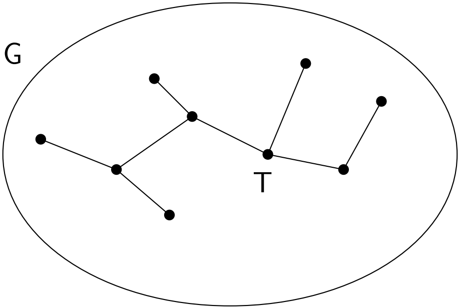
# Nearly optimal embeddings of trees

*Benny Sudakov*
*UCLA and IAS*

*Jan Vondrák*
*Princeton University*

**Question:**

Given a graph $G$, what trees $T$ can be embedded in $G$?




*Goal:* Find sufficient conditions on $G$ in order to contain *all trees* from a certain family.

# Basic facts

**Folklore result**

Any graph $G$ of minimum degree $d$ contains all trees with $d$ edges.

# Basic facts

**Folklore result**

Any graph $G$ of minimum degree $d$ contains all trees with $d$ edges.

This is obviously tight ($G$ is a clique of size $d+1$). So, embedding trees of size $|T| > d$ requires some additional assumptions...

**Obvious restrictions**

- $|T| \leq |G|$.
- Degrees in $T \leq$ degrees in $G$.

**Meta-result**

In suitable classes of graphs, trees can be embedded up to trivial bounds on size and degrees.

# Natural Classes of Graphs

**Meta-result**

In suitable classes of graphs, trees can be embedded up to trivial bounds on size and degrees.

Examples:

1. Graphs of *girth* $g$: not containing any cycle shorter than $g$.
2. $H$-free graphs: not containing a bipartite subgraph $H$.
3. Expanding graphs: any "sufficiently small" set of vertices $X$, has many neighbors outside of $X$.
4. Random graphs.

These graphs typically have order $n$ much larger than min degree $d$; e.g., for girth $2k+1$, the number of vertices must be $n = \Omega(d^k)$.

# The Erdös-Sós conjecture

**Erdös-Sós Conjecture**

Any graph $G$ of **average** degree $d$ contains all trees with $d$ edges.

- *Brandt-Dobson, Haxell-Łuczak, Jiang '01:* Any graph of girth $2k+1$ and minimum degree $d$ contains all trees with $kd$ edges and maximum degree $\leq d$.
- *Ajtai-Komlós-Simonovits-Szemerédi: (unpublished)* For sufficiently large $d$, the Erdös-Sós conjecture is true: any graph of average degree $d$ contains all trees of size at most $d$.

# The Erdös-Sós conjecture

**Erdös-Sós Conjecture**

Any graph $G$ of **average** degree $d$ contains all trees with $d$ edges.

- *Brandt-Dobson, Haxell-Łuczak, Jiang '01:* Any graph of girth $2k+1$ and minimum degree $d$ contains all trees with $kd$ edges and maximum degree $\leq d$.
- *Ajtai-Komlós-Simonovits-Szemerédi: (unpublished)* For sufficiently large $d$, the Erdös-Sós conjecture is true: any graph of average degree $d$ contains all trees of size at most $d$.

**Note**

High girth helps. Can we embed even larger trees in such graphs?

# Tree embeddings in expanding graphs

**Definition**

$$N_G(X) = \{v \in V(G) : \text{there is } u \in X \text{ adjacent to } v\}$$

- *Pósa '76, Friedman-Pippenger '87:* If $|N_G(X)| \geq (d+1)|X|$ for all $X \subset V(G)$, $|X| \leq 2t-2$, then $G$ contains all trees of size $t$ and maximum degree $\leq d$.
- *Benjamini, Schramm '97:* Any infinite graph with a positive Cheeger constant $h(G) = \inf_X \frac{|N(X) \setminus X|}{|X|}$ contains an infinite tree with positive Cheeger constant.

# Tree embeddings in expanding graphs

**Definition**

$$N_G(X) = \{v \in V(G) : \text{there is } u \in X \text{ adjacent to } v\}$$

- *Pósa '76, Friedman-Pippenger '87:* If $|N_G(X)| \geq (d+1)|X|$ for all $X \subset V(G)$, $|X| \leq 2t-2$, then $G$ contains all trees of size $t$ and maximum degree $\leq d$.
- *Benjamini, Schramm '97:* Any infinite graph with a positive Cheeger constant $h(G) = \inf_X \frac{|N(X) \setminus X|}{|X|}$ contains an infinite tree with positive Cheeger constant.

