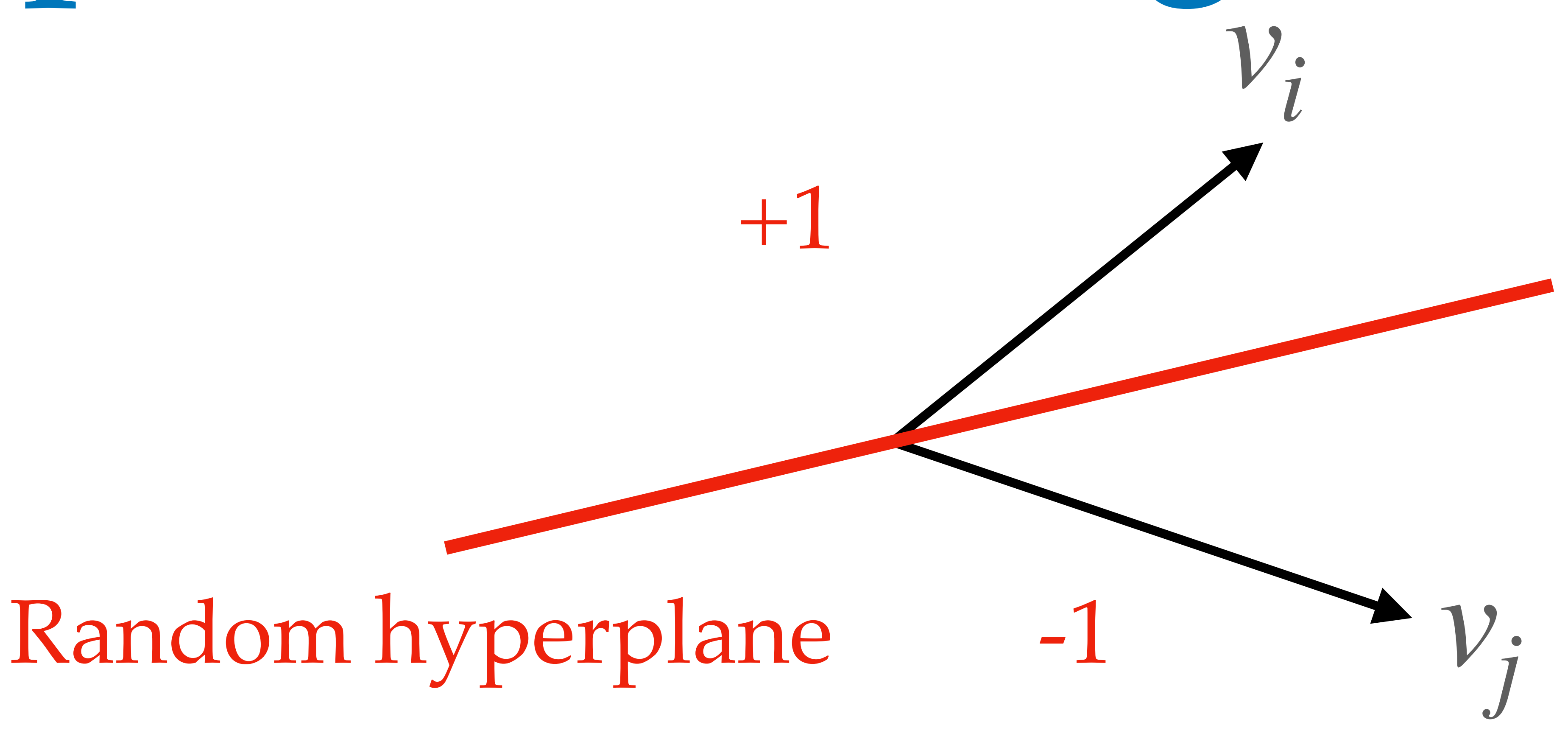


Goemans-Williamson: hyperplane rounding

$$\mathbb{E}[\text{Cut}] = \sum_{(i,j)\in E} \frac{\arccos(\langle v_i, v_j \rangle)}{\pi}$$

$$\text{OPT} \leq \text{SDP} = \sum_{(i,j)\in E} \frac{1}{2}(1 - \langle v_i, v_j \rangle)$$

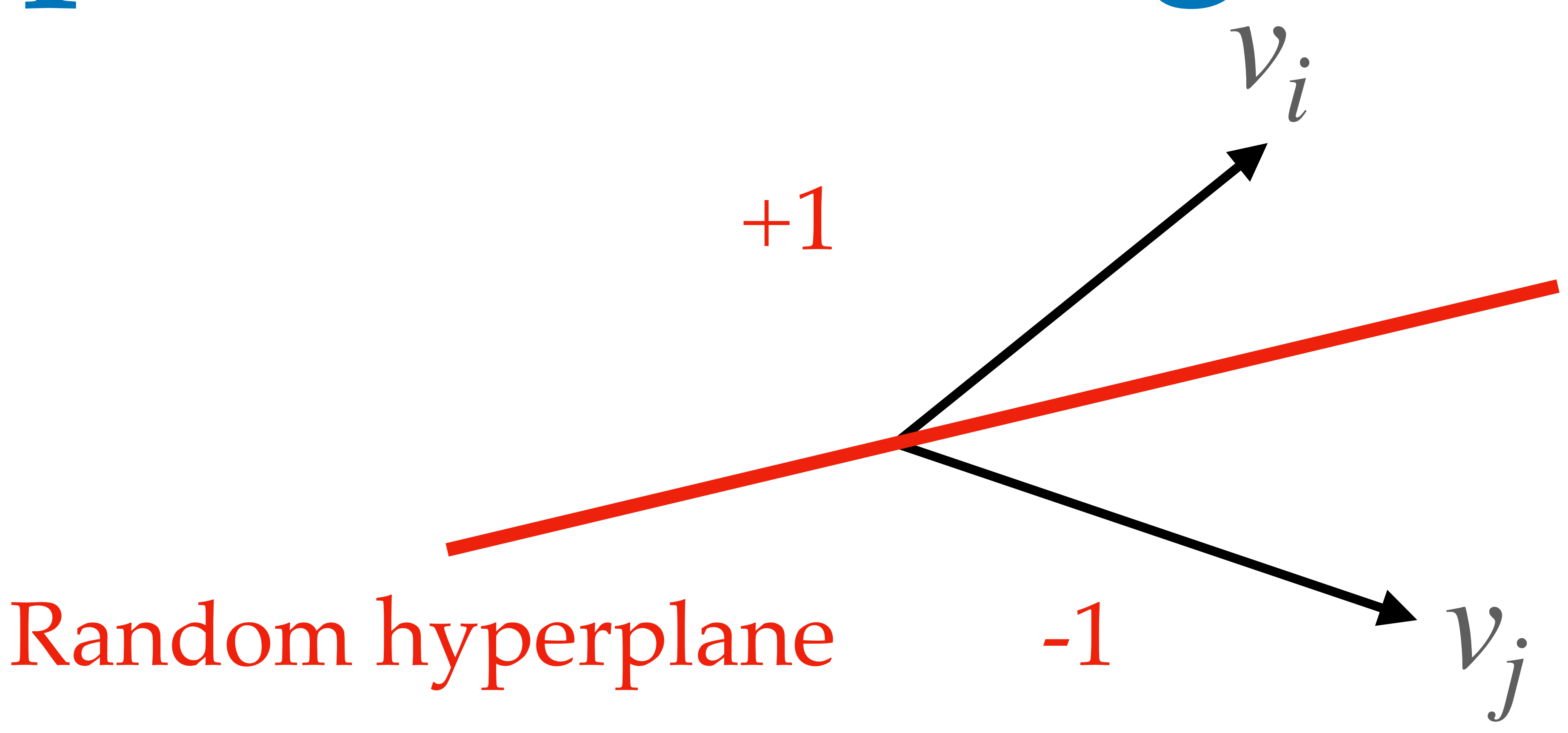
$$\min_{\rho\in[-1,1]} \frac{\frac{1}{\pi}\arccos(\rho)}{\frac{1}{2}(1-\rho)} = 0.878, \text{ minimized at } \rho^* \approx -0.69.$$



Goemans-Williamson: hyperplane rounding

$$\mathbb{E}[\text{Cut}] = \sum_{(i,j)\in E} \frac{\arccos(\langle v_i, v_j\rangle)}{\pi}$$

$$\text{OPT} \leq \text{SDP} = \sum_{(i,j)\in E} \frac{1}{2}(1 - \langle v_i, v_j\rangle)$$



$$\min_{\rho\in[-1,1]} \frac{\frac{1}{\pi}\arccos(\rho)}{\frac{1}{2}(1-\rho)} = 0.878, \text{ minimized at } \rho^* \approx -0.69.$$

$$\mathbb{E}[\text{Cut}] \geq 0.878 \cdot \text{SDP} \geq 0.878 \cdot \text{OPT}.$$

Local updates

How do we improve GW rounding to $0.878 + \epsilon(d)$?

Local updates

How do we improve GW rounding to $0.878 + \epsilon(d)$?

Observation 1: we can assume that almost all edges $(i, j) \in E$ satisfy $\langle v_i, v_j \rangle \approx -0.69$.

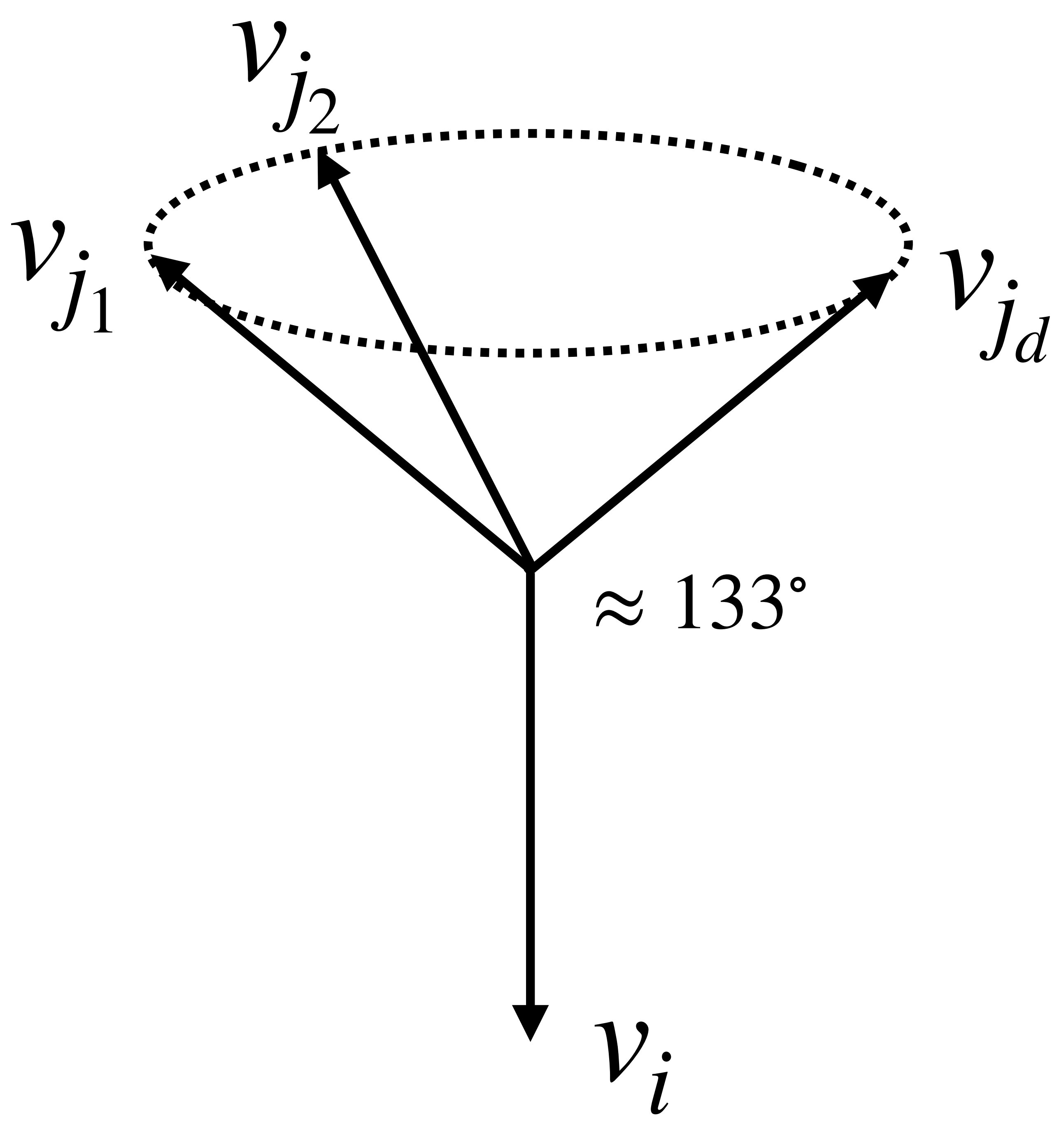
- Otherwise we already get an improvement.

Local updates

How do we improve GW rounding to $0.878 + \epsilon(d)$?

Observation 1: we can assume that almost all edges $(i, j) \in E$ satisfy $\langle v_i, v_j \rangle \approx -0.69$.

- Otherwise we already get an improvement.



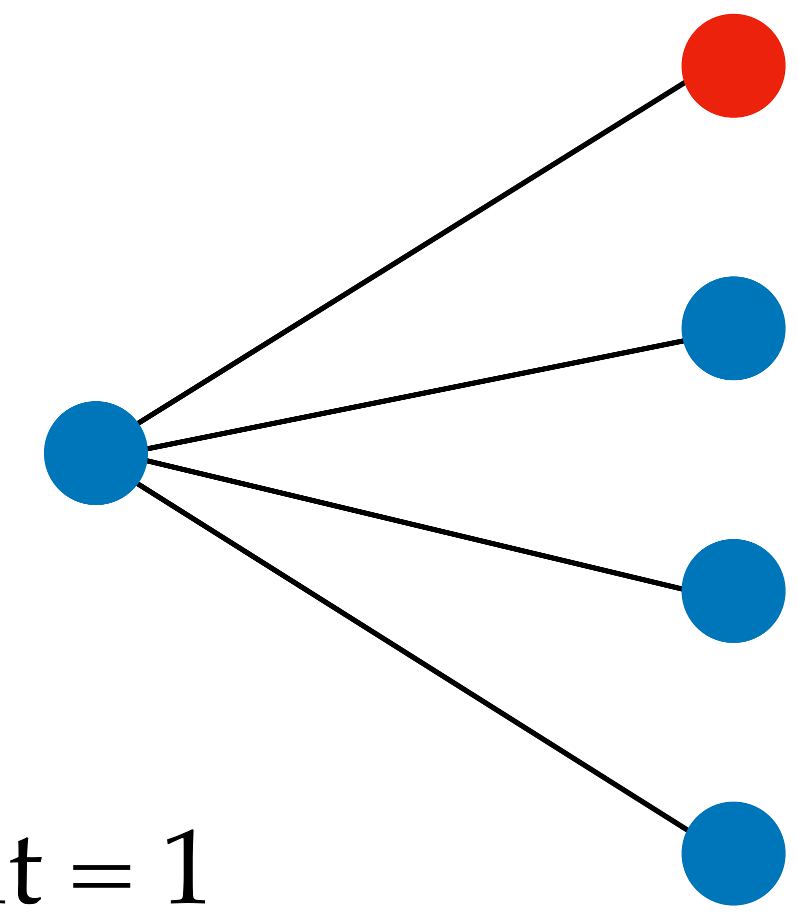
Local updates

Local updates

Observation 2: we can flip a vertex if more than half of its neighbors have the same sign.

