The Metric Relaxation for 0-Extension Admits an $\Omega\left(\log^{2/3} k\right)$ Gap

Nitzan Tur

Joint work with: Roy Schwartz

Problem Definition

Problem Definition

Input:

- $\mathcal{G} = (V, E)$ equipped with $w : E \to \mathbb{R}^{+}$.

Problem Definition

Input:

- $\mathcal{G} = (V, E)$ equipped with $w : E \to \mathbb{R}^+$.
- $T = \{t_1, \ldots, t_k\} \subseteq V$ terminals.

Problem Definition

Input:

- $\mathcal{G} = (V, E)$ equipped with $w : E \to \mathbb{R}^+$.
- $T = \{t_1, \ldots, t_k\} \subseteq V$ terminals.
- $D : T \times T \to \mathbb{R}^+$ a semi-metric.

Problem Definition

Input:

- $\mathcal{G} = (V, E)$ equipped with $w : E \to \mathbb{R}^+$.
- $T = \{t_1, \ldots, t_k\} \subseteq V$ terminals.
- $D : T \times T \to \mathbb{R}^+$ a semi-metric.

Goal: Find $f : V \to T$, identity on T, minimizing:

$$\sum_{(u,v) \in E} w_e \cdot D(f(u), f(v)).$$

The Metric Extension Relaxation

A solution f:

1. Extends D from T to V.
2. Satisfies: $\min_{i=1}^{k}\{D(u, t_i)\} = 0,\ \forall u \in V$.

The Metric Extension Relaxation

A solution f:

1. Extends D from T to V.
2. Satisfies: $\min_{i=1}^{k}\{D(u,t_i)\} = 0,\ \forall u \in V$.

The metric extension relaxation (MET) ignores 2 above [Karzanov-98]:

The Metric Extension Relaxation

A solution f:

1. Extends D from T to V.
2. Satisfies: $\min_{i=1}^{k}\{D(u,t_i)\} = 0, \ \forall u \in V$.

The metric extension relaxation (MET) ignores 2 above [Karzanov-98]:

$$
\begin{aligned}
(MET) \qquad \min \quad & \sum_{e=(u,v)\in E} w_e \cdot \delta(u,v) & \\
s.t. \quad & (V,\delta) \text{ is a semi-metric space} & (1)\\
& \delta(t_i,t_j) = D(t_i,t_j) \qquad \forall t_i,t_j \in T, i \neq j & (2)
\end{aligned}
$$

Known Results - Upper Bounds

$O(\log(k))$ [Călinsecu-Karloff-Rabani-05]

$O\left(\frac{\log(k)}{\log\log(k)}\right)$ [Fakcharoenphol-Harrelson-Rao-Talwar-03]

} round (MET)

Known Results - Upper Bounds

$O(\log(k))$	[Călinsecu-Karloff-Rabani-05]	}	round (MET)
$O\left(\frac{\log(k)}{\log\log(k)}\right)$	[Fakcharoenphol-Harrelson-Rao-Talwar-03]		

Above algorithms consist of two steps:

1. Select "scale" for each vertex.
2. Decompose the metric δ in each scale.

Known Results - Lower Bounds

(MET) admits an integrality gap of $\Omega(\sqrt{\log k})$ [Călinsecu-Karloff-Rabani-05].

Known Results - Lower Bounds

(MET) admits an integrality gap of $\Omega(\sqrt{\log k})$ [Călinsecu-Karloff-Rabani-05].

Earthmover based relaxation [Chekuri-Khanna-Naor-Zosin-04]:

- Embeds vertices to Δ_k.

Known Results - Lower Bounds

(MET) admits an integrality gap of $\Omega(\sqrt{\log k})$ [Călinsecu-Karloff-Rabani-05].

Earthmover based relaxation [Chekuri-Khanna-Naor-Zosin-04]:

- Embeds vertices to Δ_k.
- At least as strong as (MET).

Known Results - Lower Bounds

(MET) admits an integrality gap of $\Omega(\sqrt{\log k})$ [Călinsecu-Karloff-Rabani-05].

Earthmover based relaxation [Chekuri-Khanna-Naor-Zosin-04]:

- Embeds vertices to Δ_k.
- At least as strong as (MET).
- Assuming UGC [Manokaran-Naor-Raghavendra-Schwartz-08]:

$$\text{integrality gap of } \alpha \ \Rightarrow\ \alpha\text{-hardness.}$$

Known Results - Lower Bounds

(MET) admits an integrality gap of $\Omega(\sqrt{\log k})$ [Călinsecu-Karloff-Rabani-05].

Earthmover based relaxation [Chekuri-Khanna-Naor-Zosin-04]:

- Embeds vertices to Δ_k.
- At least as strong as (MET).
- Assuming UGC [Manokaran-Naor-Raghavendra-Schwartz-08]:

$$\text{integrality gap of } \alpha \;\Rightarrow\; \alpha\text{-hardness.}$$

- Admits integrality gap of $\Omega(\sqrt{\log k})$ [Karloff-Khot-Mehta-Rabani-09].