**Note**

Since $H$-free graphs (for bipartite $H$) are locally expanding, this gives a tree-embedding result for any class of $H$-free graphs. E.g., graphs of girth $2k+1$ contain all trees of size $O(d^{k-1})$. Is this the best we can do? Graphs of girth $k$ must have size $\Omega(d^k)$.

# Tree embeddings in random graphs

**Definition**

$G_{n,p}$ contains each possible edge independently with probability $p$.

- *Ajtai-Komlós-Szemerédi, de la Vega '79:* A random graph $G_{n,d/n}$ contains with high probability (w.h.p.) a path of length $c(d)n$ where $\lim_{d\to\infty} c(d) = 1$.
- *de la Vega '88:* For any tree $T$ of size $c_1 n$ and maximum degree $\Delta \leq c_2 d$, $G_{n,d/n}$ contains $T$ w.h.p.
- *Alon-Krivelevich-Sudakov '07:* $G_{n,d/n}$ contains **all trees** of size $(1-\epsilon)n$ and maximum degree $\Delta = \tilde{O}(d^{1/3})$ w.h.p.

# Tree embeddings in random graphs

**Definition**

$G_{n,p}$ contains each possible edge independently with probability $p$.

- *Ajtai-Komlós-Szemerédi, de la Vega '79:* A random graph $G_{n,d/n}$ contains with high probability (w.h.p.) a path of length $c(d)n$ where $\lim_{d\to\infty} c(d) = 1$.
- *de la Vega '88:* For any tree $T$ of size $c_1 n$ and maximum degree $\Delta \le c_2 d$, $G_{n,d/n}$ contains $T$ w.h.p.
- *Alon-Krivelevich-Sudakov '07:* $G_{n,d/n}$ contains **all trees** of size $(1-\epsilon)n$ and maximum degree $\Delta = \tilde{O}(d^{1/3})$ w.h.p.

**Question**

Does $G_{n,d/n}$ contain large trees with degrees proportional to $d$?

# Our results - graphs of high girth

**Theorem 1**

Let $\epsilon < \frac{1}{k}$, $d$ sufficiently large. Any graph of girth $2k+1$ and min degree $d$ contains all trees of size $\frac{\epsilon}{10}d^k$ and max degree $\leq (1-\epsilon)d$.

# Our results - graphs of high girth

**Theorem 1**

Let $\epsilon < \frac{1}{k}$, $d$ sufficiently large. Any graph of girth $2k+1$ and min degree $d$ contains all trees of size $\frac{\epsilon}{10} d^k$ and max degree $\leq (1-\epsilon)d$.

*Remarks:*

- From Friedman-Pippenger, we get trees of size $O(d^{k-1})$.
- In particular, for $C_4$-free graphs, it gives trees of size $O(d)$, which is trivial. We can embed trees of size $|T| = O(d^2)$, which might be the size of $G$ (projective plane).
- Jiang proves that $G$ contains all trees of max degree $\leq d$ and size $\leq kd$. If we strengthen the max degree condition slightly, to $(1-\epsilon)d$, we can embed trees of size $\epsilon d^k$.

# Our results - $K_{s,t}$-free graphs

**Theorem 2**

Let $s \geq t \geq 2$. Any $K_{s,t}$-free graph of min degree $d$ contains all trees of size $cd^{1+1/(t-1)}$ and max degree $\leq \frac{1}{256}d$.

# Our results - $K_{s,t}$-free graphs

**Theorem 2**

Let $s \geq t \geq 2$. Any $K_{s,t}$-free graph of min degree $d$ contains all trees of size $cd^{1+1/(t-1)}$ and max degree $\leq \frac{1}{256}d$.

*Remarks:*

- From Friedman-Pippenger, we do not get any non-trivial result, since subsets of size $\Omega(d)$ do not expand enough.
- Since there are $K_{s,t}$-free graphs with minimum degree $d$ and $O(d^{1+1/(t-1)})$ vertices (known examples for $s > (t-1)!$), one cannot aspire to embed trees of larger size.

# Our results - random graphs

**Theorem 3**

Let $d \geq n^{\epsilon}$ for some constant $\epsilon > 0$. Then the random graph $G_{n,d/n}$ contains w.h.p. all trees of size $\frac{1}{16}\epsilon n$ and max degree $\leq \epsilon d$.

**Theorem 3**

Let $d \geq n^{\epsilon}$ for some constant $\epsilon > 0$. Then the random graph $G_{n,d/n}$ contains w.h.p. all trees of size $\frac{1}{16}\epsilon n$ and max degree $\leq \epsilon d$.

