A Fast New Algorithm for Weak Graph Regularity

László Miklós Lovász
MIT

Joint work with Jacob Fox and Yufei Zhao

January 11, 2019
1. Regularity lemma background
2. Frieze-Kannan regularity
3. Algorithmic regularity
4. Proof Sketch
5. Further remarks
1. Regularity lemma background
2. Frieze-Kannan regularity
3. Algorithmic regularity
4. Proof Sketch
5. Further remarks
Szemerédi’s regularity lemma

Roughly speaking, in any graph, the vertices can be partitioned into a bounded number of parts, such that the graph is “random-like” between almost all pairs of parts.
Szemerédi’s regularity lemma

Roughly speaking, in any graph, the vertices can be partitioned into a bounded number of parts, such that the graph is “random-like” between almost all pairs of parts.
Szemerédi’s regularity lemma

Roughly speaking, in any graph, the vertices can be partitioned into a bounded number of parts, such that the graph is “random-like” between almost all pairs of parts.

- Gives a rough structural result for all graphs
Szemerédi’s regularity lemma

Roughly speaking, in any graph, the vertices can be partitioned into a bounded number of parts, such that the graph is “random-like” between almost all pairs of parts.

- Gives a rough structural result for all graphs
- Very important tool in graph theory
Szemerédi’s regularity lemma

Roughly speaking, in any graph, the vertices can be partitioned into a bounded number of parts, such that the graph is “random-like” between almost all pairs of parts.

- Gives a rough structural result for all graphs
- Very important tool in graph theory

How can we find such a partition algorithmically?
Regularity between sets

Given $X$ and $Y$ sets of vertices of $G$, let
Regularity between sets

Given $X$ and $Y$ sets of vertices of $G$, let

- $e(X, Y) =$ number of pairs of vertices in $X \times Y$ that have an edge between them.
Given $X$ and $Y$ sets of vertices of $G$, let

- $e(X, Y) = \text{number of pairs of vertices in } X \times Y \text{ that have an edge between them.}$
- $d(X, Y) = \frac{e(X, Y)}{|X||Y|}$. 

Roughly says graph between $X$ and $Y$ is "random-like".
Regularity between sets

Given $X$ and $Y$ sets of vertices of $G$, let

- $e(X, Y) = \text{number of pairs of vertices in } X \times Y \text{ that have an edge between them.}$
- $d(X, Y) = \frac{e(X, Y)}{|X||Y|}.$

**Definition**

Given a graph $G$ and two sets of vertices $X$ and $Y$, we say the pair $(X, Y)$ is $\epsilon$-regular if for any $X' \subset X$ with $|X'| \geq \epsilon |X|$, $Y' \subset Y$ with $|Y'| \geq \epsilon |Y|$, we have

$$\left| d(X', Y') - d(X, Y) \right| \leq \epsilon.$$
Regularity between sets

Given $X$ and $Y$ sets of vertices of $G$, let

- $e(X, Y) = \text{number of pairs of vertices in } X \times Y \text{ that have an edge between them.}$
- $d(X, Y) = \frac{e(X, Y)}{|X||Y|}.$

**Definition**

Given a graph $G$ and two sets of vertices $X$ and $Y$, we say the pair $(X, Y)$ is $\epsilon$-regular if for any $X' \subset X$ with $|X'| \geq \epsilon |X|$, $Y' \subset Y$ with $|Y'| \geq \epsilon |Y|$, we have

$$\left| d(X', Y') - d(X, Y) \right| \leq \epsilon.$$

Roughly says graph between $X$ and $Y$ is “random-like”.
Szemerédi’s regularity lemma

Definition
Given a partition $\mathcal{P}$ of the set of vertices $V$, we say it is equitable if the size of any two parts differs by at most one.
Definition
Given a partition $\mathcal{P}$ of the set of vertices $V$, we say it is \textit{equitable} if the size of any two parts differs by at most one.