Known Results - Summary

Comments:

1. Known algorithms do not know how to exploit earthmover metrics.

Known Results - Summary

Comments:

1. Known algorithms do not know how to exploit earthmover metrics.
2. $O(\sqrt{\log k})$ barrier for designing and analyzing gap instances.

Known Results - Summary

Comments:

1. Known algorithms do not know how to exploit earthmover metrics.
2. $O(\sqrt{\log k})$ barrier for designing and analyzing gap instances.

Question: bridge the gap between $O\left(\frac{\log(k)}{\log\log(k)}\right)$ and $\Omega(\sqrt{\log k})$ for (MET)?

Our Results

Theorem [Schwartz-T-20]

For every k, (MET) admits an integrality gap of $\Omega(\log^{2/3}(k))$ for 0-Extension.

Our Results

Theorem [Schwartz-T-20]

For every k, (MET) admits an integrality gap of $\Omega(\log^{2/3}(k))$ for 0-Extension.

Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Graph Extensions

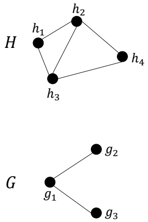
Definition

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:

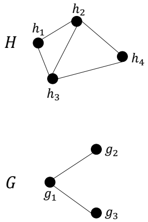
Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:

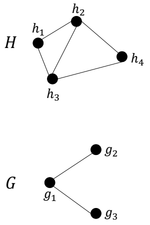


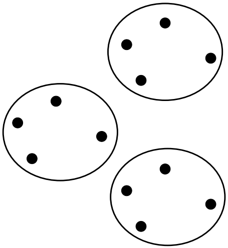
Vertices:

- $V_G \times V_H$.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\mathrm{Ext}(G, H)$ is a distribution over graphs:



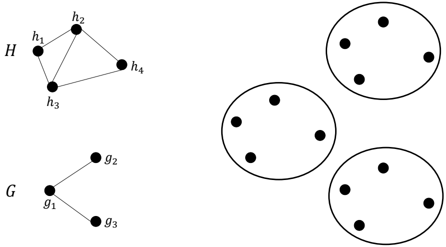


Vertices:

- $V_G \times V_H$.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:

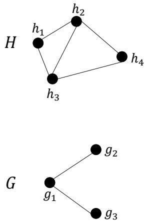


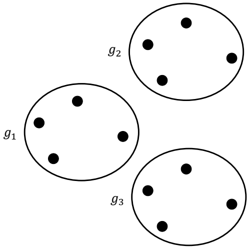
Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



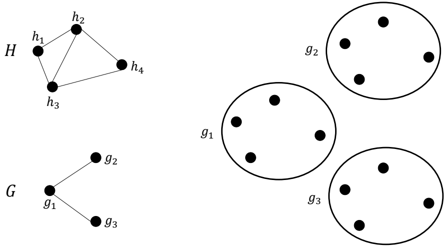


Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



Vertices:

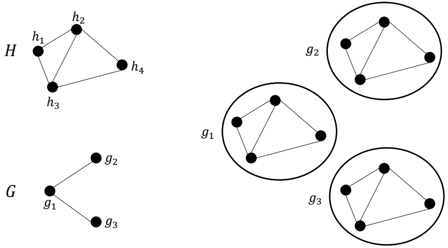
- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Edges:

- intra-cloud edges are E_H.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



Vertices:

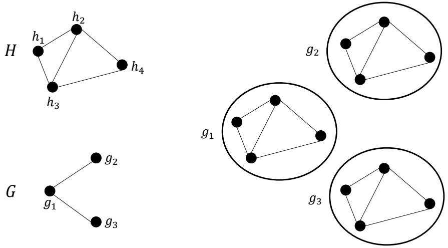
- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Edges:

- intra-cloud edges are E_H.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

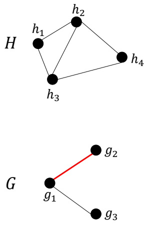
Edges:

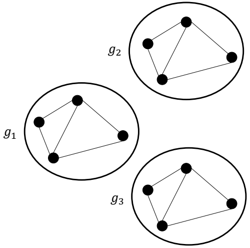
- intra-cloud edges are E_H.
- inter-cloud edges $(g_i, g_j) \in E_G$:

  uniform random matching.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:





Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

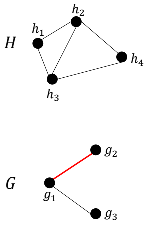
Edges:

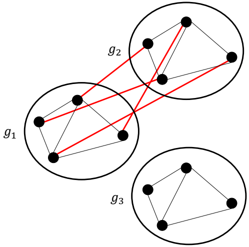
- intra-cloud edges are E_H.
- inter-cloud edges $(g_i, g_j) \in E_G$:

  uniform random matching.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:





Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

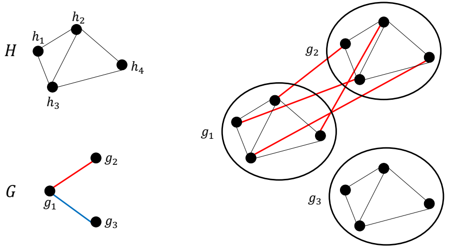
Edges:

- intra-cloud edges are E_H.
- inter-cloud edges $(g_i, g_j) \in E_G$:

  uniform random matching.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\text{Ext}(G, H)$ is a distribution over graphs:



Vertices:

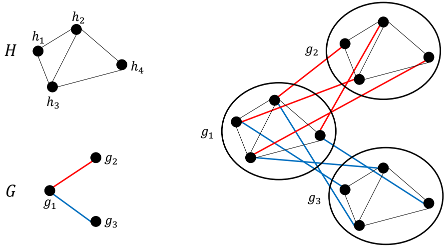
- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Edges:

- intra-cloud edges are E_H.
- inter-cloud edges $(g_i, g_j) \in E_G$:
  uniform random matching.

Definition

Given $G = (V_G, E_G)$ and $H = (V_H, E_H)$, $\mathrm{Ext}(G, H)$ is a distribution over graphs:



Vertices:

- $V_G \times V_H$.
- $\{(g, h) : h \in V_H\}$ is g's cloud.

Edges:

- intra-cloud edges are E_H.
- inter-cloud edges $(g_i, g_j) \in E_G$:

  uniform random matching.

Graph Extensions

Comments:

Graph Extensions

Comments:

1. Naturally captures edge lengths:

$$\{\ell_H(e)\}_{e \in E_H},\ \{\ell_G(e)\}_{e \in E_G} \quad \Rightarrow \quad \ell(e) = \begin{cases} \ell_H(e) & \text{(intra-cloud)} \\ \ell_G(e) & \text{(inter-cloud)} \end{cases}$$

Graph Extensions

Comments:

1. Naturally captures edge lengths:

$$\{\ell_H(e)\}_{e\in E_H}, \{\ell_G(e)\}_{e\in E_G} \quad \Rightarrow \quad \ell(e) = \begin{cases} \ell_H(e) & \text{(intra-cloud)} \\ \ell_G(e) & \text{(inter-cloud)} \end{cases}$$

2. H has no edges $\Rightarrow$ graph extensions coincide with lifts of graphs.

Graph Extensions

Comments:

1. Naturally captures edge lengths:

$$\{\ell_H(e)\}_{e\in E_H},\ \{\ell_G(e)\}_{e\in E_G} \quad \Rightarrow \quad \ell(e) = \begin{cases} \ell_H(e) & \text{(intra-cloud)} \\ \ell_G(e) & \text{(inter-cloud)} \end{cases}$$

2. H has no edges $\Rightarrow$ graph extensions coincide with lifts of graphs.

3. Relates to group extensions:

$$G \text{ and } H \text{ are Cayley graphs and } K \text{ is a group extension of } G \text{ by } H$$
$$\Downarrow$$
$$K\text{'s Cayley graph is in the support of } \mathrm{Ext}(G, H)$$

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions “split”.
- Most graph extensions do not “split”.

Intuition

Given $X \sim \mathrm{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Intuition

Given $X \sim \mathrm{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Intuition

Given $X \sim \mathrm{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

What is a Split?

Recall proof overview:

- Small gap implies that graph extensions “split”.
- Most graph extensions do not “split”.

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

Notes:

- Need to quantify **most** and **close**.
- Captures split extensions of groups.

The Instance

Instance Definition

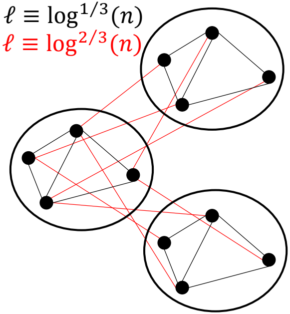
$X \sim \mathrm{Ext}(G, H)$ where:

1. G and H are constant degree high girth expanders on n vertices.
2. $\ell_H(e) \equiv \log^{1/3}(n)$ and $\ell_G(e) \equiv \log^{2/3}(n)$.

Instance Definition

$X \sim \text{Ext}(G, H)$ where:

1. G and H are constant degree high girth expanders on n vertices.
2. $\ell_H(e) \equiv \log^{1/3}(n)$ and $\ell_G(e) \equiv \log^{2/3}(n)$.