*Remarks:*

- For every fixed tree $T$ of size $O(n)$ and max degree $O(d)$, it was proved by De la Vega that $T \subset G_{n,d/n}$ w.h.p. However, it is much harder to prove that $G_{n,p}$ contains *all trees* w.h.p.
- Simultaneous embedding was known for trees of size $(1-\epsilon)n$ and degree $\tilde{O}(d^{1/3})$ [Alon-Krivelevich-Sudakov]. We improve the degree bound to $O(d)$, at the cost of a constant factor in the size of $T$.

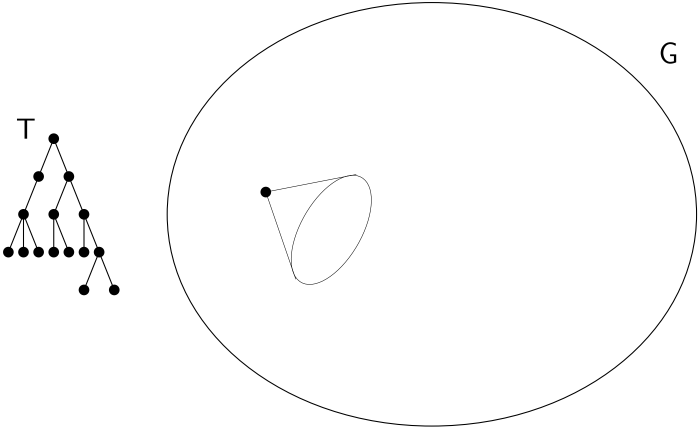
# The Algorithm

**Self-avoiding tree-indexed random walk**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.

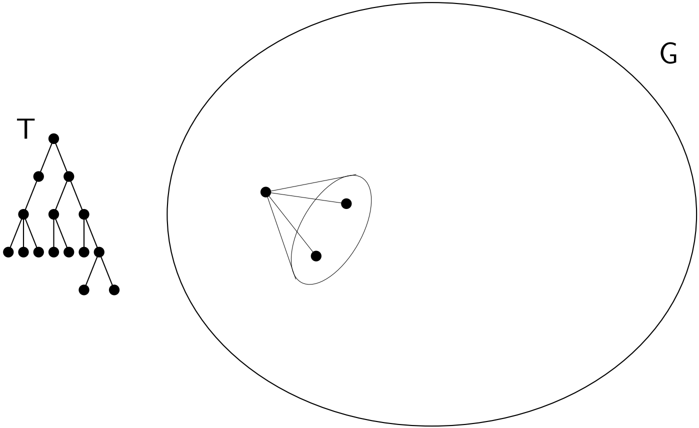
**Self-avoiding tree-indexed random walk**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



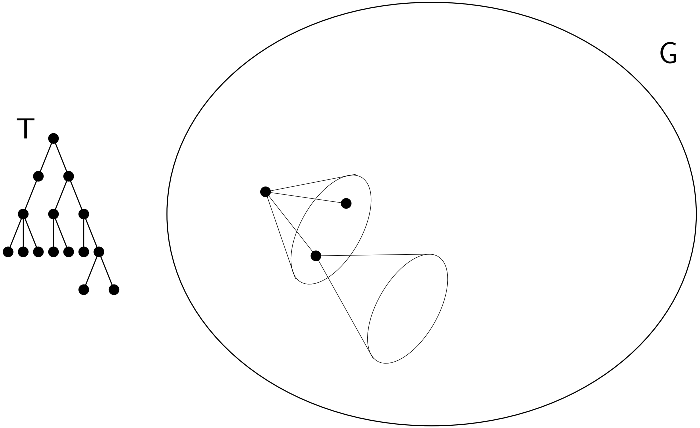
**Self-avoiding tree-indexed random walk**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



### SELF-AVOIDING TREE-INDEXED RANDOM WALK

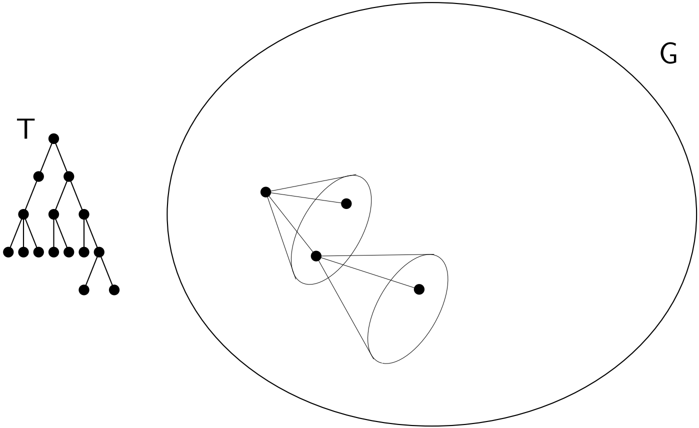
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