Definition
Given an equitable partition $\mathcal{P}$ of the set of vertices $V$, it is \textit{$\epsilon$-regular} if all but $\epsilon|\mathcal{P}|^2$ pairs are $\epsilon$-regular.
Szemerédi’s regularity lemma

Definition
Given a partition $\mathcal{P}$ of the set of vertices $V$, we say it is \textit{equitable} if the size of any two parts differs by at most one.

Definition
Given an equitable partition $\mathcal{P}$ of the set of vertices $V$, it is $\epsilon$-\textit{regular} if all but $\epsilon|\mathcal{P}|^2$ pairs are $\epsilon$-regular.

Szemerédi’s regularity lemma
For every $\epsilon > 0$, there is an $M(\epsilon)$ such that for any graph $G = (V, E)$, there is an equitable, $\epsilon$-regular partition of the vertices into at most $M(\epsilon)$ parts.
Drawback of regularity lemma

The standard proof gives $M(\epsilon) \leq T(\epsilon^{-5})$, where $T$ is the tower function, i.e. $T(5) = 2^{2^{2^{2^2}}} = 2^{65536}$.
The standard proof gives $M(\varepsilon) \leq T(\varepsilon^{-5})$, where $T$ is the tower function, i.e. $T(5) = 2^{2^{2^{2^2}}} = 2^{65536}$.

Unfortunately, Gowers (1997) showed $M(\varepsilon) \geq T(\varepsilon^{-c})$ for some $c$. 
The standard proof gives $M(\epsilon) \leq T(\epsilon^{-5})$, where $T$ is the tower function, i.e. $T(5) = 2^{2^{2^2}} = 2^{65536}$.

Unfortunately, Gowers (1997) showed $M(\epsilon) \geq T(\epsilon^{-c})$ for some $c$.

*Frieze-Kannan* regularity lemma: weaker regularity property, better bounds.
1. Regularity lemma background

2. Frieze-Kannan regularity

3. Algorithmic regularity

4. Proof Sketch

5. Further remarks
**Frieze-Kannan (weak) regularity lemma**

**Definition**

Given a partition $\mathcal{P} = \{V_1, V_2, ..., V_k\}$ of the set of vertices $V$, it is *Frieze-Kannan $\epsilon$-regular* (FK-$\epsilon$-regular) if for any pair of sets $S, T \subseteq V$, we have

$$
\left| e(S, T) - \sum_{i,j=1}^{k} d(V_i, V_j) |S \cap V_i| |T \cap V_j| \right| \leq \epsilon |V|^2
$$
Frieze-Kannan (weak) regularity lemma

**Definition**

Given a partition $\mathcal{P} = \{V_1, V_2, \ldots, V_k\}$ of the set of vertices $V$, it is *Frieze-Kannan $\epsilon$-regular* (FK-$\epsilon$-regular) if for any pair of sets $S, T \subseteq V$, we have

$$\left| e(S, T) - \sum_{i,j=1}^{k} d(V_i, V_j)|S \cap V_i||T \cap V_j| \right| \leq \epsilon|V|^2$$

**Frieze-Kannan regularity lemma**

Let $\epsilon > 0$. Every graph has a Frieze-Kannan $\epsilon$-regular partition with at most $2^{2/\epsilon^2}$ parts.
Cut distance

**Definition**

Given two (weighted) graphs $G_1$ and $G_2$ on the same vertex set $V$, we define their *cut distance*

$$d_\square(G_1, G_2) = \frac{1}{|V|^2} \max_{S, T \subseteq V} |e_{G_1}(S, T) - e_{G_2}(S, T)|.$$
Cut distance

**Definition**

Given two (weighted) graphs $G_1$ and $G_2$ on the same vertex set $V$, we define their *cut distance*

$$d_{\square}(G_1, G_2) = \frac{1}{|V|^2} \max_{S,T \subseteq V} |e_{G_1}(S, T) - e_{G_2}(S, T)|.$$  

**Definition**

Given two (weighted) bipartite graphs $G_1$ and $G_2$ between the same vertex sets $V$ and $W$, we define their *cut distance*

$$d_{\square}(G_1, G_2) = \frac{1}{|V||W|} \max_{S \subseteq V, T \subseteq W} |e_{G_1}(S, T) - e_{G_2}(S, T)|.$$
Weak regularity

Given a partition $\mathcal{P}$, let $G_{\mathcal{P}}$ be the weighted graph obtained by taking, between each $X, Y \in \mathcal{P}$, weighted edges with weight $d(X, Y)$. 