Local updates

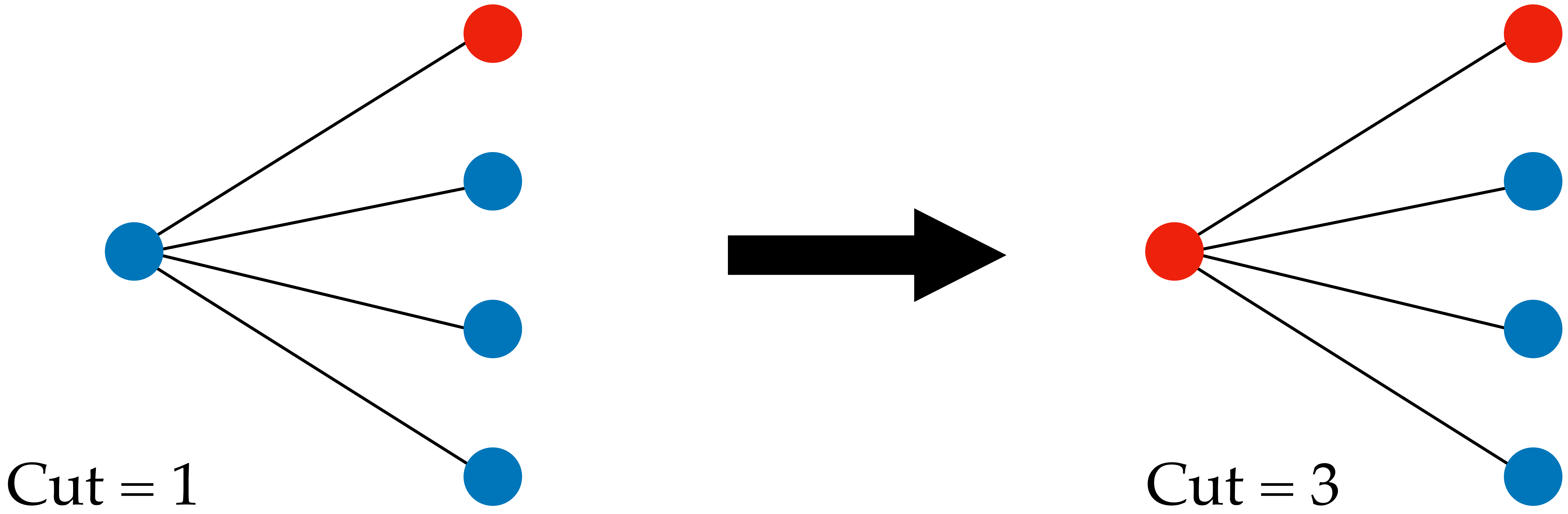
Observation 2: we can flip a vertex if more than half of its neighbors have the same sign.



Cut = 1

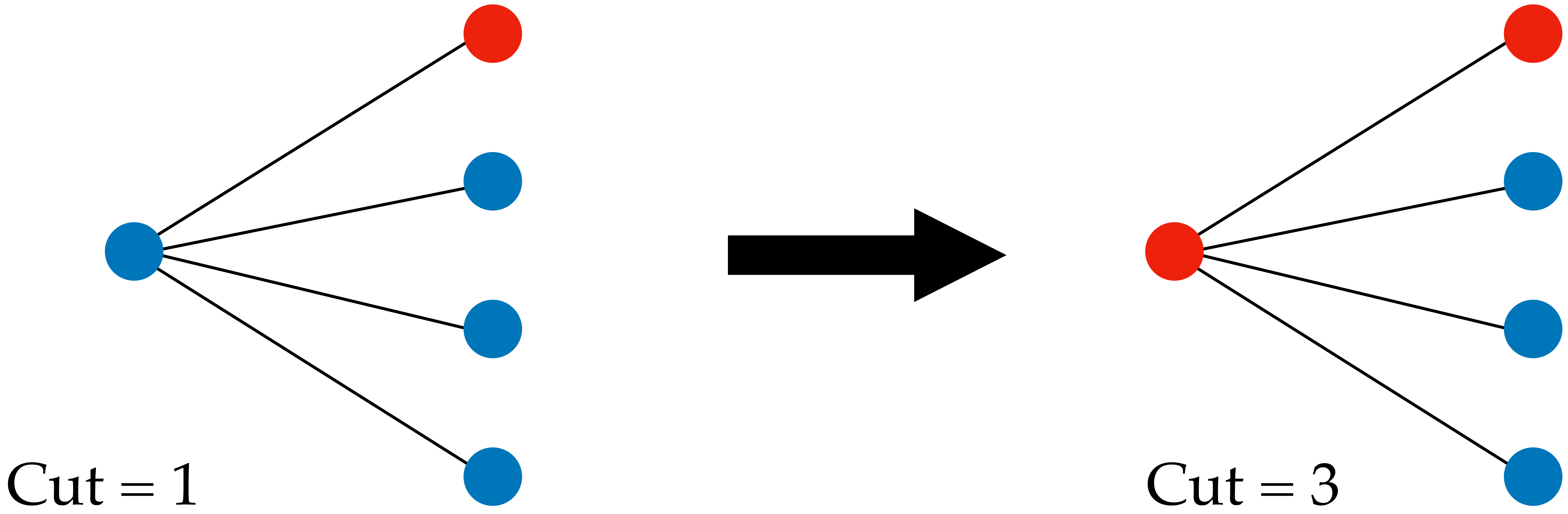
Local updates

Observation 2: we can flip a vertex if more than half of its neighbors have the same sign.



Local updates

Observation 2: we can flip a vertex if more than half of its neighbors have the same sign.



What's the probability that this occurs?

Local updates

By rotation invariance, we can assume WLOG,

Local updates

By rotation invariance, we can assume WLOG,

- $v_i = (1, 0, 0, \ldots)$,

Local updates

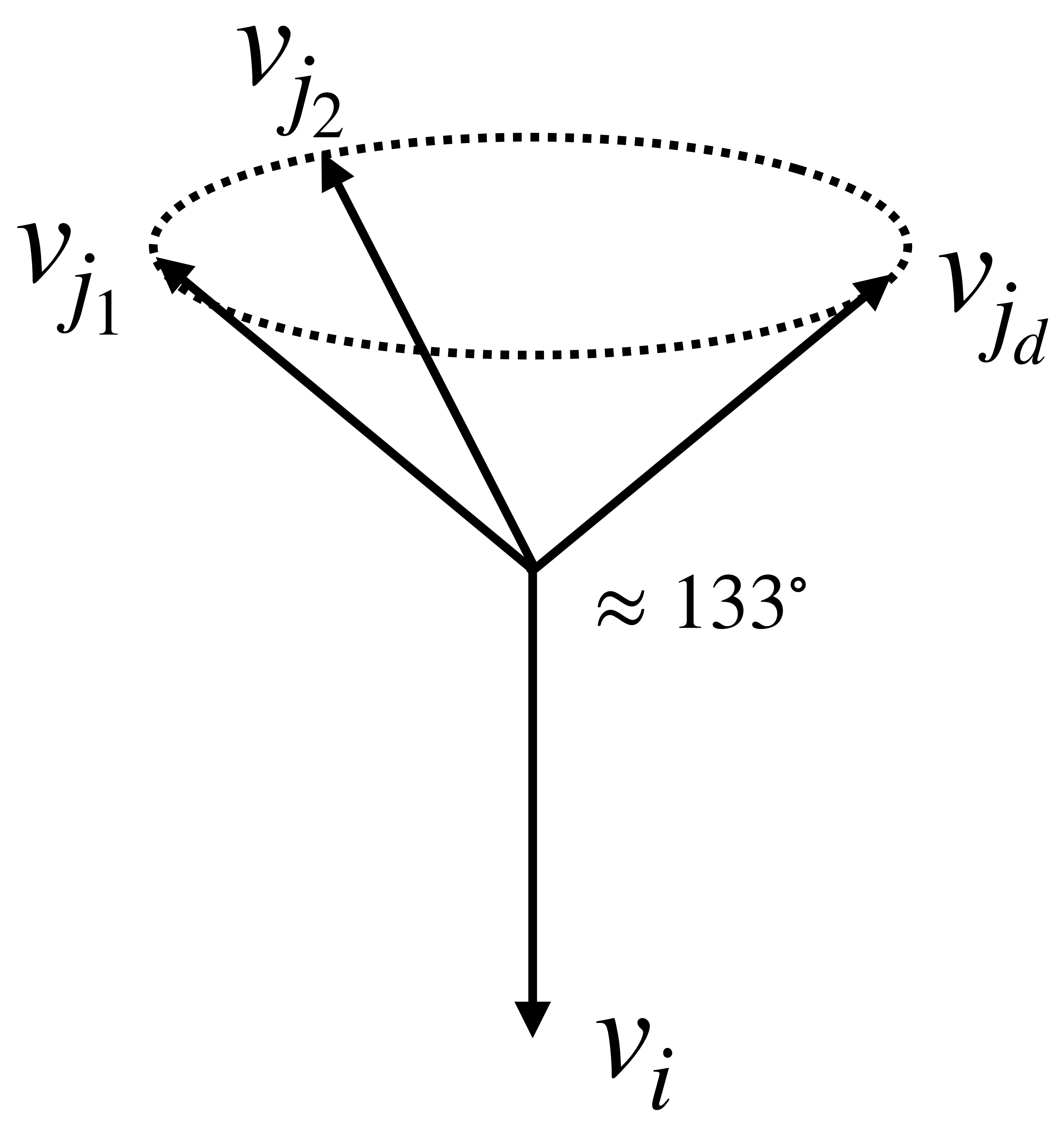
By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

Local updates

By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots)$,
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

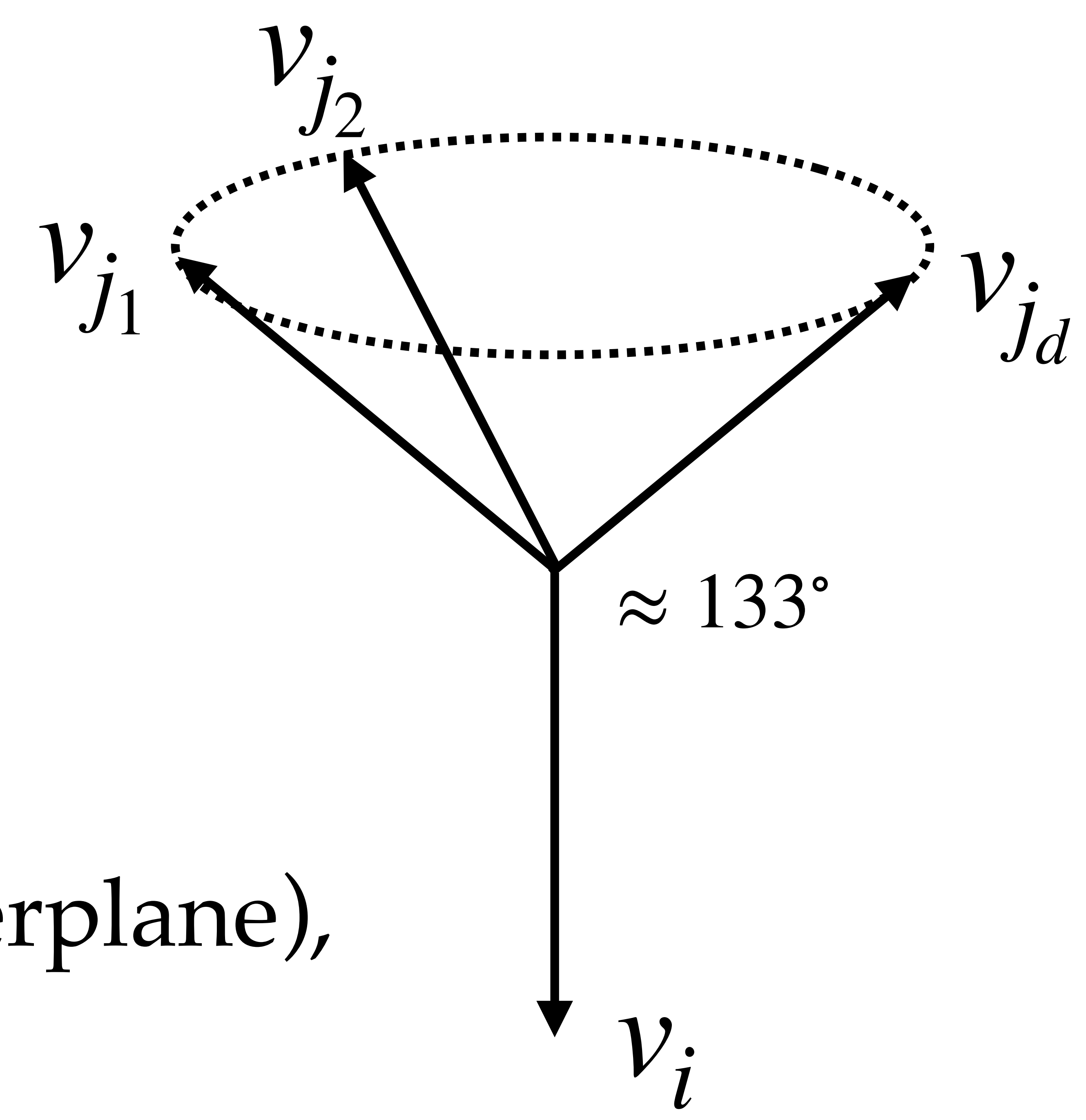


Local updates

By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots)$,
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