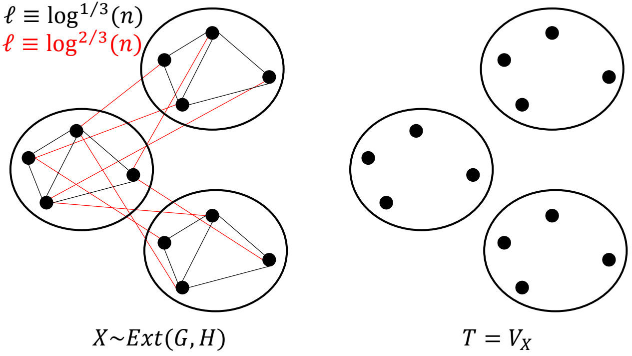


$X \sim Ext(G, H)$

Instance Definition

$X \sim \text{Ext}(G, H)$ where:

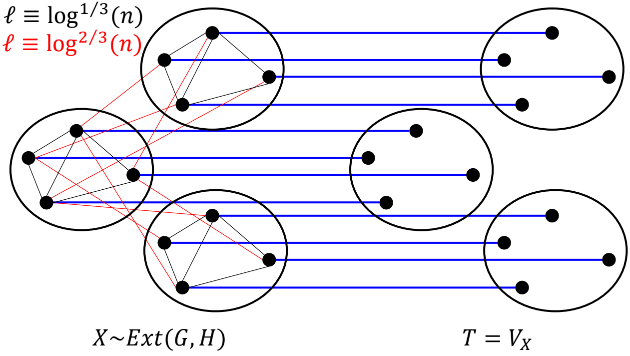
1. G and H are constant degree high girth expanders on n vertices.
2. $\ell_H(e) \equiv \log^{1/3}(n)$ and $\ell_G(e) \equiv \log^{2/3}(n)$.



Instance Definition

$X \sim \text{Ext}(G, H)$ where:

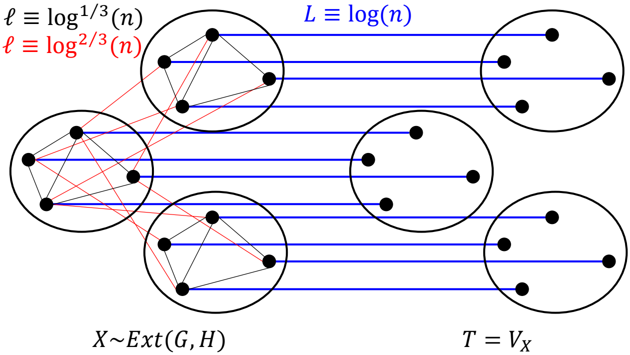
1. G and H are constant degree high girth expanders on n vertices.
2. $\ell_H(e) \equiv \log^{1/3}(n)$ and $\ell_G(e) \equiv \log^{2/3}(n)$.



Instance Definition

$X \sim \mathrm{Ext}(G, H)$ where:

1. G and H are constant degree high girth expanders on n vertices.
2. $\ell_H(e) \equiv \log^{1/3}(n)$ and $\ell_G(e) \equiv \log^{2/3}(n)$.

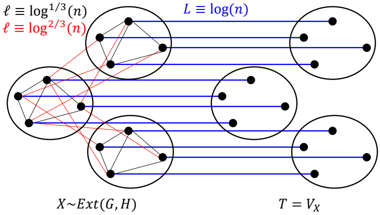


Instance Definition (cont.)

$\mathcal{G}$ and T are defined, what remains?

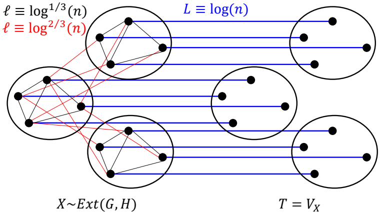
Instance Definition (cont.)

$\mathcal{G}$ and T are defined, what remains?



Instance Definition (cont.)

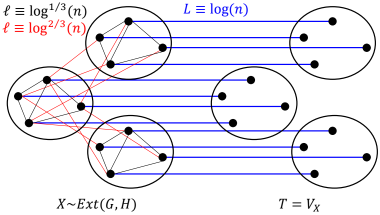
$\mathcal{G}$ and T are defined, what remains?



- (T, D) shortest path metric on $\mathcal{G}$.

Instance Definition (cont.)

$\mathcal{G}$ and T are defined, what remains?



- (T, D) shortest path metric on $\mathcal{G}$.
- Weights w are inverse of length.

The Fractional Solution

- Our construction naturally gives a solution to (MET).

The Fractional Solution

- Our construction naturally gives a solution to (MET).
- Each edge costs 1.

The Fractional Solution

- Our construction naturally gives a solution to (MET).
- Each edge costs 1.
- There are $\Theta(n^2)$ edges in the instance.

The Fractional Solution

- Our construction naturally gives a solution to (MET).
- Each edge costs 1.
- There are $\Theta(n^2)$ edges in the instance.
- $\Theta(n^2)$ in total.

Split

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions “split”.
- Most graph extensions do not “split”.

Split

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions “split”.
- Most graph extensions do not “split”.

- Assume we have a small gap $O(\varepsilon^2 \log^{2/3}(n))$:
$$f : V_X \to T \text{ costs } O(\varepsilon^2 \log^{2/3}(n) \cdot n^2).$$
- At most εn^2 edges cost more than $\varepsilon \log^{2/3}(n)$.