Frieze-Kannan regularity lemma

Let $\epsilon > 0$. Every graph has a partition with at most $2^{2/\epsilon^2}$ parts such that $d(\mathcal{G}, G_{\mathcal{P}}) \leq \epsilon$.

Counting lemma

Given two graphs $G_1$ and $G_2$ on the same vertex set, for any graph $H$ on $k$ vertices, we have $|\text{hom}(H, G_1) - \text{hom}(H, G_2)| \leq e(H) d(\mathcal{G}_1, G_2) n^k$. 
Weak regularity

Given a partition $\mathcal{P}$, let $G_\mathcal{P}$ be the weighted graph obtained by taking, between each $X, Y \in \mathcal{P}$, weighted edges with weight $d(X, Y)$.

Partition $\mathcal{P}$ is FK-$\epsilon$-regular if and only if $d_{\square}(G, G_\mathcal{P}) \leq \epsilon$. 
Weak regularity

Given a partition $\mathcal{P}$, let $G_{\mathcal{P}}$ be the weighted graph obtained by taking, between each $X, Y \in \mathcal{P}$, weighted edges with weight $d(X, Y)$.

Partition $\mathcal{P}$ is FK-$\epsilon$-regular if and only if $d_{\square}(G, G_{\mathcal{P}}) \leq \epsilon$.

**Frieze-Kannan regularity lemma**

Let $\epsilon > 0$. Every graph has a partition with at most $2^{2/\epsilon^2}$ parts such that $d_{\square}(G, G_{\mathcal{P}}) \leq \epsilon$. 
Weak regularity

Given a partition $\mathcal{P}$, let $G_\mathcal{P}$ be the weighted graph obtained by taking, between each $X, Y \in \mathcal{P}$, weighted edges with weight $d(X, Y)$.

Partition $\mathcal{P}$ is FK-$\epsilon$-regular if and only if $d_{\square}(G, G_\mathcal{P}) \leq \epsilon$.

Frieze-Kannan regularity lemma

Let $\epsilon > 0$. Every graph has a partition with at most $2^{2/\epsilon^2}$ parts such that $d_{\square}(G, G_\mathcal{P}) \leq \epsilon$.

Counting lemma

Given two graphs $G_1$ and $G_2$ on the same vertex set, for any graph $H$ on $k$ vertices, we have

$$|\text{hom}(H, G_1) - \text{hom}(H, G_2)| \leq e(H)d_{\square}(G_1, G_2)n^k.$$
1. Regularity lemma background
2. Frieze-Kannan regularity
3. Algorithmic regularity
4. Proof Sketch
5. Further remarks
Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular.

Deterministic algorithms
- Kohayakawa-Rödl-Thoma (2003) - \(O(\epsilon(n^2))\)-time algorithm.
- Alon-Naor (2006) - Polynomial-time algorithm, at most \(T(\epsilon - O(1))\) parts. \((\omega < 2^{373})\)

Folklore/Tao blog post (2010)
Randomized algorithm in time \(O(\epsilon(1))\), \(\epsilon\)-regular partition.
Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular.
If pair \((X, Y)\) not \(\epsilon\)-regular, find \(S, T\) showing they are not \(\epsilon^4/16\)-regular, time \(O_\epsilon(n^{\omega+o(1)})\).

Deterministic algorithms
Kohayakawa-Rödl-Thoma (2003) - \(O_\epsilon(n^2)\)-time algorithm.
Alon-Naor (2006) - Polynomial-time algorithm, at most \(T(O_\epsilon(\epsilon^{-7}))\) parts. (\(\omega < 2^{373}\).)