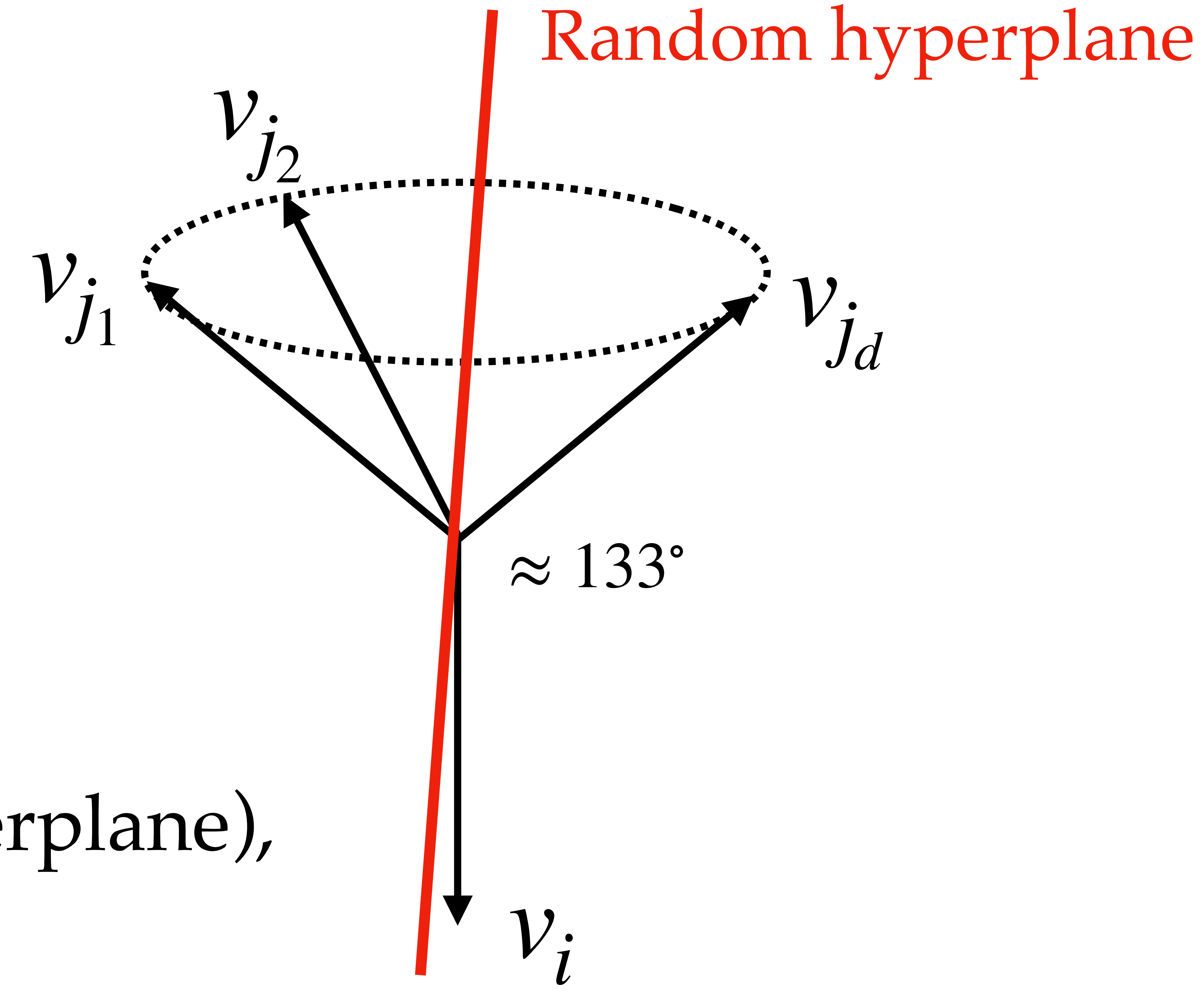
[FKL'02]: suppose $\langle g, v_i \rangle = g_1 \approx 0$ (i.e., v_i close to the hyperplane),



Local updates

By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots)$,
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.



[FKL'02]: suppose $\langle g, v_i \rangle = g_1 \approx 0$ (i.e., v_i close to the hyperplane),

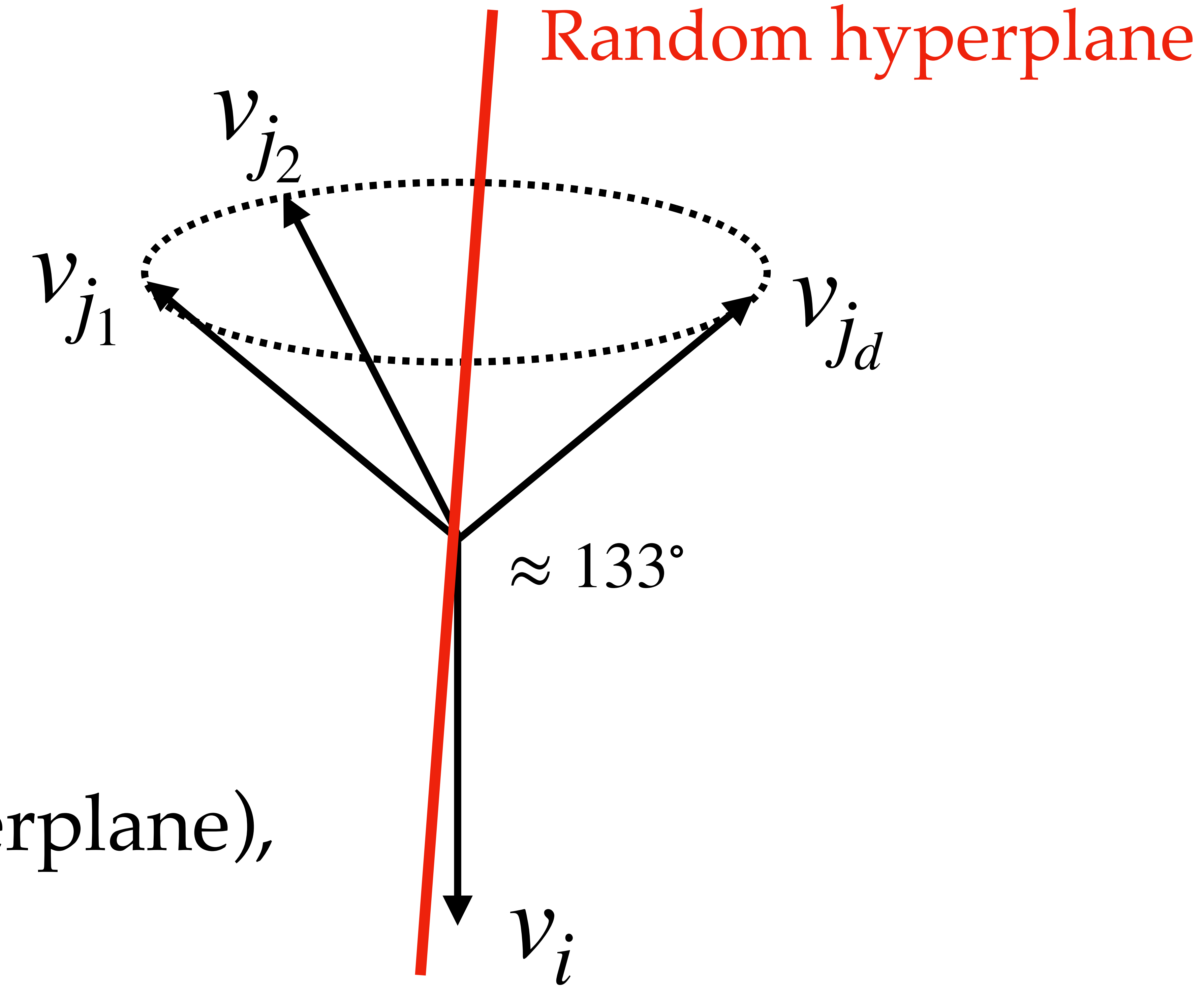
Local updates

By rotation invariance, we can assume WLOG,

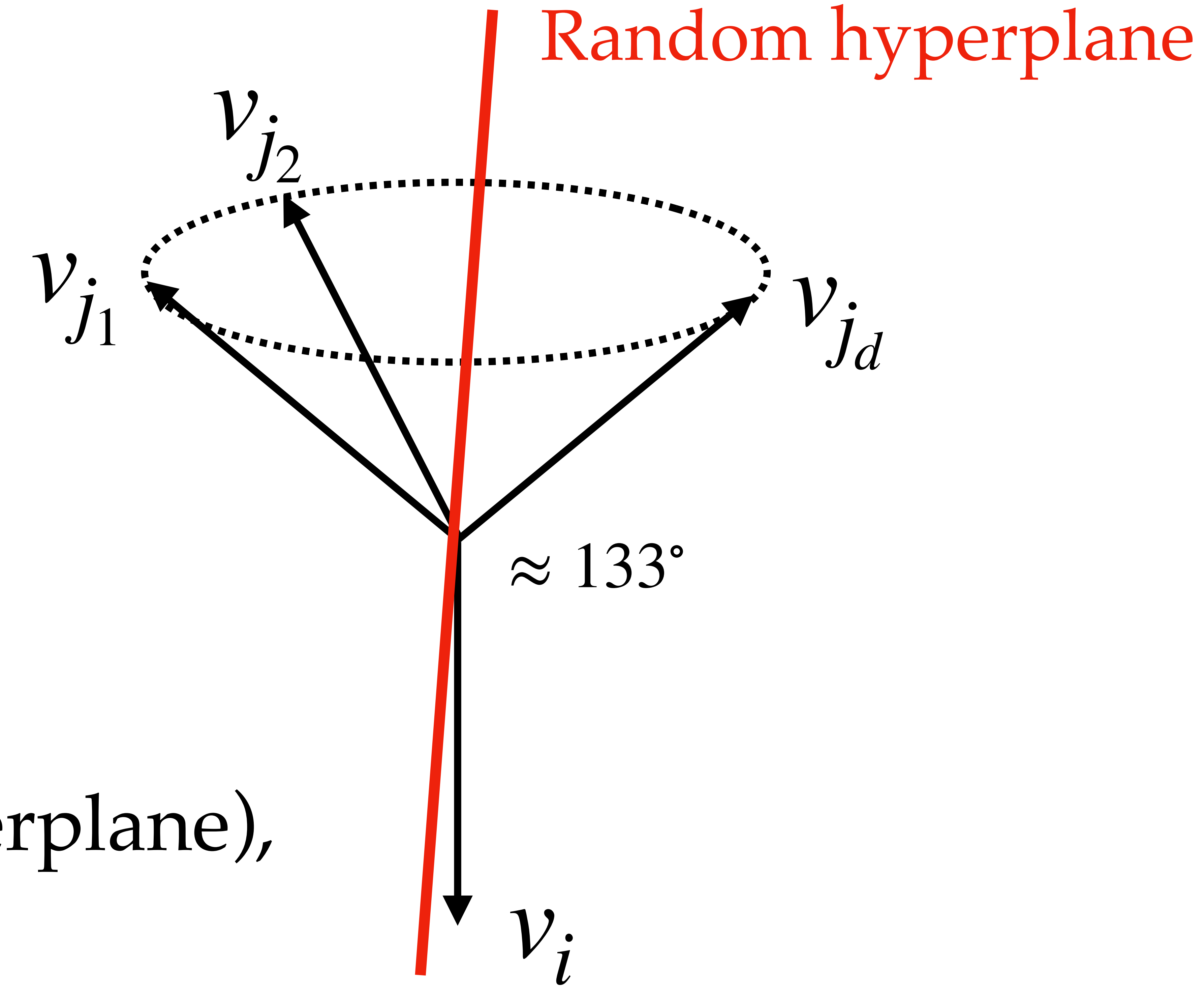
- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

[FKL'02]: suppose $\langle g, v_i \rangle = g_1 \approx 0$ (i.e., v_i close to the hyperplane),

- then $\text{sign}(\langle g, v_j \rangle)$ is roughly independent of $\text{sign}(g_1)$.



Local updates



By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

[FKL'02]: suppose $\langle g, v_i \rangle = g_1 \approx 0$ (i.e., v_i close to the hyperplane),

- then $\text{sign}(\langle g, v_j \rangle)$ is roughly independent of $\text{sign}(g_1)$.
- With probability $\approx 1/2$, vertex i will be placed on the majority side (hence we can flip).

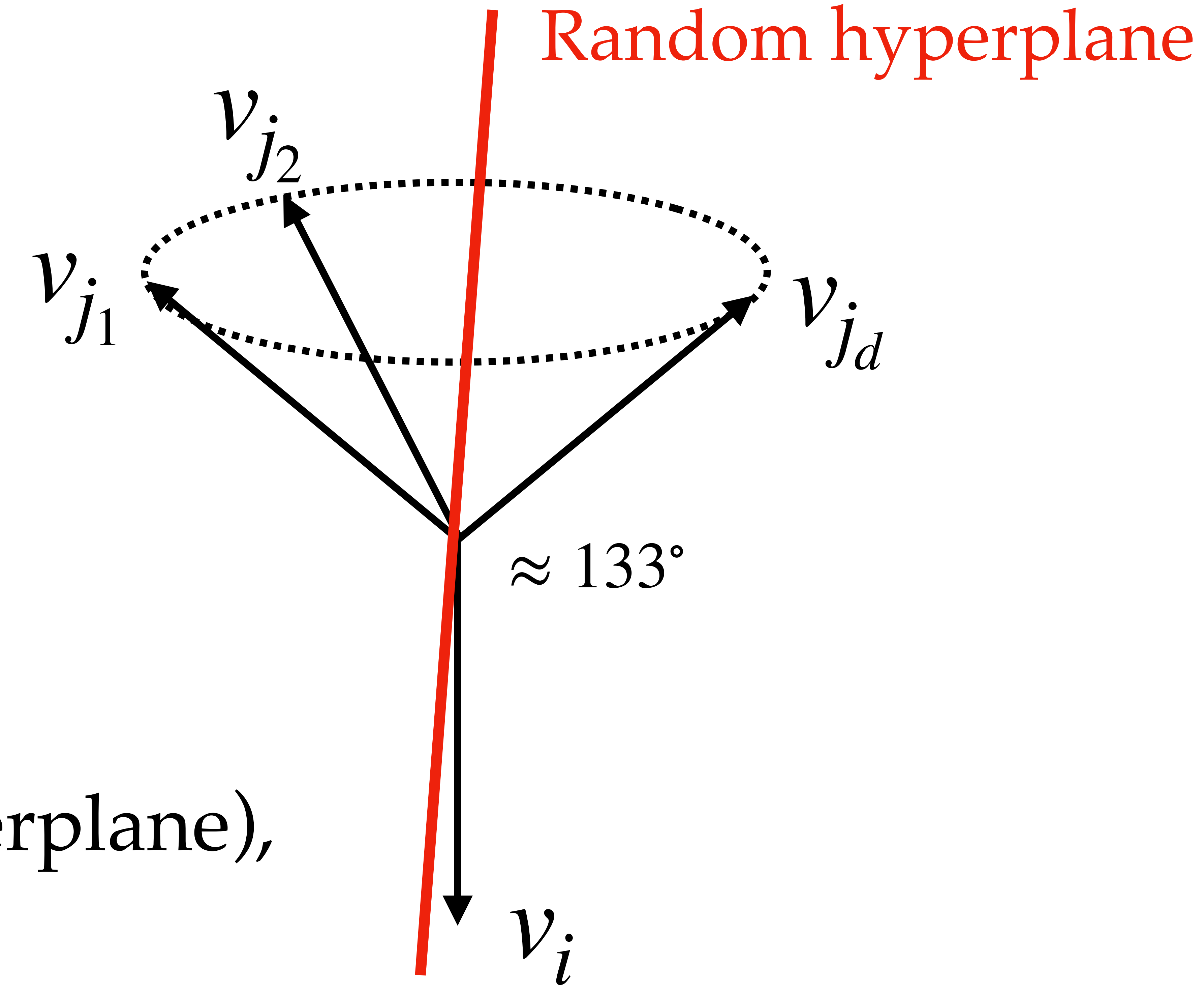
Local updates

By rotation invariance, we can assume WLOG,

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

[FKL'02]: suppose $\langle g, v_i \rangle = g_1 \approx 0$ (i.e., v_i close to the hyperplane),

- then $\mathrm{sign}(\langle g, v_j \rangle)$ is roughly independent of $\mathrm{sign}(g_1)$.
- With probability $\approx 1/2$, vertex i will be placed on the majority side (hence we can flip).