# The Algorithm

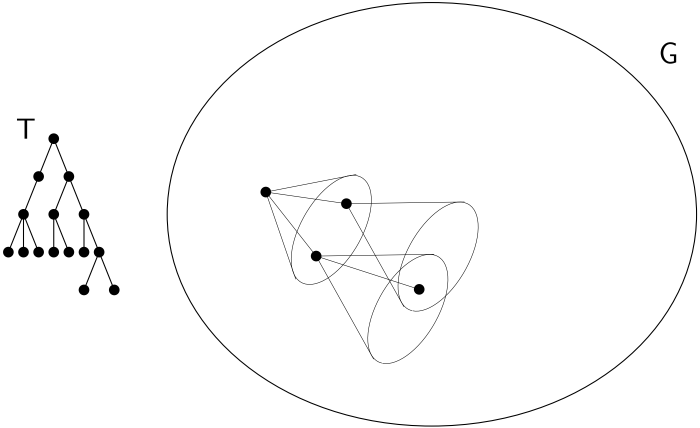
**Self-avoiding tree-indexed random walk**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



### Self-avoiding tree-indexed random walk

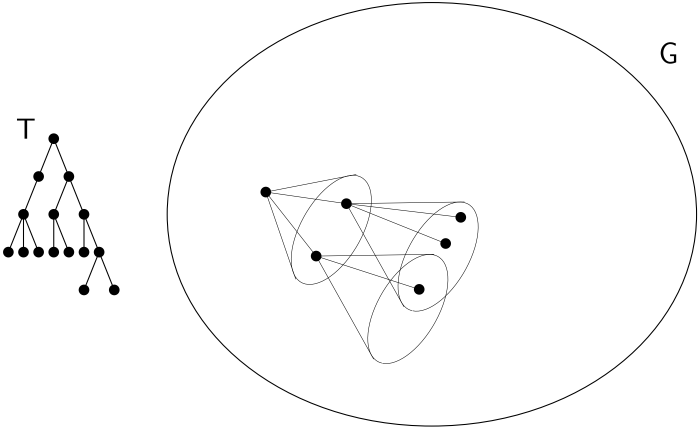
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



# The Algorithm

## Self-avoiding tree-indexed random walk

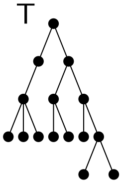
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.

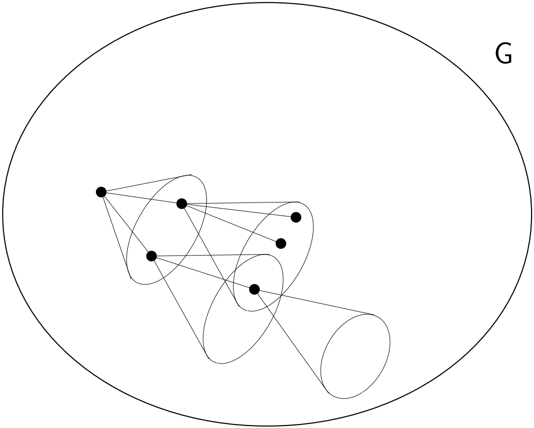


# The Algorithm

**Self-avoiding tree-indexed random walk**

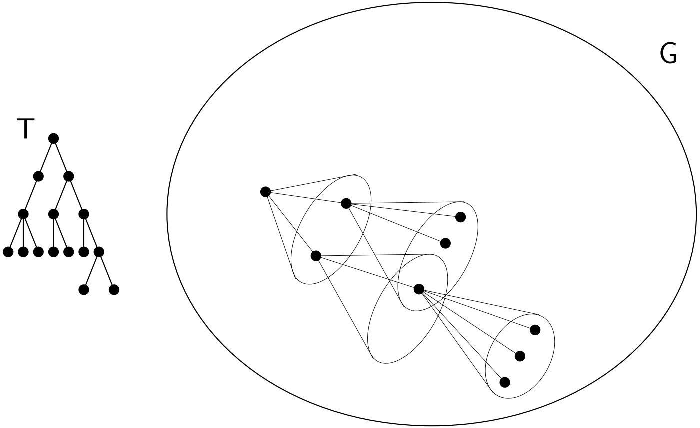
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