Split

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

- Assume we have a small gap $O(\varepsilon^2 \log^{2/3}(n))$:
$$f : V_X \to T \text{ costs } O(\varepsilon^2 \log^{2/3}(n) \cdot n^2).$$
- At most εn^2 edges cost more than $\varepsilon \log^{2/3}(n)$.

Conclusion: $\delta(f(u), f(v)) \leq \varepsilon \log^{2/3}(n) \delta(u, v)$ for $1 - \varepsilon$ of the edges.

Split - Existence of Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

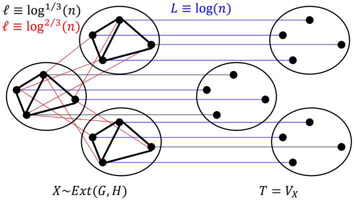
1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

Split - Existence of Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



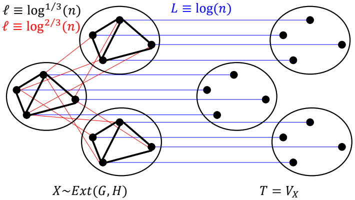
- Distance between terminals $\geq L = \log(n)$.
- Intra-cloud neighbors distance is $\log^{1/3}(n)$.

Split - Existence of Representatives

Intuition

Given $X \sim \text{Ext}(G,H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



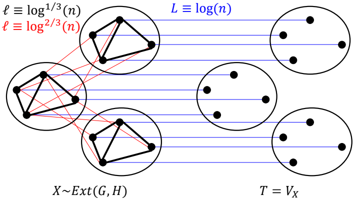
- Distance between terminals $\geq L = \log(n)$.
- Intra-cloud neighbors distance is $\log^{1/3}(n)$.
- Most intra-cloud neighbors are assigned to the same terminal.
- H is an expander.

Split - Existence of Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



- Distance between terminals $\geq L = \log(n)$.
- Intra-cloud neighbors distance is $\log^{1/3}(n)$.
- Most intra-cloud neighbors are assigned to the same terminal.
- H is an expander.

Most clouds have a consensus and this consensus is the representative.

Split - Representatives are Close

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

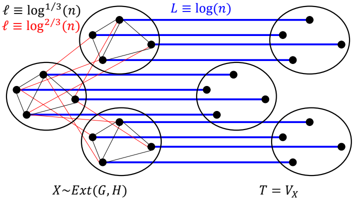
1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

Split - Representatives are Close

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



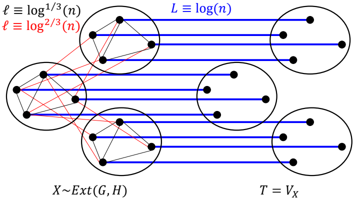
- Blue edges length is $\log(n)$.
- Red edges (inter-cloud) length is $\log^{2/3}(n)$.

Split - Representatives are Close

Intuition

Given $X \sim \mathrm{Ext}(G,H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



- Blue edges length is $\log(n)$.
- Red edges (inter-cloud) length is $\log^{2/3}(n)$.

Cloud of $f(g)$ is $\underbrace{\varepsilon \log^{2/3}(n)}_{\text{gap}} \cdot \log(n) / \log^{2/3}(n) = \varepsilon \log(n)$ **hops** away in G from g.

Split - Neighboring Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

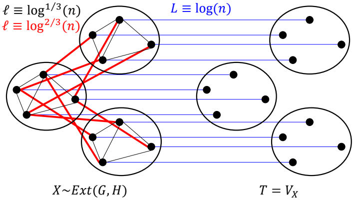
1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

Split - Neighboring Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



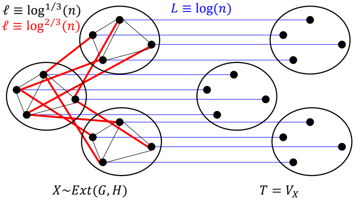
- Red edges (inter-cloud) length is $\log^{2/3}(n)$.
- Blue edges (intra-cloud) length is $\log^{1/3}(n)$.

Split - Neighboring Representatives

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.



- Red edges (inter-cloud) length is $\log^{2/3}(n)$.
- Blue edges (intra-cloud) length is $\log^{1/3}(n)$.

$f(g_1)$ and $f(g_2)$ are $\underbrace{\varepsilon \log^{2/3}(n)}_{\text{gap}} \cdot \log^{2/3}(n) / \log^{1/3}(n) = \varepsilon \log(n)$ **hops** away in X.

Split - Topology

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

Split - Topology

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

- We have $f : V_G \to V_X$, the representative map.
- Let $\pi : V_X \to V_G$ projection.

Split - Topology

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

- We have $f : V_G \to V_X$, the representative map.
- Let $\pi : V_X \to V_G$ projection.

Topological Property: $\pi \circ f$ preserves the cycle structure of G.