Loses an extra $1/d$ factor to handle when d is even.

Vertex dependencies

Naively, flip all such vertices?

Vertex dependencies

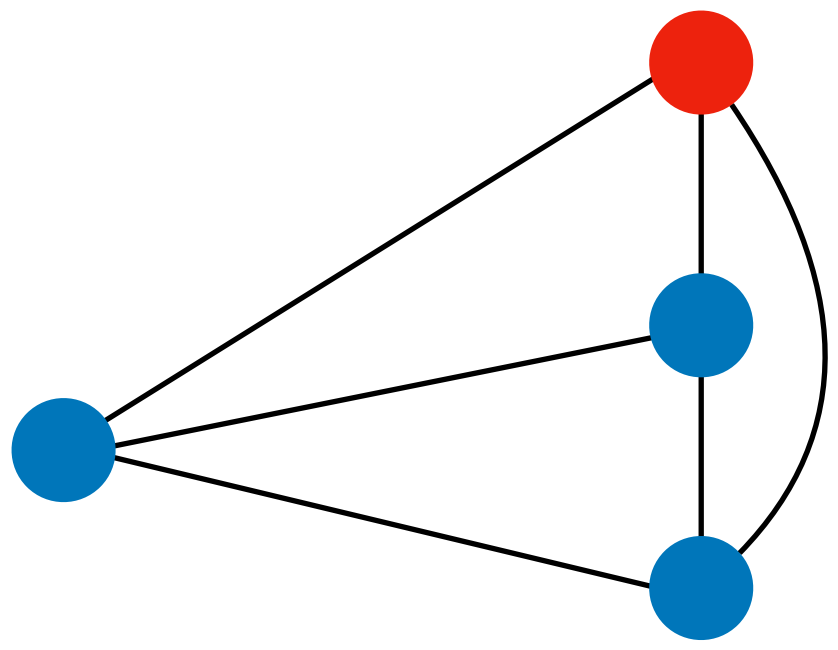
Naively, flip all such vertices?

Problem: flipping a vertex may “kill” the gains from other vertices.

Vertex dependencies

Naively, flip all such vertices?

Problem: flipping a vertex may "kill" the gains from other vertices.

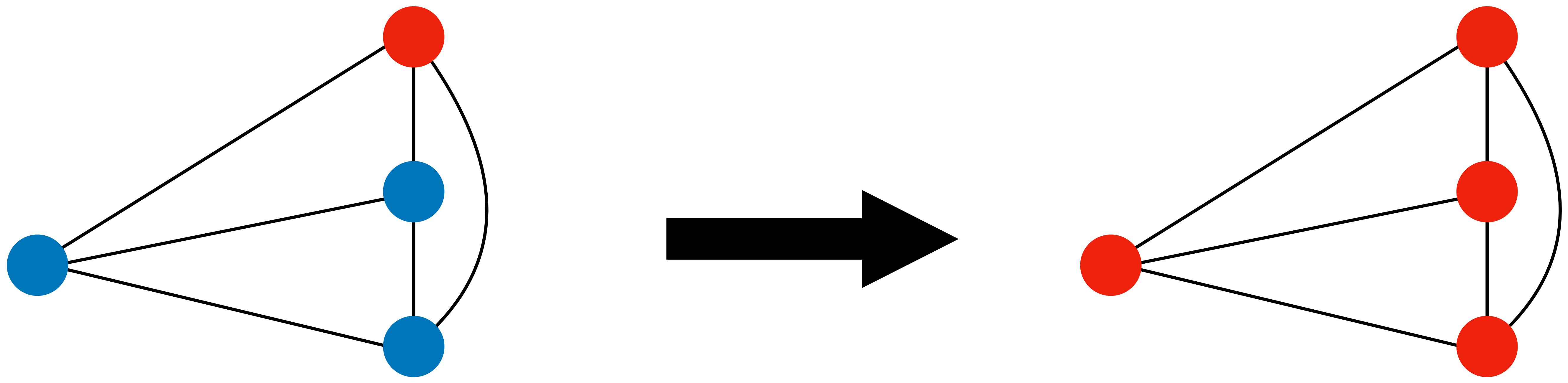


All 3 blue vertices have more than half their neighbors colored blue.

Vertex dependencies

Naively, flip all such vertices?

Problem: flipping a vertex may “kill” the gains from other vertices.



All 3 blue vertices have more than half
their neighbors colored blue.

Summary of FKL algorithm

[FKL'02] loses another $1/d$ factor to handle such dependencies.

Summary of FKL algorithm

[FKL'02] loses another $1/d$ factor to handle such dependencies.

In the end, the probability that each vertex can be flipped is $\Omega(1/d^3)$.

Summary of FKL algorithm

[FKL'02] loses another $1/d$ factor to handle such dependencies.

In the end, the probability that each vertex can be flipped is $\Omega(1/d^3)$.

$$0.878 + \frac{1}{|E|} \cdot \Omega(n/d^3) = 0.878 + \Omega(1/d^4).$$

Summary of FKL algorithm

[FKL'02] loses another $1/d$ factor to handle such dependencies.

In the end, the probability that each vertex can be flipped is $\Omega(1/d^3)$.

$$0.878 + \frac{1}{|E|} \cdot \Omega(n/d^3) = 0.878 + \Omega(1/d^4).$$

[Florén'16] handles the dependencies better to get $\Omega(1/d^3)$ improvement.

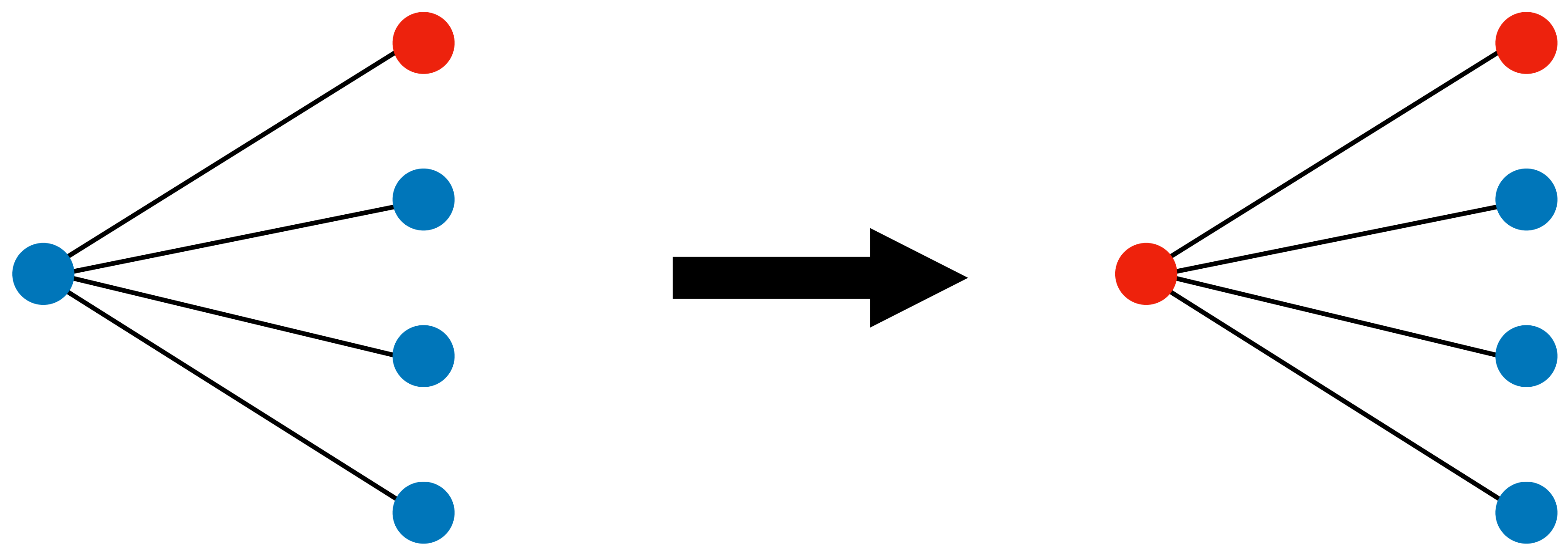
Our main idea

Our main idea

Main idea: flipping a vertex should gain more than 1 edge!

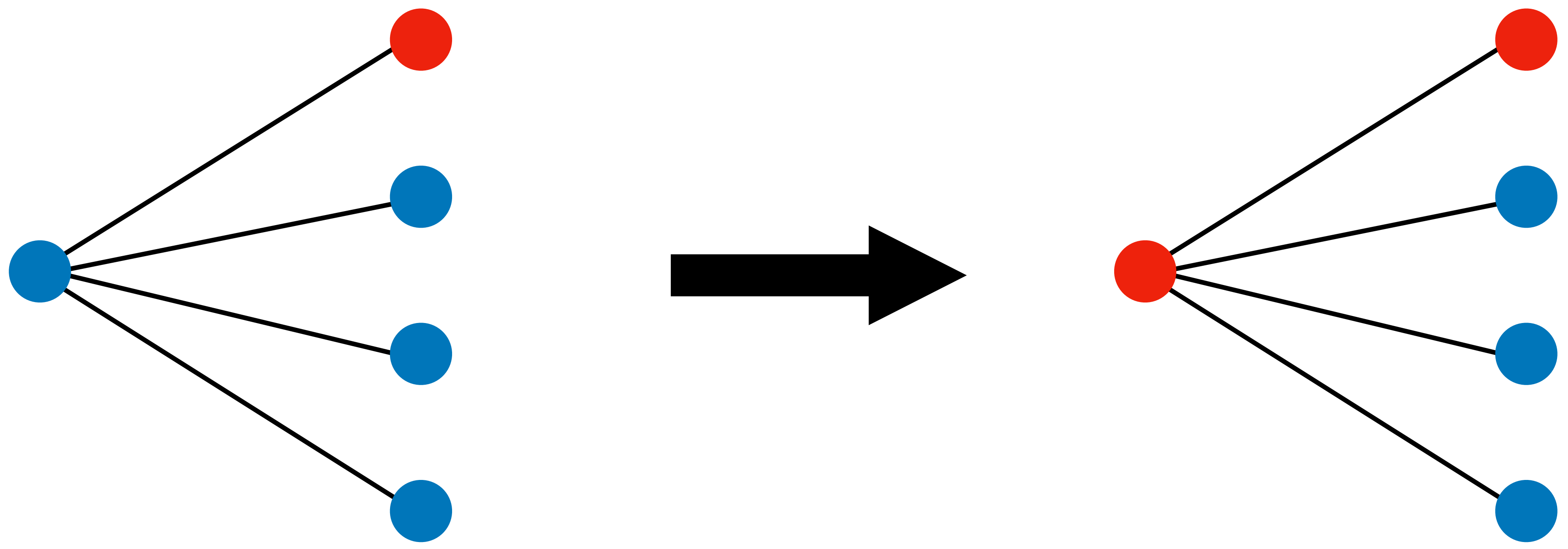
Our main idea

Main idea: flipping a vertex should gain more than 1 edge!



Our main idea

Main idea: flipping a vertex should gain more than 1 edge!

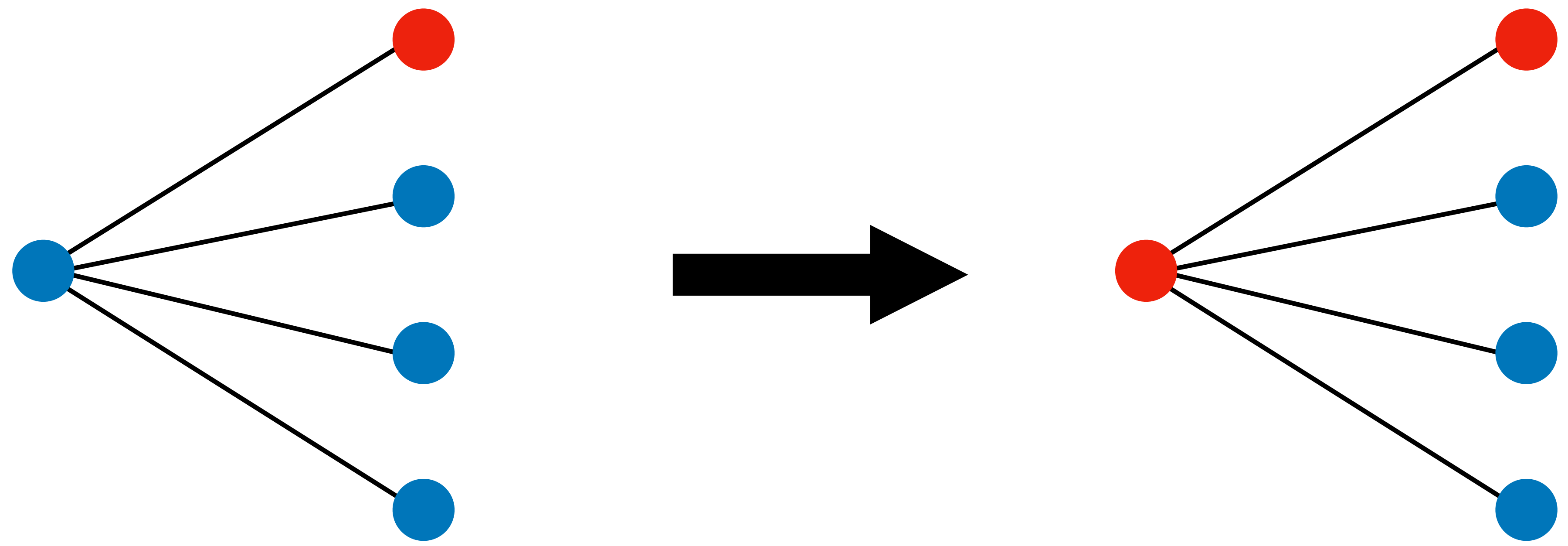


Lower bound the (expected) quantitative gain when flipping a vertex.

- Instead of the probability that a vertex is flipped.

Our main idea

Main idea: flipping a vertex should gain more than 1 edge!



Lower bound the (expected) quantitative gain when flipping a vertex.

- Instead of the probability that a vertex is flipped.

* Don't need to handle even/odd degrees.

Our algorithm

Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).

Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,

Our algorithm

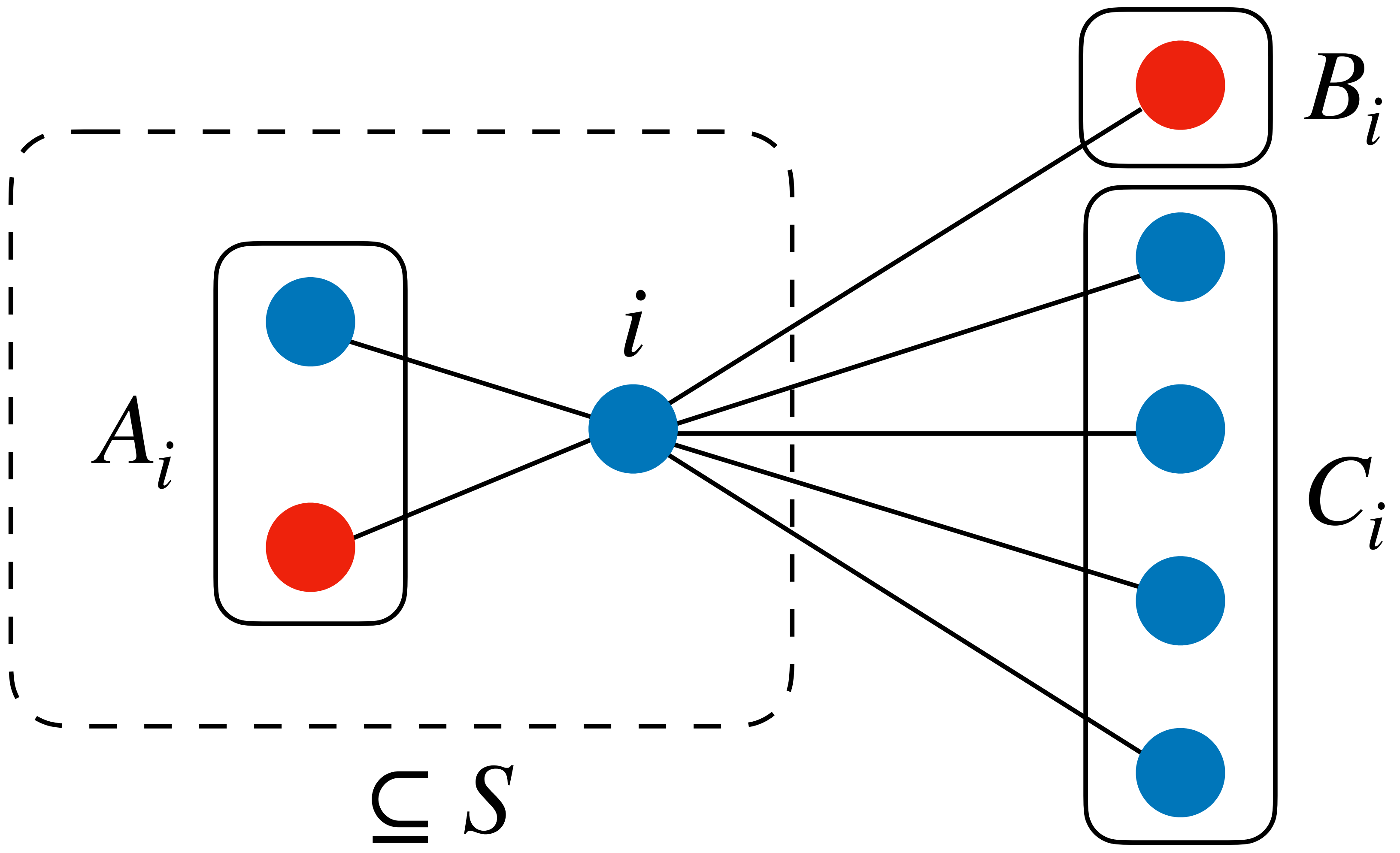
- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),

Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),
- $C_i = \{j \in N(i) : x_i \langle g, v_j \rangle \geq \epsilon\}$ (same sign as i).

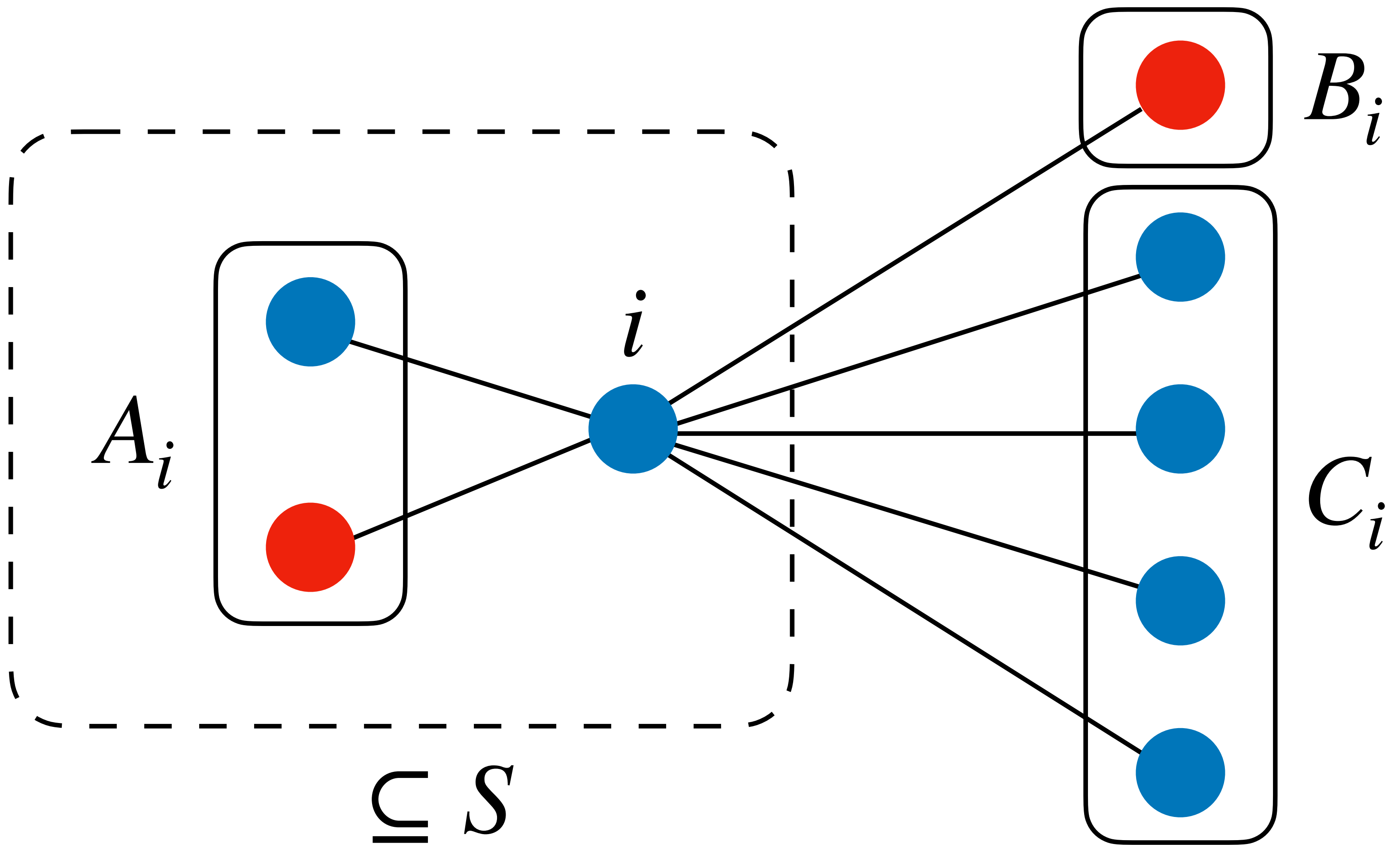
Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),
- $C_i = \{j \in N(i) : x_i \langle g, v_j \rangle \geq \epsilon\}$ (same sign as i).



Our algorithm

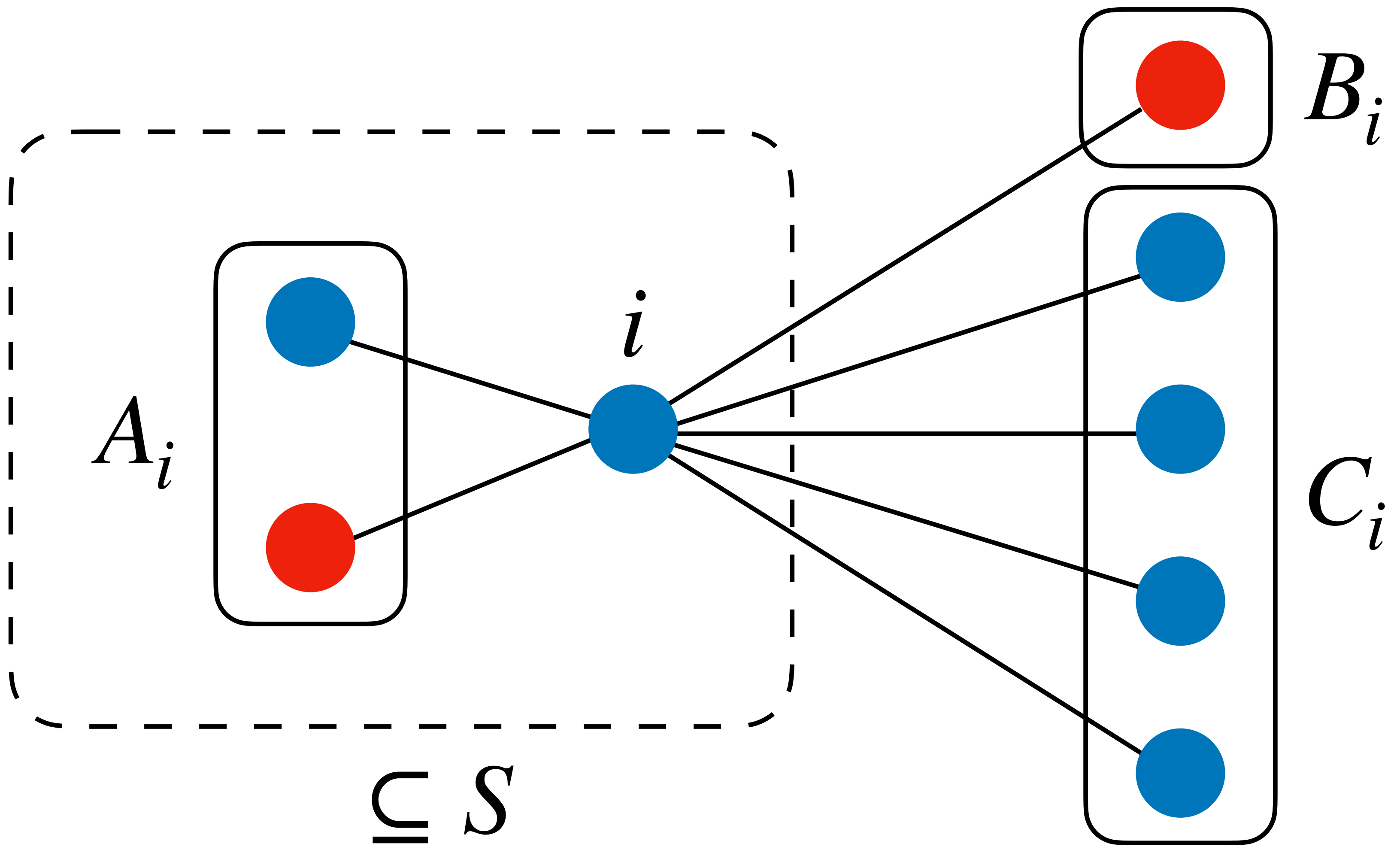
- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),
- $C_i = \{j \in N(i) : x_i \langle g, v_j \rangle \geq \epsilon\}$ (same sign as i).



Flip if $|C_i| > |A_i| + |B_i|$

Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),
- $C_i = \{j \in N(i) : x_i \langle g, v_j \rangle \geq \epsilon\}$ (same sign as i).

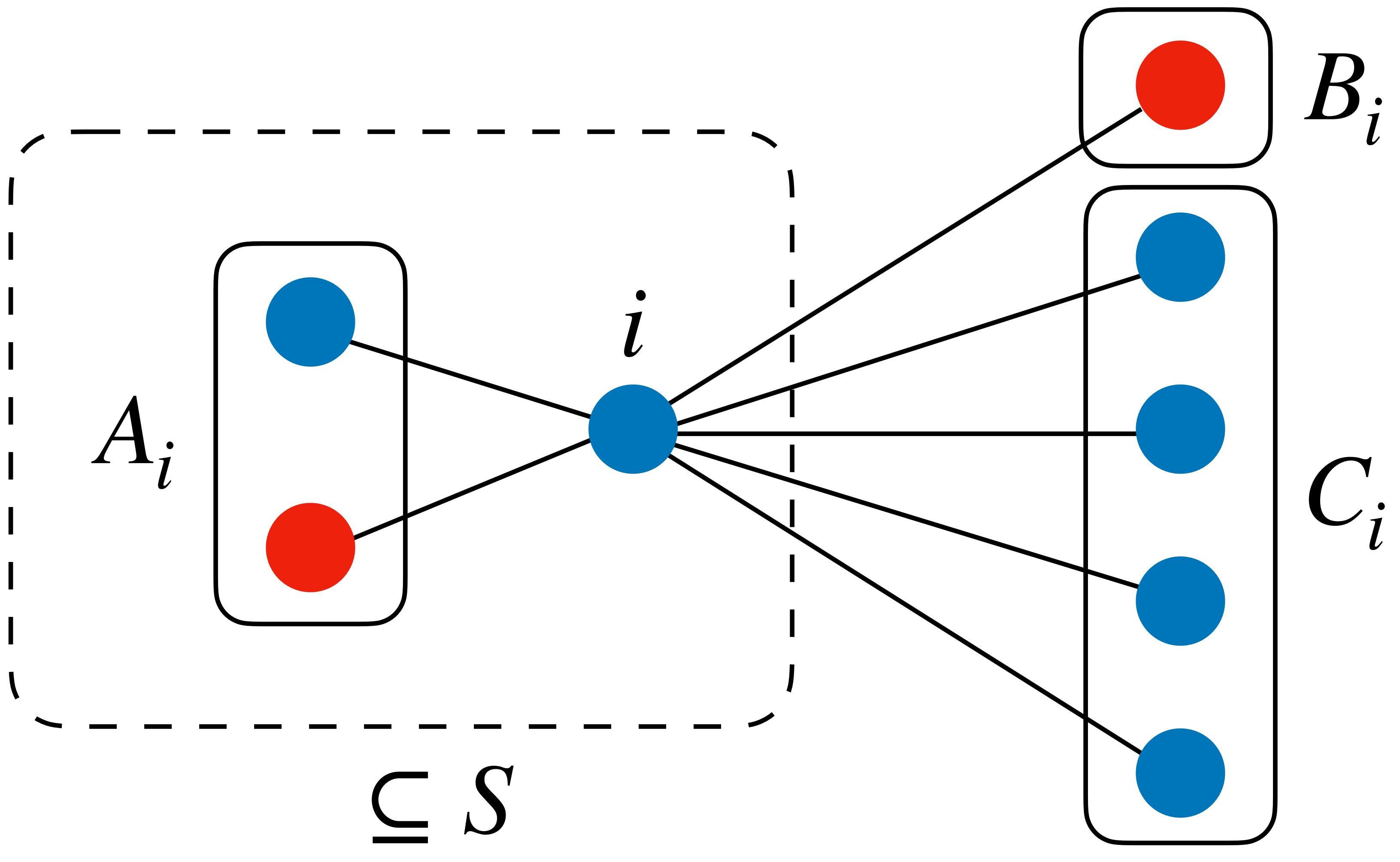


$$\text{Flip if } |C_i| > |A_i| + |B_i|$$

$$\text{Gain} \geq |C_i| - (|A_i| + |B_i|) = 2|C_i| - d$$

Our algorithm

- Candidate set $S = \{i \in V : \langle g, v_i \rangle \in (-\epsilon, \epsilon)\}$ (close to the hyperplane).
- $A_i = N(i) \cap S$,
- $B_i = \{j \in N(i) : x_i \langle g, v_j \rangle \leq -\epsilon\}$ (opposite sign as i),
- $C_i = \{j \in N(i) : x_i \langle g, v_j \rangle \geq \epsilon\}$ (same sign as i).