Folklore/Tao blog post (2010) Randomized algorithm in time \(O_\epsilon(1)\), \(\epsilon\)-regular partition.
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NP-hard to test whether a pair ((X, Y)) is (\epsilon)-regular.</td>
</tr>
<tr>
<td>If pair ((X, Y)) not (\epsilon)-regular, find (S, T) showing they are not (\epsilon^4/16)-regular, time (O_\epsilon(n^{\omega+o(1)})). Implies algorithm with at most (T(\epsilon^{-O(1)})) parts. ((\omega &lt; 2.373))</td>
</tr>
</tbody>
</table>
Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular. If pair \((X, Y)\) not \(\epsilon\)-regular, find \(S, T\) showing they are not \(\epsilon^4/16\)-regular, time \(O_\epsilon(n^{\omega+o(1)})\). Implies algorithm with at most \(T(\epsilon^{-O(1)})\) parts. \((\omega < 2.373)\)

Deterministic algorithms
Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular. If pair \((X, Y)\) not \(\epsilon\)-regular, find \(S, T\) showing they are not \(\epsilon^4/16\)-regular, time \(O_\epsilon(n^{\omega+o(1)})\). Implies algorithm with at most \(T(\epsilon^{-O(1)})\) parts. \((\omega < 2.373)\)

Deterministic algorithms

Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular.
If pair \((X, Y)\) not \(\epsilon\)-regular, find \(S, T\) showing they are not \(\epsilon^4/16\)-regular, time \(O_\epsilon(n^{\omega+o(1)})\). Implies algorithm with at most \(T(\epsilon^{-O(1)})\) parts. (\(\omega < 2.373\))

Deterministic algorithms

- Kohayakawa-Rödl-Thoma (2003) - \(O_\epsilon(n^2)\)-time algorithm.
Algorithmic regularity


NP-hard to test whether a pair \((X, Y)\) is \(\epsilon\)-regular. If pair \((X, Y)\) not \(\epsilon\)-regular, find \(S, T\) showing they are not \(\epsilon^4/16\)-regular, time \(O_{\epsilon}(n^{\omega+o(1)})\). Implies algorithm with at most \(T(\epsilon^{-O(1)})\) parts. (\(\omega < 2.373\))

**Deterministic algorithms**

- Alon-Naor (2006) - Polynomial-time algorithm, at most \(T(O(\epsilon^{-7}))\) parts.
Algorithmic regularity


NP-hard to test whether a pair $(X, Y)$ is $\epsilon$-regular. If pair $(X, Y)$ is not $\epsilon$-regular, find $S, T$ showing they are not $\epsilon^4/16$-regular, time $O_\epsilon(n^{\omega+o(1)})$. Implies algorithm with at most $T(\epsilon^{-O(1)})$ parts. ($\omega < 2.373$)

### Deterministic algorithms

- Kohayakawa-Rödl-Thoma (2003) - $O_\epsilon(n^2)$-time algorithm.
- Alon-Naor (2006) - Polynomial-time algorithm, at most $T(O(\epsilon^{-7}))$ parts.

### Folklore/Tao blog post (2010)

Randomized algorithm in time $O_\epsilon(1)$, $\epsilon$-regular partition.
Frieze-Kannan (1996)

Constant time \textit{probabilistic} algorithm.
Algorithmic Frieze-Kannan

Frieze-Kannan (1996)
Constant time *probabilistic* algorithm.

Dellamonica-Kalyanasundaram-Martin-Rödl-Shapira
Deterministic algorithm which finds a Frieze-Kannan $\epsilon$-regular partition
Algorithmic Frieze-Kannan

**Frieze-Kannan (1996)**
Constant time *probabilistic* algorithm.

**Dellamonica-Kalyanasundaram-Martin-Rödl-Shapira**
Deterministic algorithm which finds a Frieze-Kannan $\epsilon$-regular partition
- in time $\epsilon^{-6} n^{\omega+o(1)}$ into at most $2^{O(\epsilon^{-7})}$ parts (2012)
Algorithmic Frieze-Kannan

Frieze-Kannan (1996)

Constant time *probabilistic* algorithm.