Split - Topology

Intuition

Given $X \sim \text{Ext}(G, H)$ a split assigns to **most** clouds g a **representative** $f(g) \in V_X$ where:

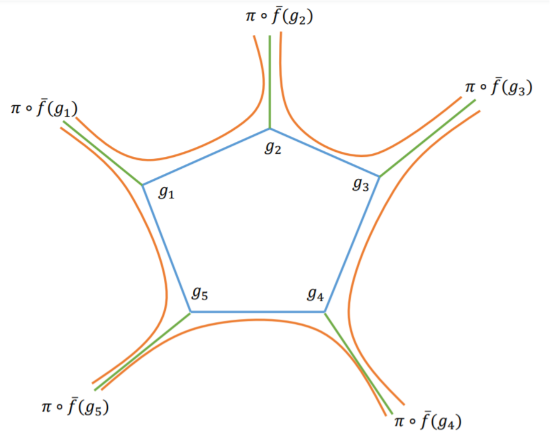
1. g's representative $f(g)$ is **close** to cloud g in G.
2. **most** neighboring clouds $(g_1, g_2) \in E_G$ have **close** representatives in X.
3. f preserves some topological properties of G.

- We have $f : V_G \to V_X$, the representative map.
- Let $\pi : V_X \to V_G$ projection.

Topological Property: $\pi \circ f$ preserves the cycle structure of G.

Algebraic topology intuition: $\pi \circ f$ is a homeomorphism $\Rightarrow$ it preserves the first homology.

Cycle-Homeomorphism

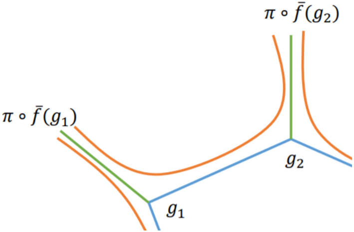


- $\pi : V_X \to V_G$, the natural projection.
- $f : V_G \to V_X$, the representative map.
- f induces a map $\bar{f} : E_G \to \mathbb{F}_2^{E_X}$, "the short path map"

We call f a *Cycle-Homeomorphism* if $\pi \circ \bar{f} : \mathbb{F}_2^{E_G} \to \mathbb{F}_2^{E_G}$ is identity on cycles.

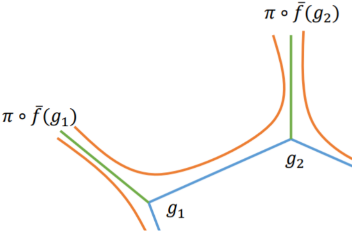
Cycle-Homeomorphism

Let g_1, g_2 neighboring clouds.



Cycle-Homeomorphism

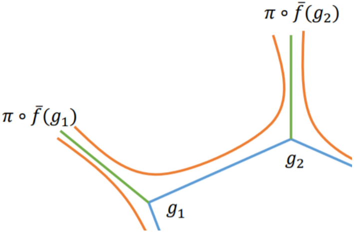
Let g_1, g_2 neighboring clouds.



- g_1, g_2 are 1 hops away.

Cycle-Homeomorphism

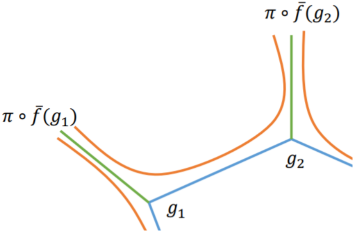
Let g_1, g_2 neighboring clouds.



- g_1, g_2 are 1 hops away.
- $\pi \circ \bar{f}(g_1), g_1$ are $\varepsilon \log(n)$ hops away.
- $\pi \circ \bar{f}(g_2), g_2$ are $\varepsilon \log(n)$ hops away.

Cycle-Homeomorphism

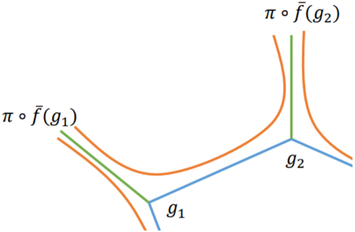
Let g_1, g_2 neighboring clouds.



- g_1, g_2 are 1 hops away.
- $\pi \circ \bar{f}(g_1), g_1$ are $\varepsilon \log(n)$ hops away.
- $\pi \circ \bar{f}(g_2), g_2$ are $\varepsilon \log(n)$ hops away.
- $\pi \circ \bar{f}(g_2), \pi \circ \bar{f}(g_1)$ are $\varepsilon \log(n)$ hops away.

Cycle-Homeomorphism

Let g_1, g_2 neighboring clouds.



- g_1, g_2 are 1 hops away.
- $\pi \circ \bar{f}(g_1), g_1$ are $\varepsilon \log(n)$ hops away.
- $\pi \circ \bar{f}(g_2), g_2$ are $\varepsilon \log(n)$ hops away.
- $\pi \circ \bar{f}(g_2), \pi \circ \bar{f}(g_1)$ are $\varepsilon \log(n)$ hops away.