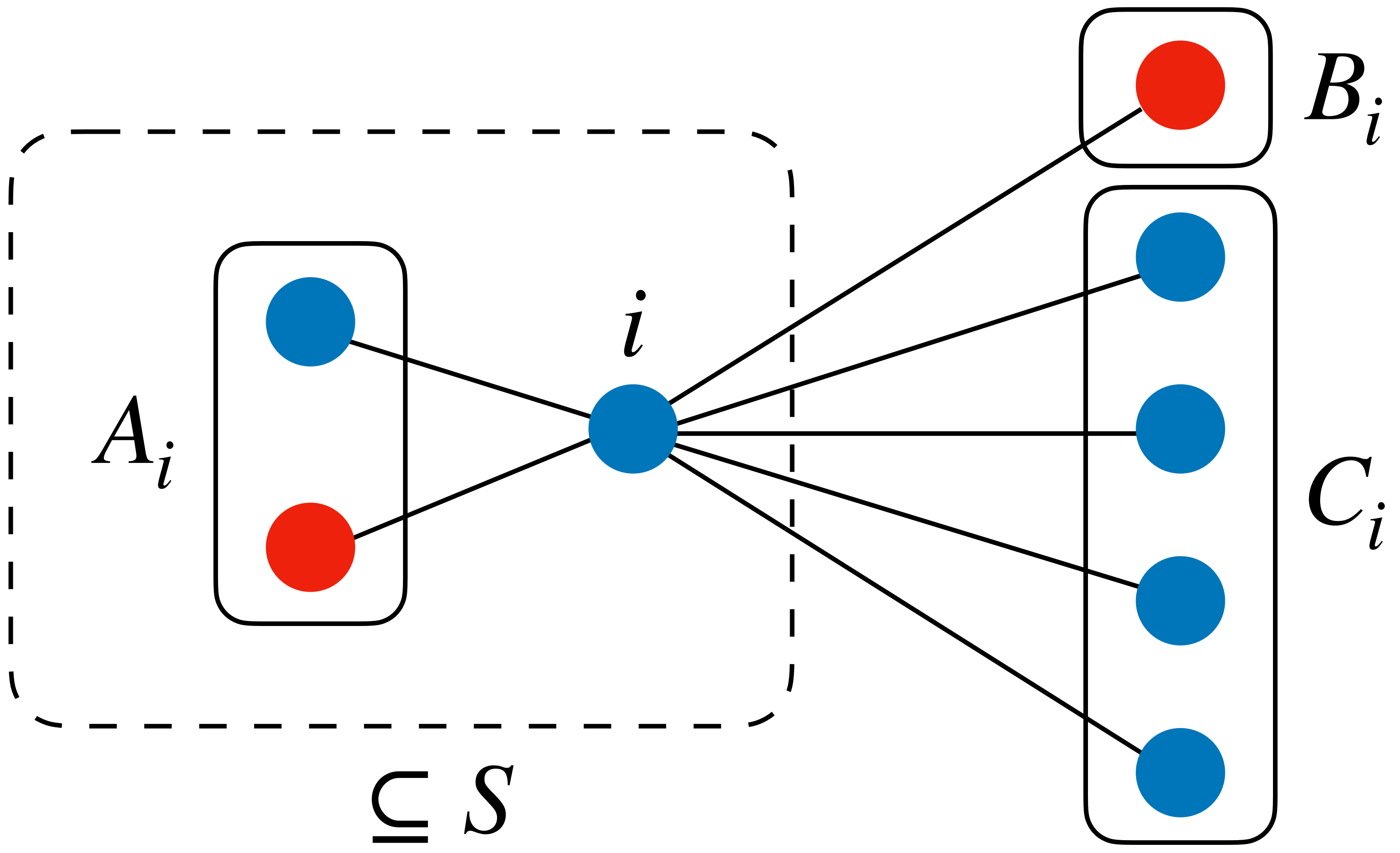


Flip if $|C_i| > |A_i| + |B_i|$

$$\text{Gain} \geq |C_i| - (|A_i| + |B_i|) = 2|C_i| - d$$

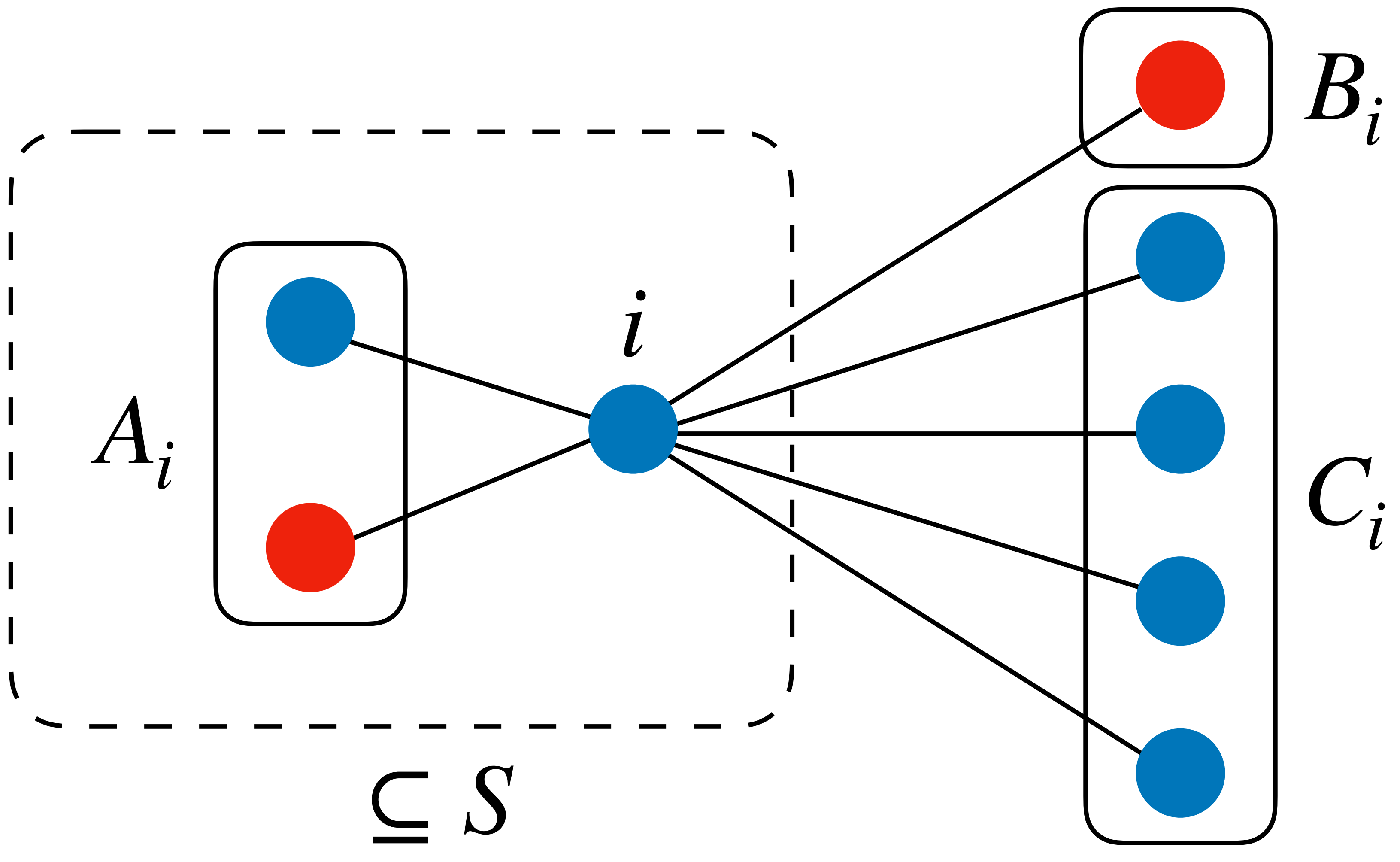
It doesn't matter how A_i is flipped in other iterations!

Analysis of our algorithm



Analysis of our algorithm

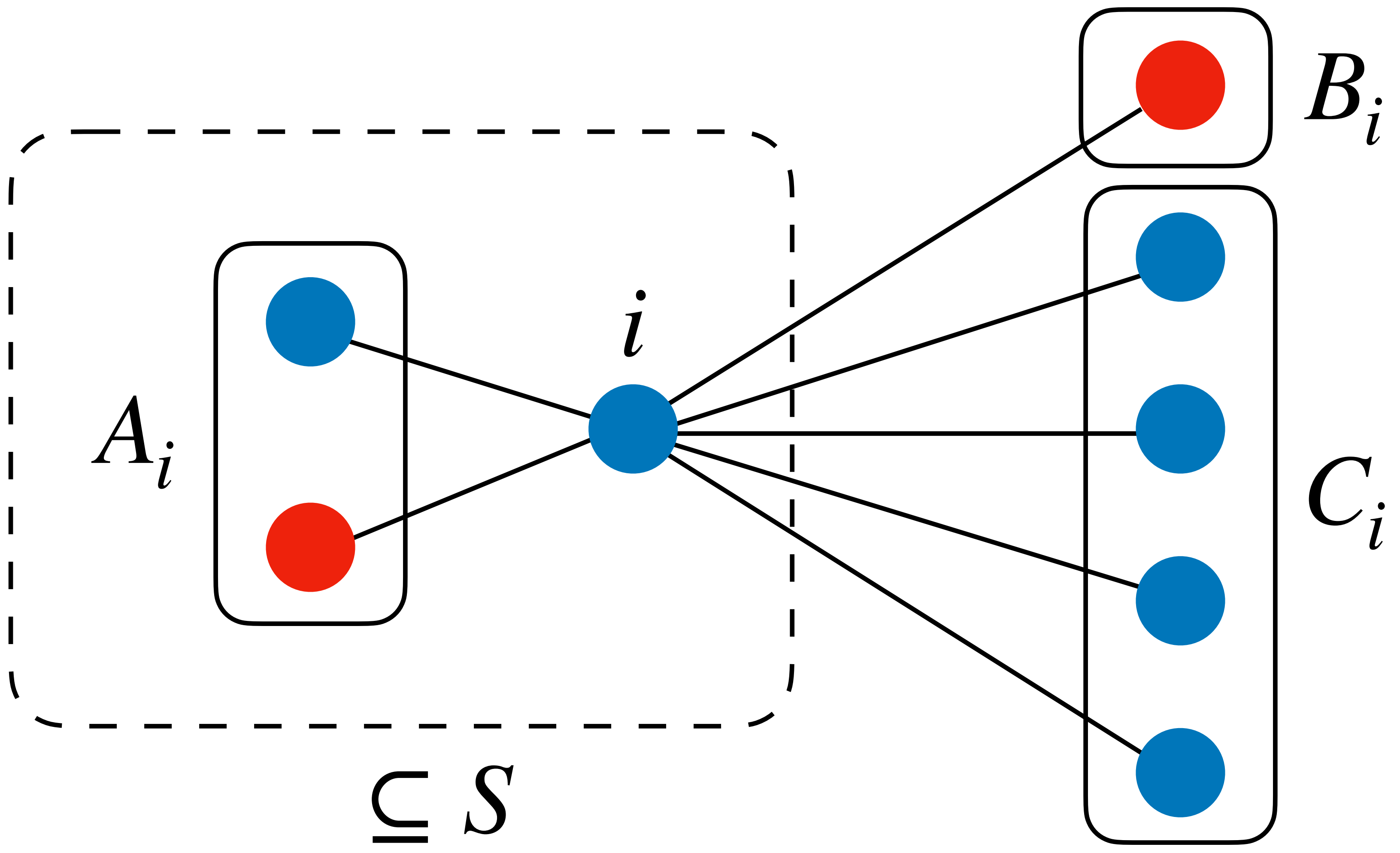
Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x, 0)$.



Analysis of our algorithm

Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x, 0)$.

Main Lemma. $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$.

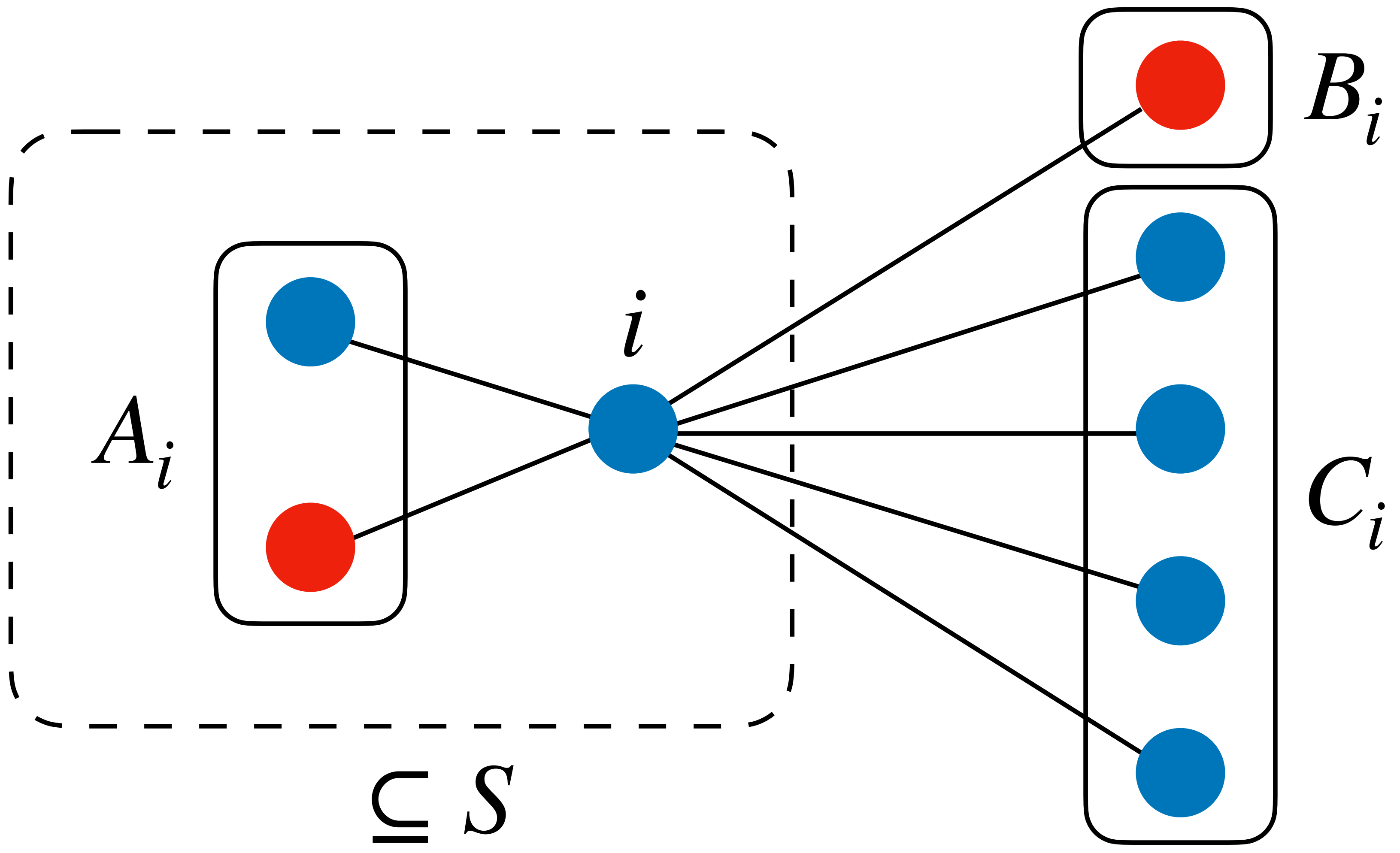


Analysis of our algorithm

Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x,0)$.