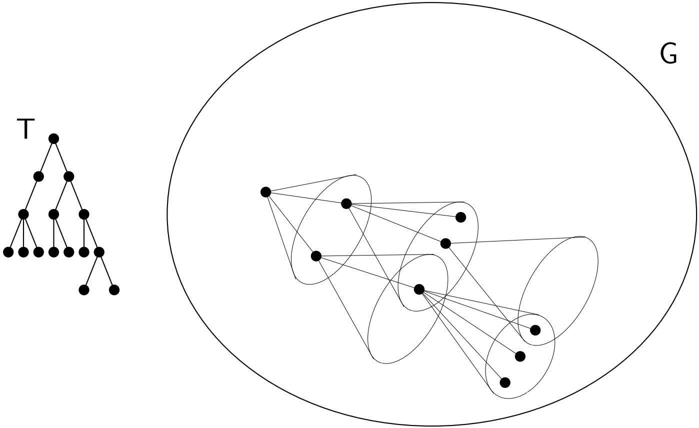
**SELF-AVOIDING TREE-INDEXED RANDOM WALK**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



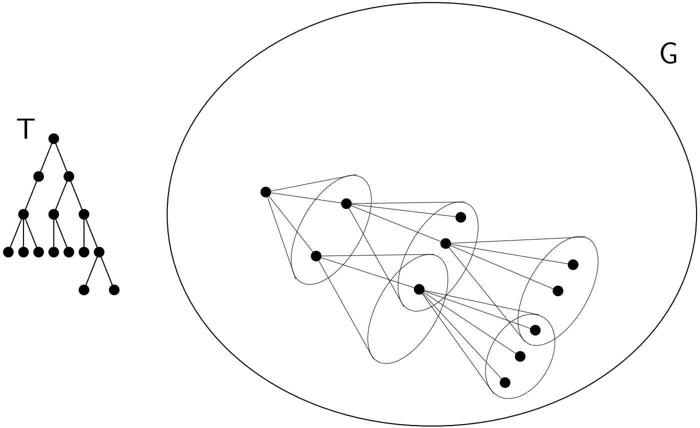
**SELF-AVOIDING TREE-INDEXED RANDOM WALK**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



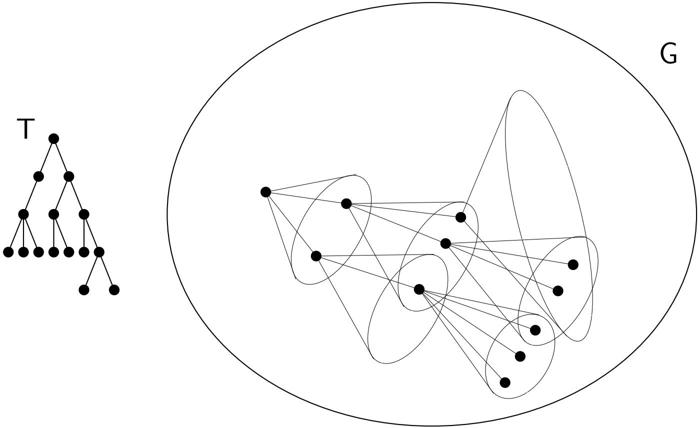
**Self-avoiding tree-indexed random walk**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