- The cycle $g_1 \to g_2 \to \pi \circ \bar{f}(g_2) \to \pi \circ \bar{f}(g_1) \to g_1$ has $O(\varepsilon \log(n))$ edges.
- The girth of G has $\Omega(\log(n))$.
- This cycle is trivial.
- f is cycle-homeomorphism.

Splits (probably) Do Not Exist

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions “split”.
- Most graph extensions do not “split”.

Informally:

- We need to choose a vertex “in” each cloud.

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Informally:

- We need to choose a vertex "in" each cloud.
- Neighboring clouds have "neighboring" representatives.

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Informally:

- We need to choose a vertex "in" each cloud.
- Neighboring clouds have "neighboring" representatives.
- We have $|V_G|$ "variables" and $|E_G|$ "constraints".

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Informally:

- We need to choose a vertex "in" each cloud.
- Neighboring clouds have "neighboring" representatives.
- We have $|V_G|$ "variables" and $|E_G|$ "constraints".
- Each variable has "n" possibilities.

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Informally:

- We need to choose a vertex "in" each cloud.
- Neighboring clouds have "neighboring" representatives.
- We have $|V_G|$ "variables" and $|E_G|$ "constraints".
- Each variable has "n" possibilities.
- Each constraint holds with probability of "$1/n$".

Splits (probably) Do Not Exist

Recall Proof Overview:

- Construction of graph extensions.
- Small gap implies that graph extensions "split".
- Most graph extensions do not "split".

Informally:

- We need to choose a vertex "in" each cloud.
- Neighboring clouds have "neighboring" representatives.
- We have $|V_G|$ "variables" and $|E_G|$ "constraints".
- Each variable has "n" possibilities.
- Each constraint holds with probability of "$1/n$".
- If $|E_G| \geq 2|V_G|$, then split should not exist (via union bound).

Splits (probably) Do Not Exist (cont.)

Two issues:

1. All requirements, *e.g.*, “in” and “neighboring”, hold approximately:

Splits (probably) Do Not Exist (cont.)

Two issues:

1. All requirements, *e.g.*, “in” and “neighboring”, hold approximately:
    - Each variable has $n^{1+\varepsilon}$ possibilities.
    - Each constraint holds with probability of $1/n^{1-\varepsilon}$.

Splits (probably) Do Not Exist (cont.)

Two issues:

1. All requirements, *e.g.*, "in" and "neighboring", hold approximately:
   - Each variable has $n^{1+\varepsilon}$ possibilities.
   - Each constraint holds with probability of $1/n^{1-\varepsilon}$.
2. The constraints are not probabilistically independent:

Splits (probably) Do Not Exist (cont.)

Two issues:

1. All requirements, *e.g.*, “in” and “neighboring”, hold approximately:
   - Each variable has $n^{1+\varepsilon}$ possibilities.
   - Each constraint holds with probability of $1/n^{1-\varepsilon}$.
2. The constraints are not probabilistically independent:
   - Define a suitable combinatorial structure that allows enough independence.
   - Linearly independent (modulo 2) cycles imply probabilistic independence.

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

$$\text{number of certificates} \leq n^{(1+O(\varepsilon))n}$$

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

$$\text{number of certificates} \leq n^{(1+O(\varepsilon))n}$$

- Provides enough (almost) independent constraints:

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

$$\text{number of certificates} \leq n^{(1+O(\varepsilon))n}$$

- Provides enough (almost) independent constraints:

  at least $|E_G| - |V_G|$ constraints each satisfied with probability $\leq n^{-(1-O(\varepsilon))}$

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

  $$\text{number of certificates} \leq n^{(1+O(\varepsilon))n}$$

- Provides enough (almost) independent constraints:

  at least $|E_G| - |V_G|$ constraints each satisfied with probability $\leq n^{-(1-O(\varepsilon))}$

Conclusion: no split exists by union bound!

Certificates

A combinatorial structure that satisfies:

- Existence of split $\Rightarrow$ existence of certificate.
- There are not too many certificates:

  $$\text{number of certificates} \leq n^{(1+O(\varepsilon))n}$$

- Provides enough (almost) independent constraints:

  at least $|E_G| - |V_G|$ constraints each satisfied with probability $\leq n^{-(1-O(\varepsilon))}$

Conclusion: no split exists by union bound!

A certificate encodes a "formal roadmap" of:

union of all shortest paths in X between $f(g_1)$ and $f(g_2)$ for $(g_1, g_2) \in E_G$

Certificates - Inner Connected Components

A certificate's core is an Inner Connected Component graph:

Certificates - Inner Connected Components

A certificate's core is an Inner Connected Component graph:

1. Union of all shortest paths in X between $f(g_1)$ and $f(g_2)$ for $(g_1, g_2) \in E_G$.

Certificates - Inner Connected Components

A certificate's core is an Inner Connected Component graph:

1. Union of all shortest paths in X between $f(g_1)$ and $f(g_2)$ for $(g_1, g_2) \in E_G$.
2. Contract all intra-cloud edges.