Dellamonica-Kalyanasundaram-Martin-Rödl-Shapira

Deterministic algorithm which finds a Frieze-Kannan $\epsilon$-regular partition

- in time $\epsilon^{-6} n^{\omega + o(1)}$ into at most $2^{O(\epsilon^{-7})}$ parts (2012)
- in time $O(2^{2\epsilon^{-O(1)}} n^2)$ into at most $2^{\epsilon^{-O(1)}}$ parts (2015)
<table>
<thead>
<tr>
<th>Algorithmic Frieze-Kannan</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Frieze-Kannan (1996)</strong></td>
</tr>
<tr>
<td>Constant time <em>probabilistic</em> algorithm.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Dellamonica-Kalyanasundaram-Martin-Rödl-Shapira</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic algorithm which finds a Frieze-Kannan $\epsilon$-regular partition</td>
</tr>
<tr>
<td>- in time $\epsilon^{-6} n^{\omega+o(1)}$ into at most $2^{O(\epsilon^{-7})}$ parts (2012)</td>
</tr>
<tr>
<td>- in time $O(2^{2\epsilon^{-O(1)}} n^2)$ into at most $2^{\epsilon^{-O(1)}}$ parts (2015)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Fox-L.-Zhao</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic algorithm which finds a Frieze-Kannan $\epsilon$-regular partition in time $\epsilon^{-O(1)} n^2$.</td>
</tr>
</tbody>
</table>
We can prove something slightly stronger.
We can prove something slightly stronger.

**Fox-L.-Zhao**

There is an $\epsilon^{-O(1)} n^2$-time algorithm which, given $\epsilon > 0$, an $n$-vertex graph $G$, outputs $r \leq \epsilon^{-O(1)}$, subsets $S_1, S_2, ..., S_r, T_1, T_2, ..., T_r \subset V(G)$ and numbers $c_1, c_2, ..., c_r = \pm \epsilon^8 / 300$ such that

$$d_{\square}(G, c_1 K_{S_1, T_1} + c_2 K_{S_2, T_2} + ... + c_r K_{S_r, T_r}) \leq \epsilon.$$
1. Regularity lemma background
2. Frieze-Kannan regularity
3. Algorithmic regularity
4. Proof Sketch
5. Further remarks
There is a \((C/\epsilon)^O(1)n^2\)-time algorithm which does the following.

Suppose that an \(n \times n\) matrix \(A\) has \(\|A\|_{\text{max}} \leq C\), and each row \(\|a_i\|_2^2 \leq n\) and column \(\|a_j\|_2^2 \leq n\). Then the algorithm outputs either that each singular value of \(A\) is at most \(\epsilon n\), i.e. its operator norm \(\|A\| \leq \epsilon n\), or sets \(S, T \subseteq [n]\) such that \(\left\| \sum_{i \in S, k \in T} a_{i,k} \right\| \geq \epsilon^{8/100} n^2\).
Main lemma

Main step

There is a \((C/\epsilon)^{O(1)}n^2\)-time algorithm which does the following. Suppose that an \(n \times n\) matrix \(A\) has \(\|A\|_{max} \leq C\), and each row \(\|a_i\|_2^2 \leq n\) and column \(\|a^j\|_2^2 \leq n\).
Main step

There is a \((C/\epsilon)^{O(1)}n^2\)-time algorithm which does the following. Suppose that an \(n \times n\) matrix \(A\) has \(\|A\|_{\text{max}} \leq C\), and each row \(\|a_i\|_2^2 \leq n\) and column \(\|a^j\|_2^2 \leq n\). Then the algorithm outputs either
Main lemma

Main step

There is a \((C/\epsilon)^{O(1)}n^2\)-time algorithm which does the following. Suppose that an \(n \times n\) matrix \(A\) has \(\|A\|_{\max} \leq C\), and each row \(\|a_i\|_2^2 \leq n\) and column \(\|a_j\|_2^2 \leq n\). Then the algorithm outputs either