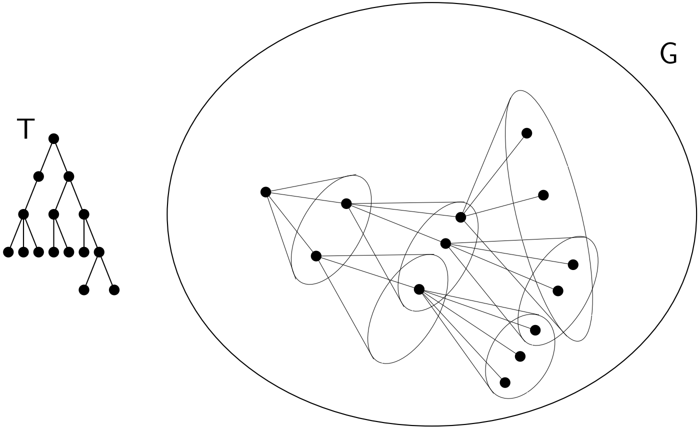
### SELF-AVOIDING TREE-INDEXED RANDOM WALK

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



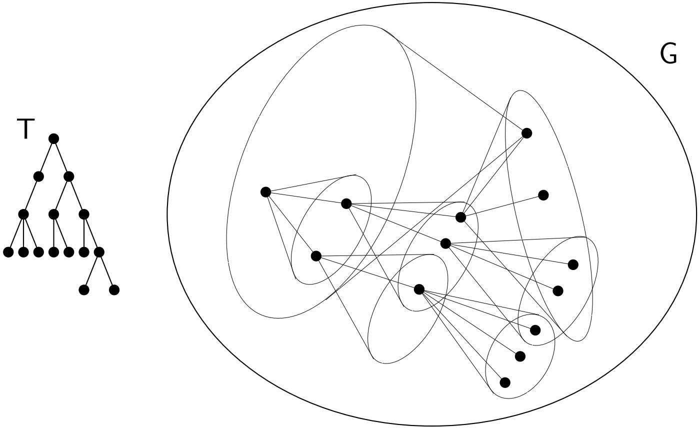
**SELF-AVOIDING TREE-INDEXED RANDOM WALK**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



**SELF-AVOIDING TREE-INDEXED RANDOM WALK**

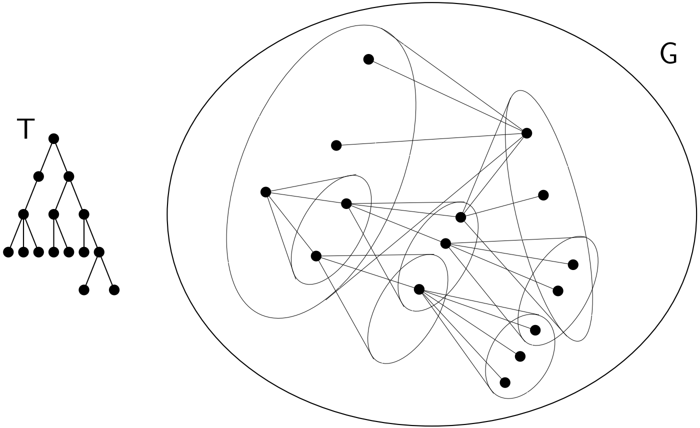
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



# The Algorithm

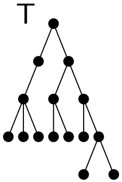
**Self-avoiding tree-indexed random walk**

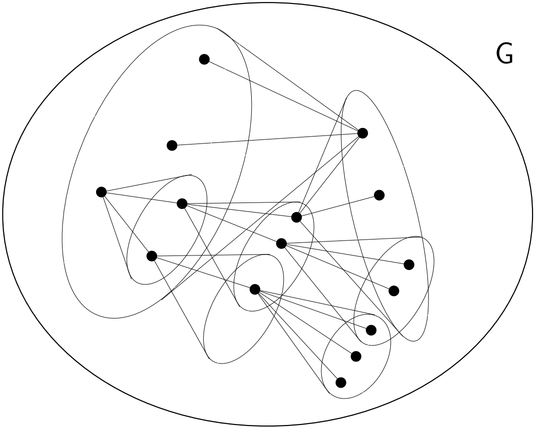
Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



**SELF-AVOIDING TREE-INDEXED RANDOM WALK**

Let $T$ be a rooted tree. Start by embedding the root arbitrarily. In each step, pick $u \in V(T)$ which is embedded already, and place its children randomly among the unoccupied neighbors of $f(u)$.



**META-CLAIM**

The image $f(T)$ behaves essentially like a random subset of $G$, in particular for each neighborhood $N(v)$ we expect $f(T)$ to occupy only a $|T|/|G| << 1$ fraction of $N(v)$.

- For each vertex $v \in V$, we define a *bad event* if $N(v)$ was visited too often by the embedding.
- Using martingale tail inequalities and the structure of $G$, we analyze the probability of a *bad event*.
- A careful counting scheme estimates the probability that any bad event occurs.

# Conclusion

## Open questions

- Instead of requiring girth $2k+1$ in Theorem 1, suppose $G$ has no cycles of length $2k$. Does our algorithm still work?
- It seems that the algorithm should work for any *pseudorandom graph*, but our analysis breaks down because two vertices might share too many neighbors.
- For random graphs $G_{n,d/n}$, the analysis can be extended to degrees $d = \omega(e^{\sqrt{\log n}})$. What about sparse graphs, with $d$ constant?