Main Lemma. $\mathbb{E}[\Delta_i | i \in S] \geq \tilde{\Omega}(1)$.

Note: $\Pr[i \in S] = \Theta(\epsilon)$. We will choose $\epsilon = \tilde{\Theta}(1/d)$.

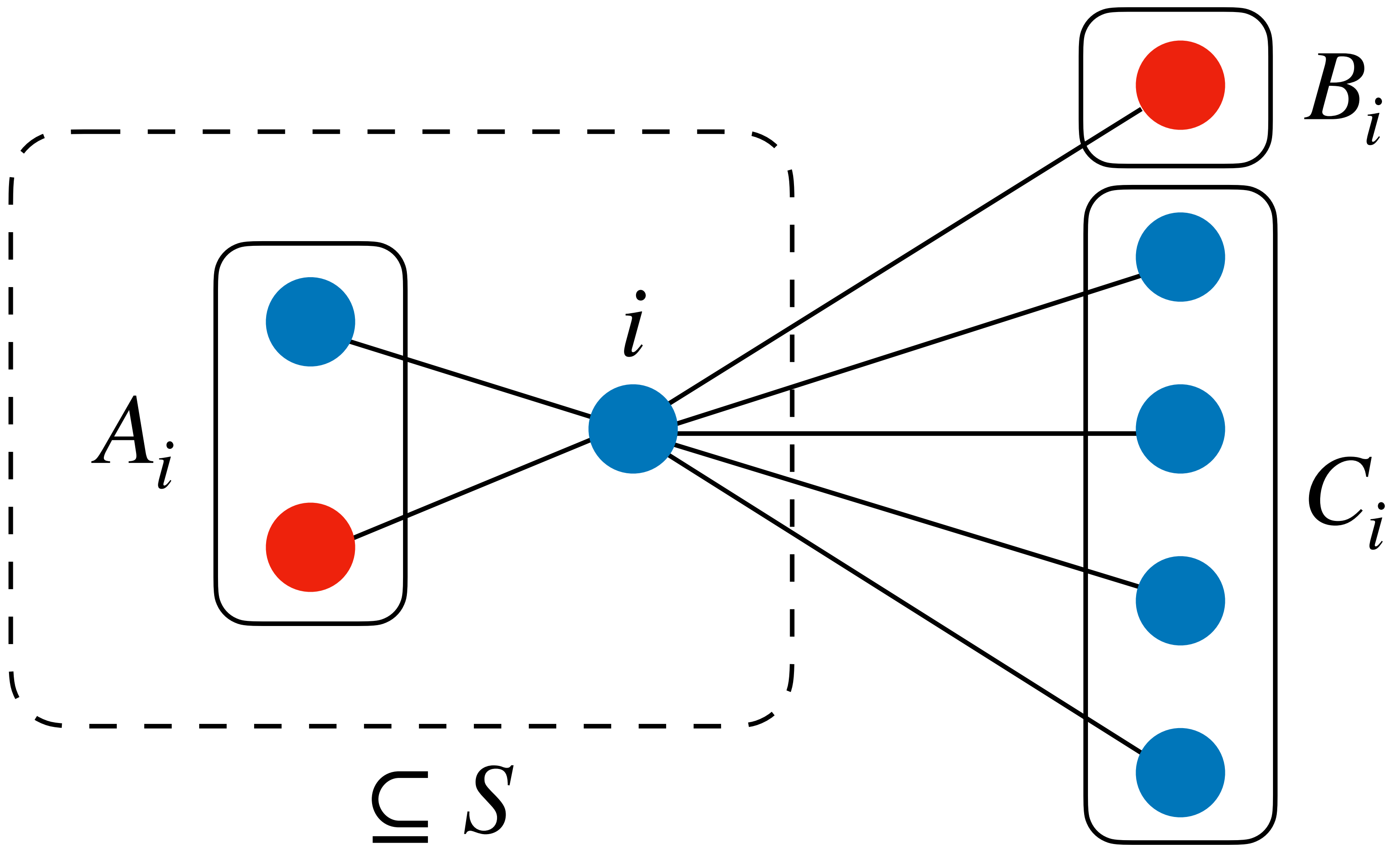


Analysis of our algorithm

Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x, 0)$.

Main Lemma. $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$.

Note: $\Pr[i \in S] = \Theta(\epsilon)$. We will choose $\epsilon = \tilde{\Theta}(1/d)$.



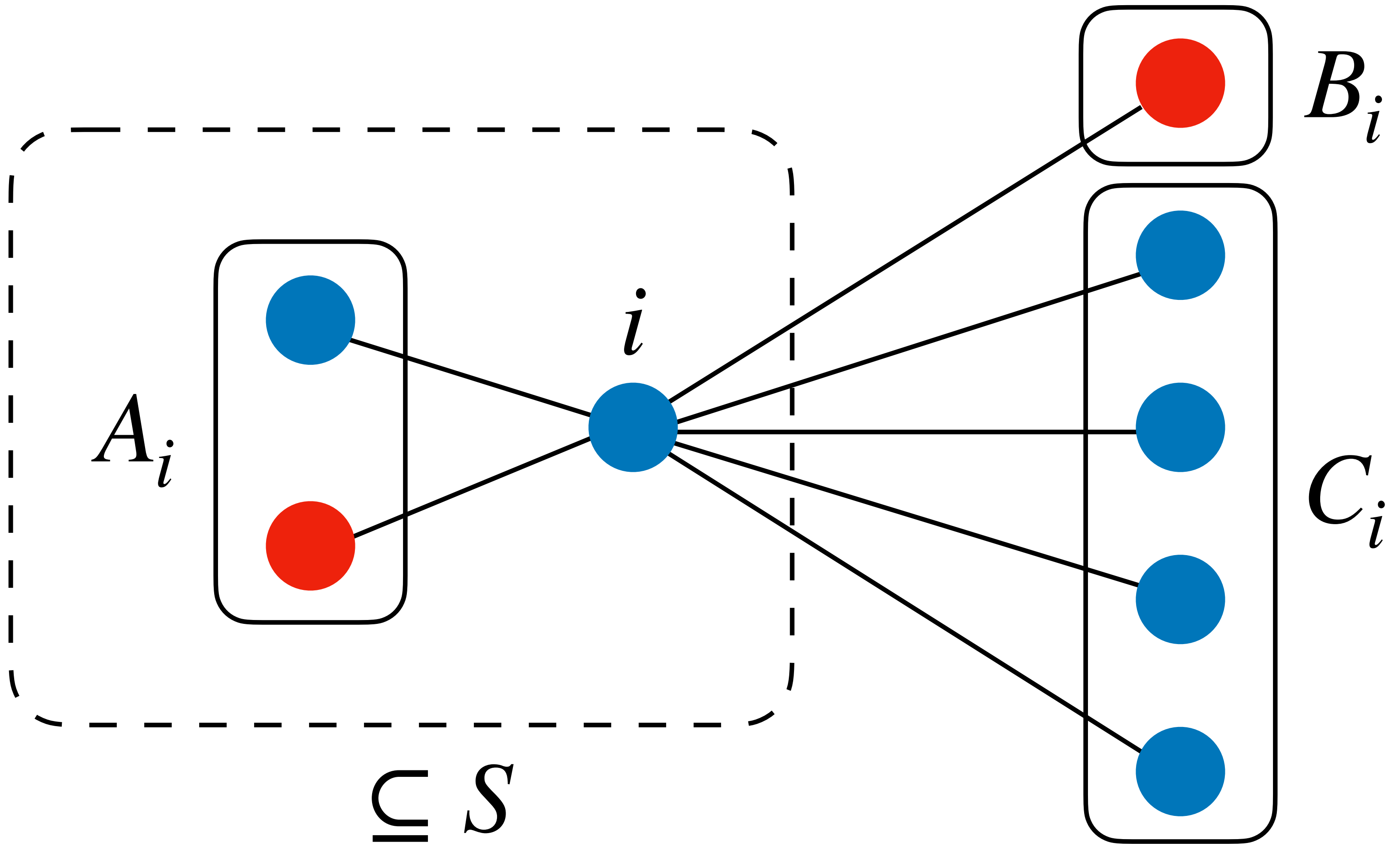
Proof of main result:

Analysis of our algorithm

Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x,0)$.

Main Lemma. $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$.

Note: $\Pr[i \in S] = \Theta(\epsilon)$. We will choose $\epsilon = \tilde{\Theta}(1/d)$.



Proof of main result:

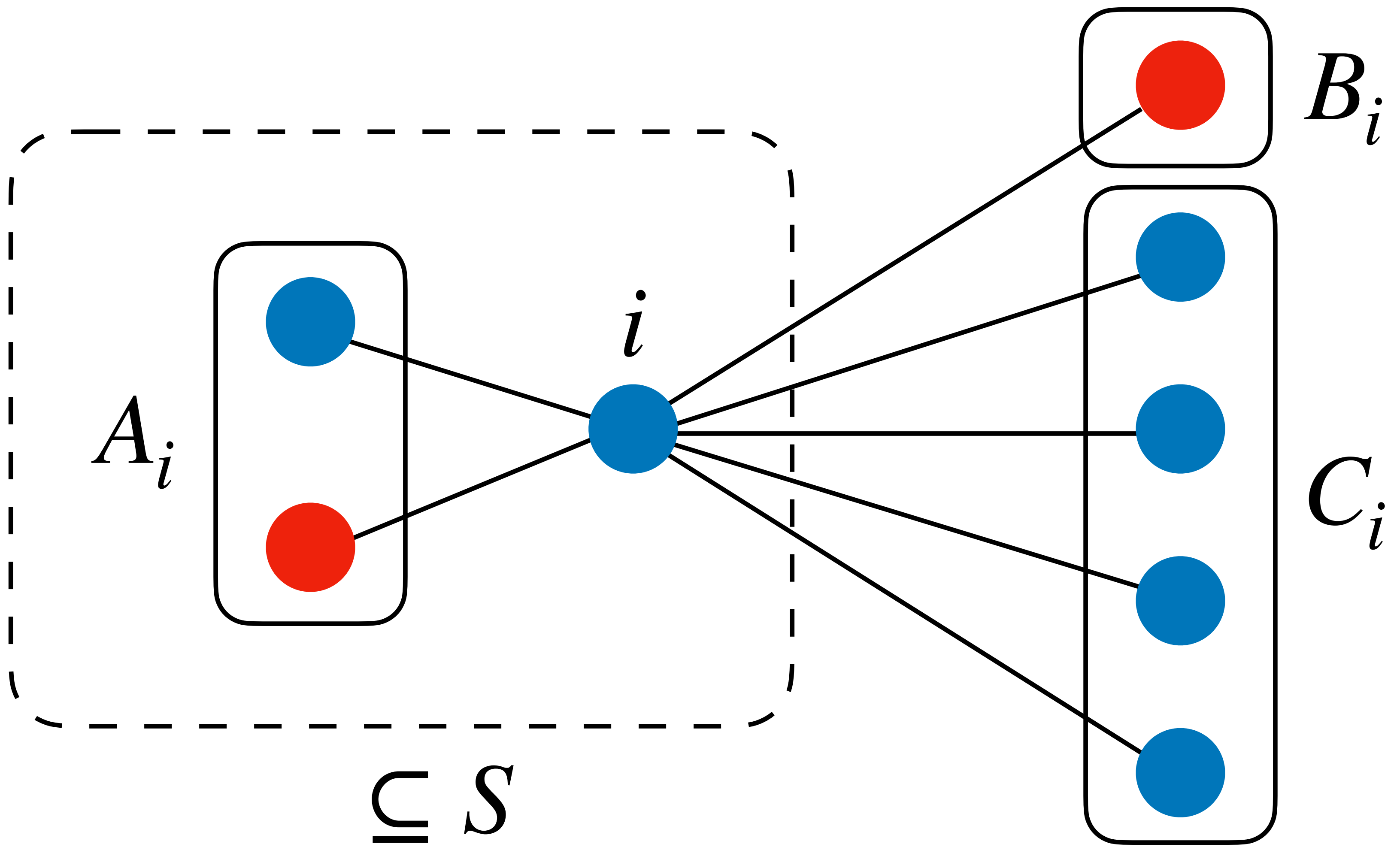
$$0.878 + \frac{1}{|E|} \cdot \sum_{i \in V} \mathbb{E}[\Delta_i \mid i \in S] \Pr[i \in S] = 0.878 + \frac{2}{nd} \cdot \tilde{\Omega}(1) \cdot \tilde{\Omega}(1/d)$$

Analysis of our algorithm

Denote the local gain: $\Delta_i = (2|C_i| - d)_+$ where $(x)_+ = \max(x, 0)$.

Main Lemma. $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$.

Note: $\Pr[i \in S] = \Theta(\epsilon)$. We will choose $\epsilon = \tilde{\Theta}(1/d)$.



Proof of main result:

$$0.878 + \frac{1}{|E|} \cdot \sum_{i \in V} \mathbb{E}[\Delta_i \mid i \in S] \Pr[i \in S] = 0.878 + \frac{2}{nd} \cdot \tilde{\Omega}(1) \cdot \tilde{\Omega}(1/d)$$

$$= 0.878 + \tilde{\Omega}(1/d^2). \qquad \blacksquare$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

$:= \hat{v}_j$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

$:= \hat{v}_j$

$$\langle g, v_j \rangle \approx -0.69 \cdot g_1 + 0.726 \cdot \langle g', \hat{v}_j \rangle$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

$:= \hat{v}_j$

$$\langle g, v_j \rangle \approx -0.69 \cdot g_1 + 0.726 \cdot \langle g', \hat{v}_j \rangle$$

$:= h_j$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1,\ 0,\ 0,\ \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

$:= \hat{v}_j$

$$\langle g, v_j \rangle \approx -0.69 \cdot g_1 + 0.726 \cdot \langle g', \hat{v}_j \rangle$$

$:= h_j$

$h \sim \mathcal{N}(0, \Sigma)$ where $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle$.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1, 0, 0, \ldots),$
- $v_j = (-0.69, *, *, \ldots)$ for $j \in N(i)$.