- that each singular value of \(A\) is at most \(\epsilon n\), i.e. its operator norm \(\|A\| \leq \epsilon n\), or
There is a \((C/\epsilon)^{O(1)}n^2\)-time algorithm which does the following. Suppose that an \(n \times n\) matrix \(A\) has \(\|A\|_{\text{max}} \leq C\), and each row \(\|a_i\|_2^2 \leq n\) and column \(\|a_j\|_2^2 \leq n\). Then the algorithm outputs either

- that each singular value of \(A\) is at most \(\epsilon n\), i.e. its operator norm \(\|A\| \leq \epsilon n\), or
- sets \(S, T \subseteq [n]\) such that

\[
\left| \sum_{i \in S, k \in T} a_{i,k} \right| \geq \frac{\epsilon^8}{100} n^2.
\]
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_\square = o(n^2)$. 

Key idea: Can check fourth condition for pairs along a constant degree expander graph. If off, can find a pair of sets with non-small density. Idea of testing along expanders first appeared in Kohayakawa-Rödl-Thomason paper.
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_{\Box} = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_{\square} = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
- $trAA^T A A^T = o(n^4)$. 

**Key idea:** Can check fourth condition for pairs along a constant degree expander graph. If off, can find a pair of sets with non-small density. The idea of testing along expanders first appeared in Kohayakawa-Rödl-Thomason paper.
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_{\square} = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
- $trAA^T A A^T = o(n^4)$.
- Most pairs of rows have inner product $o(n^2)$. 

**Key idea**: Can check fourth condition for pairs along a constant degree expander graph. If off, can find a pair of sets with non-small density. Idea of testing along expanders first appeared in Kohayakawa-Rödl-Thoma paper.
Main idea

Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_\square = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
- $trAA^T AA^T = o(n^4)$.
- Most pairs of rows have inner product $o(n^2)$.

*Key idea:* Can check fourth condition for pairs along a constant degree *expander graph.*
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_{\square} = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
- $trAA^T A^T = o(n^4)$.
- Most pairs of rows have inner product $o(n^2)$.

**Key idea:** Can check fourth condition for pairs along a constant degree expander graph.

If off, can find a pair of sets with non-small density.
Let $A$ be a “bounded” matrix. The following are “roughly equivalent” (Chung-Graham-Wilson, Thomason):

- $\|A\|_{\Box} = o(n^2)$.
- Singular values of $A$ are all $o(n)$.
- $trAA^T AA^T = o(n^4)$.
- Most pairs of rows have inner product $o(n^2)$.

*Key idea*: Can check fourth condition for pairs along a constant degree expander graph.

If off, can find a pair of sets with non-small density.

Idea of testing along expanders first appeared in Kohayakawa-Rödl-Thoma paper.
Iterative proof

Sequence of iterative steps, $A = A_0, A_1, \cdots, A_s$. 
Iterative proof

Sequence of iterative steps, $A = A_0, A_1, \cdots, A_s$.

In each step, apply main lemma to $A_l$. If $\|A_l\| \leq \epsilon n$, done. Else find sets $S$ and $T$. WLOG sum across $S \times T$ is $\gtrsim \epsilon^8 n^2$.
Sequence of iterative steps, $A = A_0, A_1, \cdots, A_s$.

In each step, apply main lemma to $A_l$. If $\|A_l\| \leq \epsilon n$, done. Else find sets $S$ and $T$. WLOG sum across $S \times T$ is $\gtrsim \epsilon^8 n^2$.

By slightly decreasing sum across sets, can assume each row and column has positive, not too small sum.
Iterative proof

Sequence of iterative steps, $A = A_0, A_1, \ldots, A_s$.

In each step, apply main lemma to $A_l$. If $\|A_l\| \leq \epsilon n$, done. Else find sets $S$ and $T$. WLOG sum across $S \times T$ is $\geq \epsilon^8 n^2$.

By slightly decreasing sum across sets, can assume each row and column has positive, not too small sum.