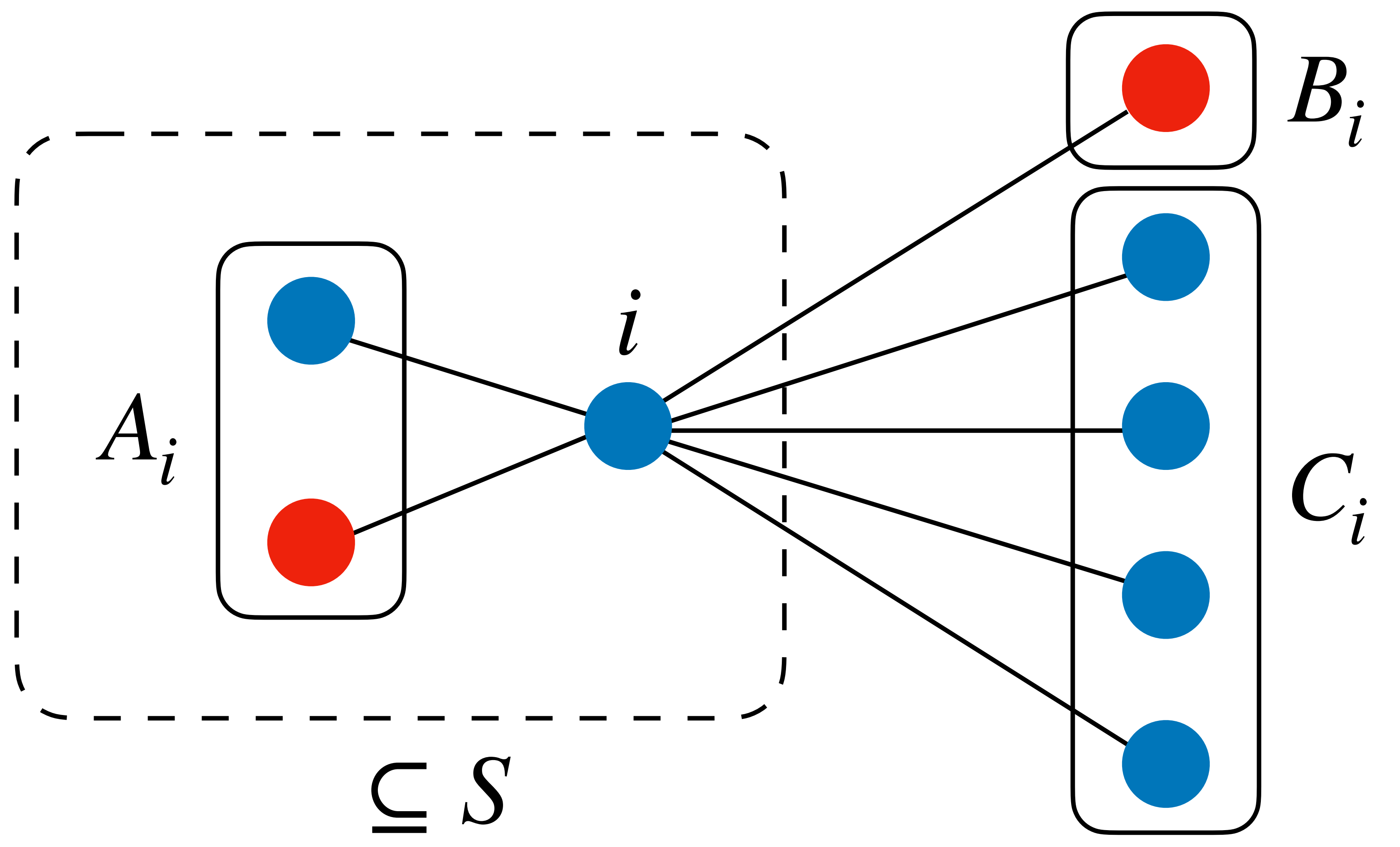
$:= \hat{v}_j$

$$\langle g, v_j \rangle \approx -0.69 \cdot g_1 + 0.726 \cdot \langle g', \hat{v}_j \rangle$$

$:= h_j$

$h \sim \mathcal{N}(0, \Sigma)$ where $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle$.

Turns out, $|C_i| \geq |\{j \in N(i) : h_j \leq -3\epsilon\}| =: Z$.



Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

- $v_i = (1, 0, 0, \dots),$
- $v_j = (-0.69, *, *, \dots)$ for $j \in N(i)$.

$:= \hat{v}_j$

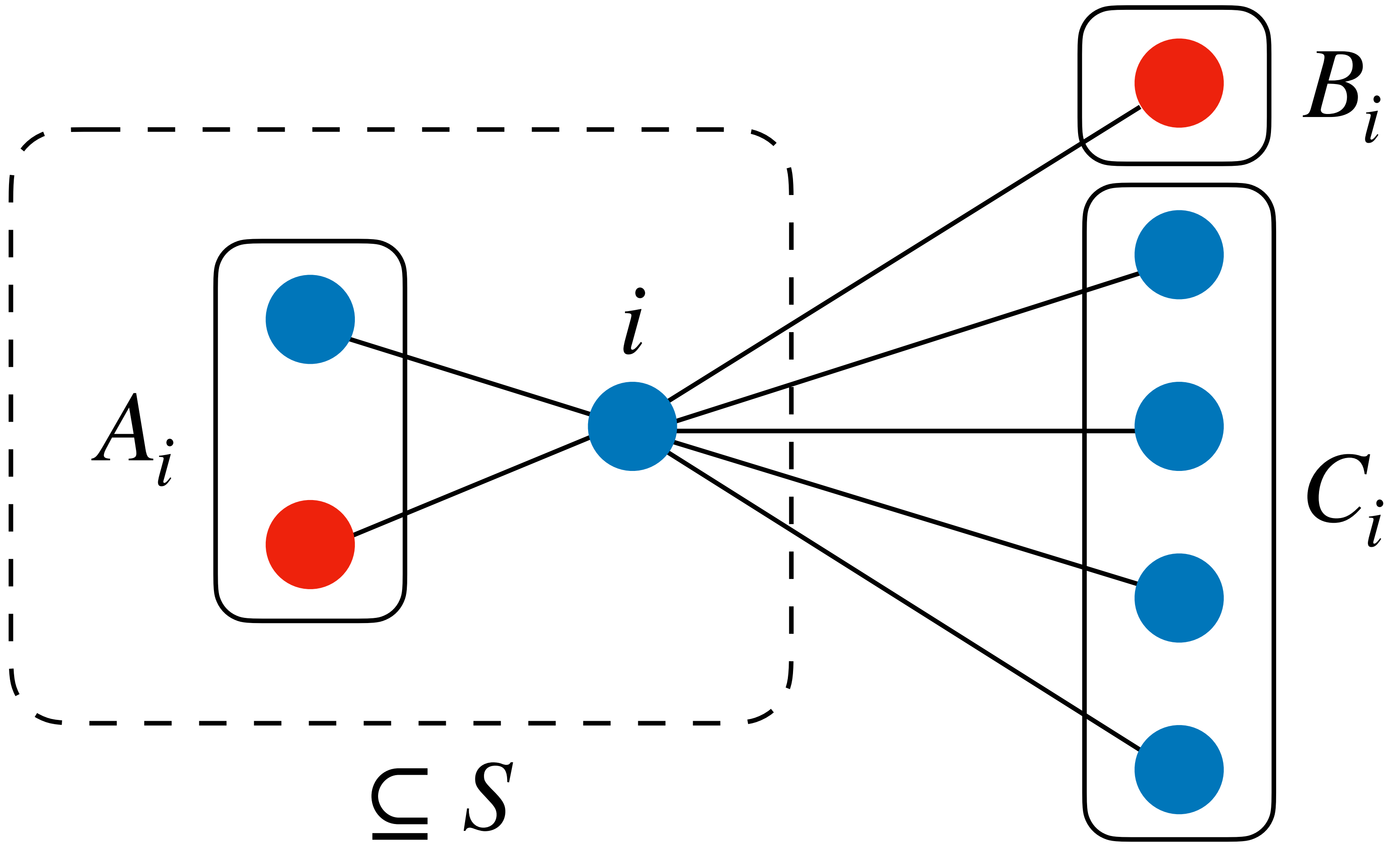
$$\langle g, v_j \rangle \approx -0.69 \cdot g_1 + 0.726 \cdot \langle g', \hat{v}_j \rangle$$

$:= h_j$

$h \sim \mathcal{N}(0, \Sigma)$ where $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle$.

Turns out, $|C_i| \geq |\{j \in N(i) : h_j \leq -3\epsilon\}| =: Z$.

So local gain $\Delta_i \geq (2Z - d)_+$.



Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

How do we lower bound $\mathbb{E}[(2Z - d)_+]$?

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

How do we lower bound $\mathbb{E}[(2Z - d)_+]$?

Idea: lower bound $\mathbb{E}[(2Z - d)^2]$!

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

How do we lower bound $\mathbb{E}[(2Z - d)_+]$?

Idea: lower bound $\mathbb{E}[(2Z - d)^2]$!

$$\mathbb{E}[(Z - d/2)^2] \geq \frac{1}{2\pi} \sum_{j,k \in [d]} \arcsin(\Sigma_{jk}) - O(\epsilon d)$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

How do we lower bound $\mathbb{E}[(2Z - d)_+]$?

Idea: lower bound $\mathbb{E}[(2Z - d)^2]$!

$$\mathbb{E}[(Z - d/2)^2] \geq \frac{1}{2\pi} \sum_{j,k \in [d]} \arcsin(\Sigma_{jk}) - O(\epsilon d)$$

$$\geq \frac{1}{2\pi} \cdot \vec{1}^\top \arcsin(\Sigma)\vec{1} - O(\epsilon d)$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Observe: $\Pr[h_j \leq -3\epsilon] = \frac{1}{2} - \Theta(\epsilon)$, so $\mathbb{E}[2Z - d] \approx -\Theta(\epsilon d)$.

How do we lower bound $\mathbb{E}[(2Z - d)_+]$?

Idea: lower bound $\mathbb{E}[(2Z - d)^2]$!

$$\mathbb{E}[(Z - d/2)^2] \geq \frac{1}{2\pi} \sum_{j,k \in [d]} \arcsin(\Sigma_{jk}) - O(\epsilon d)$$

$$\geq \frac{1}{2\pi} \cdot \vec{1}^\top \arcsin(\Sigma) \vec{1} - O(\epsilon d)$$

Applying arcsin entry-wise.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Fact: $\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq 0$.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

Fact: $\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq 0$.

Arcsin Lemma. If $\Sigma_{jj} = 1$ and $\Sigma_{jk} \geq -1/2$ for all $j, k \in [d]$, then

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

Fact: $\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq 0.$

Arcsin Lemma. If $\Sigma_{jj} = 1$ and $\Sigma_{jk} \geq -1/2$ for all $j, k \in [d]$, then

$$\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq \Omega\left(\frac{d}{\sqrt{\log d}}\right).$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

Fact: $\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq 0$.

Arcsin Lemma. If $\Sigma_{jj} = 1$ and $\Sigma_{jk} \geq -1/2$ for all $j, k \in [d]$, then

$$\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq \Omega\left(\frac{d}{\sqrt{\log d}}\right).$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

Fact: $\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq 0$.

??

Arcsin Lemma. If $\Sigma_{jj} = 1$ and $\Sigma_{jk} \geq -1/2$ for all $j, k \in [d]$, then

$$\vec{1}^\top \arcsin(\Sigma)\vec{1} \geq \Omega\left(\frac{d}{\sqrt{\log d}}\right).$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Add triangle inequality constraints to the SDP [Feige-Goemans'95, Feige-Schechtman'01]!

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Add triangle inequality constraints to the SDP [Feige-Goemans'95, Feige-Schechtman'01]!

$$\langle v_i, v_j \rangle + \langle v_i, v_k \rangle + \langle v_j, v_k \rangle \geq -1$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Add triangle inequality constraints to the SDP [Feige-Goemans'95, Feige-Schechtman'01]!

$$\langle v_i, v_j \rangle + \langle v_i, v_k \rangle + \langle v_j, v_k \rangle \geq -1$$

$$\langle v_i, v_j \rangle - \langle v_i, v_k \rangle - \langle v_j, v_k \rangle \geq -1$$

Bounding local gain

Show: $\mathbb{E}[\Delta_i \,|\, i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Add triangle inequality constraints to the SDP [Feige-Goemans'95, Feige-Schechtman'01]!

$$\langle v_i, v_j \rangle + \langle v_i, v_k \rangle + \langle v_j, v_k \rangle \geq -1$$

$$\langle v_i, v_j \rangle - \langle v_i, v_k \rangle - \langle v_j, v_k \rangle \geq -1$$

Valid constraints: for any $x_i, x_j, x_k \in \{\pm 1\}$, two of them must be equal.

Bounding local gain

Show: $\mathbb{E}[\Delta_i \mid i \in S] \geq \tilde{\Omega}(1)$

How do we guarantee that $\Sigma_{jk} = \langle \hat{v}_j, \hat{v}_k \rangle \geq -1/2$?

Add triangle inequality constraints to the SDP [Feige-Goemans'95, Feige-Schechtman'01]!

$$\langle v_i, v_j \rangle + \langle v_i, v_k \rangle + \langle v_j, v_k \rangle \geq -1$$

$$\langle v_i, v_j \rangle - \langle v_i, v_k \rangle - \langle v_j, v_k \rangle \geq -1$$

Valid constraints: for any $x_i, x_j, x_k \in \{\pm 1\}$, two of them must be equal.

Note: triangle inequality also crucial in [FKL'02, Florén'16].

Bounding local gain

Simple calculation: triangle inequality $\Longrightarrow \Sigma_{jk} \geq -0.2$.

Bounding local gain

Simple calculation: triangle inequality $\implies \Sigma_{jk} \geq -0.2$.

So, Arcsin Lemma $\implies \mathbb{E}[(2Z - d)^2] \geq \tilde{\Omega}(d)$.

Bounding local gain

Simple calculation: triangle inequality $\implies \Sigma_{jk} \geq -0.2$.

So, Arcsin Lemma $\implies \mathbb{E}[(2Z-d)^2] \geq \tilde{\Omega}(d)$.

Combined with $\mathbb{E}[2Z-d] = -\Theta(\epsilon d)$, we get

Bounding local gain

Simple calculation: triangle inequality $\implies \Sigma_{jk} \geq -0.2$.

So, Arcsin Lemma $\implies \mathbb{E}[(2Z-d)^2] \geq \tilde{\Omega}(d)$.

Combined with $\mathbb{E}[2Z-d] = -\Theta(\epsilon d)$, we get

$$\mathbb{E}[(2Z-d)_+] \geq \tilde{\Omega}(1) - O(\epsilon d).$$

Bounding local gain

Simple calculation: triangle inequality $\implies \Sigma_{jk} \geq -0.2$.

So, Arcsin Lemma $\implies \mathbb{E}[(2Z-d)^2] \geq \tilde{\Omega}(d)$.

Combined with $\mathbb{E}[2Z-d] = -\Theta(\epsilon d)$, we get

$$\mathbb{E}[(2Z-d)_+] \geq \tilde{\Omega}(1) - O(\epsilon d).$$

Setting $\epsilon = \tilde{\Theta}(1/d)$ completes the proof.

Conclusion

Hardness: Trevisan [2001] showed that $0.878 + O(1/\sqrt{d})$ is NP-hard.

Open question: improve the algorithm or the hardness?

Conclusion

Hardness: Trevisan [2001] showed that 0.878 + $O(1/\sqrt{d})$ is NP-hard.

Open question: improve the algorithm or the hardness?

Thank you!