Set $A_{l+1} = A_l - t1_s1_T^T$, $t = c\epsilon^8$. $L^2$-norm of each row and column cannot increase, and $\|A\|_{Fr}^2$ decreases by $\geq \epsilon^{16} n^2$. 
Sequence of iterative steps, $A = A_0, A_1, \cdots, A_s$.

In each step, apply main lemma to $A_l$. If $\|A_l\| \leq \epsilon n$, done. Else find sets $S$ and $T$. WLOG sum across $S \times T$ is $\gtrsim \epsilon^8 n^2$.

By slightly decreasing sum across sets, can assume each row and column has positive, not too small sum.

Set $A_{l+1} = A_l - t1_s1_T^T$, $t = c\epsilon^8$. $L^2$-norm of each row and column cannot increase, and $\|A\|_{Fr}^2$ decreases by $\gtrsim \epsilon^{16} n^2$.

Must end after $O(\epsilon^{-16})$ steps.
Regularity lemma background

Frieze-Kannan regularity

Algorithmic regularity

Proof Sketch

Further remarks
Counting subgraphs

Algorithmic problem

Count the number of copies of a graph $H$ in a graph $G$ on $n$ vertices.
Algorithmic problem

Count the number of copies of a graph $H$ in a graph $G$ on $n$ vertices.

Special case: is there a single copy?
Algorithmic problem

Count the number of copies of a graph $H$ in a graph $G$ on $n$ vertices.

Special case: is there a single copy?

Even for $K_k$, Zuckerman showed NP-hard to approximate the size of the largest clique within a factor $n^{1-\epsilon}$, building on an earlier result of Håstad.
Counting subgraphs

Algorithmic problem
Count the number of copies of a graph $H$ in a graph $G$ on $n$ vertices.

Special case: is there a single copy?

Even for $K_k$, Zuckerman showed NP-hard to approximate the size of the largest clique within a factor $n^{1-\epsilon}$, building on an earlier result of Håstad.

How fast can we approximate the count within an additive $\epsilon n |V(H)|$?
Counting subgraphs

Algorithmic problem

Count the number of copies of a graph $H$ on $k$ vertices in a graph on $n$ vertices, up to an error of at most $\epsilon n^k$. 
Counting subgraphs

Algorithmic problem

Count the number of copies of a graph $H$ on $k$ vertices in a graph on $n$ vertices, up to an error of at most $\epsilon n^k$.

A simple randomized algorithm gives 99% certainty:
Algorithmic problem

Count the number of copies of a graph $H$ on $k$ vertices in a graph on $n$ vertices, up to an error of at most $\epsilon n^k$.

A simple randomized algorithm gives 99% certainty:
Sample $10/\epsilon^2$ random $k$-sets of vertices.
Algorithmic problem

Count the number of copies of a graph $H$ on $k$ vertices in a graph on $n$ vertices, up to an error of at most $\epsilon n^k$.

A simple randomized algorithm gives 99% certainty:
Sample $10/\epsilon^2$ random $k$-sets of vertices.

What about deterministic algorithms?
Counting subgraphs

**Algorithmic problem**

Count the number of copies of a graph $H$ on $k$ vertices in a graph $G$ on $n$ vertices, up to an error of at most $\epsilon n^k$. 

Can be done in time $2^{(k/\epsilon)} O(1)n^{\omega + o(1)}$.

Fox-L.-Zhao (2017)
Can be done in time $\epsilon^{-O_H(1)}n^2$. 
**Algorithmic problem**

Count the number of copies of a graph $H$ on $k$ vertices in a graph $G$ on $n$ vertices, up to an error of at most $\varepsilon n^k$.


Can be done in time $2^{(k/\varepsilon)^O(1)} n^{\omega+o(1)}$. 
Algorithmic problem

Count the number of copies of a graph $H$ on $k$ vertices in a graph $G$ on $n$ vertices, up to an error of at most $\epsilon n^k$.


Can be done in time $2^{(k/\epsilon)^O(1)} n^{\omega+o(1)}$.

Fox-L.-Zhao (2017)

Can be done in time $\epsilon^{-O_H(1)} n^2$. 