Certificates - Inner Connected Components

A certificate's core is an Inner Connected Component graph:

1. Union of all shortest paths in X between $f(g_1)$ and $f(g_2)$ for $(g_1, g_2) \in E_G$.
2. Contract all intra-cloud edges.
3. Contract vertices of degree ≤ 2 that do not contain a representative.

Certificates - Inner Connected Components

A certificate's core is an Inner Connected Component graph:

1. Union of all shortest paths in X between $f(g_1)$ and $f(g_2)$ for $(g_1, g_2) \in E_G$.
2. Contract all intra-cloud edges.
3. Contract vertices of degree ≤ 2 that do not contain a representative.

A vertex of the above graph is an Inner Connected Component.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.
- Each component has n^{ε} vertices of X.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.
- Each component has n^{ε} vertices of X.
- Each component has $n^{\varepsilon d}$ "data" (d is its degree).

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.
- Each component has n^{ε} vertices of X.
- Each component has $n^{\varepsilon d}$ "data" (d is its degree).

Edges between components:

- Each edge is a path in X between components.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.
- Each component has n^{ε} vertices of X.
- Each component has $n^{\varepsilon d}$ "data" (d is its degree).

Edges between components:

- Each edge is a path in X between components.
- Each path has at most $\varepsilon \log(n)$ hops.

Certificates - How to Count?

Goal: upper bound the number of inner connected components graphs.

Components:

- Each component is a connected sub-graph of a cloud.
- Each component has n^{ε} vertices of X.
- Each component has $n^{\varepsilon d}$ "data" (d is its degree).

Edges between components:

- Each edge is a path in X between components.
- Each path has at most $\varepsilon \log(n)$ hops.
- Each edge contains $(d_G + d_H)^{\varepsilon \log(n)} = n^{O(\varepsilon)}$ "data".

Certificates - Bounding Probability

Observation

Scanning Inner Connected Components graph:

closing a cycle yields a constraint on a uniform random matching.

Certificates - Bounding Probability

Observation

Scanning Inner Connected Components graph:

closing a cycle yields a constraint on a uniform random matching.

Question: how to upper bound probability of obtaining a certificate?

- Remove a spanning tree.

Certificates - Bounding Probability

Observation

Scanning Inner Connected Components graph:

closing a cycle yields a constraint on a uniform random matching.

Question: how to upper bound probability of obtaining a certificate?

- Remove a spanning tree.
- Remaining edges correspond to disjoint paths.

Certificates - Bounding Probability

Observation

Scanning Inner Connected Components graph:

closing a cycle yields a constraint on a uniform random matching.

Question: how to upper bound probability of obtaining a certificate?

- Remove a spanning tree.
- Remaining edges correspond to disjoint paths.
- Each path closes a cycle “correctly” with probability of $\leq O(1/n^{1-\varepsilon})$.

Certificates - Bounding Probability

Observation

Scanning Inner Connected Components graph:

closing a cycle yields a constraint on a uniform random matching.

Question: how to upper bound probability of obtaining a certificate?

- Remove a spanning tree.
- Remaining edges correspond to disjoint paths.
- Each path closes a cycle “correctly” with probability of $\leq O(1/n^{1-\varepsilon})$.
- Each closed cycle is correct "independently".

Certificates - Bounding Probability (cont.)

Let χ be the Euler characteristic of the Inner Connected Components graph:

Certificates - Bounding Probability (cont.)

Let χ be the Euler characteristic of the Inner Connected Components graph:

- $\Pr_{X\sim\text{Ext}(G,H)}\left[\text{certificate}\right] = n^{-\chi\cdot(1-O(\varepsilon))}$.

Certificates - Bounding Probability (cont.)

Let χ be the Euler characteristic of the Inner Connected Components graph:

- $\Pr_{X \sim \text{Ext}(G,H)} [\text{certificate}] = n^{-\chi \cdot (1 - O(\varepsilon))}$.
- f is a cycle-homeomorphism $\implies$ the cycles space of the Inner Connected Components graph is larger than the cycles space of G.

Certificates - Bounding Probability (cont.)

Let χ be the Euler characteristic of the Inner Connected Components graph:

- $\Pr_{X \sim \mathrm{Ext}(G,H)}[\text{certificate}] = n^{-\chi \cdot (1 - O(\varepsilon))}$.
- f is a cycle-homeomorphism $\implies$ the cycles space of the Inner Connected Components graph is larger than the cycles space of G.
- $\chi \geq |E_G| - |V_G| = \left(\frac{d_G}{2} - 1\right) n > n$.

Certificates - Bounding Probability (cont.)

Let χ be the Euler characteristic of the Inner Connected Components graph:

- $\Pr_{X\sim\mathrm{Ext}(G,H)}[\text{certificate}] = n^{-\chi\cdot(1-O(\varepsilon))}$.
- f is a cycle-homeomorphism $\implies$ the cycles space of the Inner Connected Components graph is larger than the cycles space of G.
- $\chi \geq |E_G| - |V_G| = \left(\frac{d_G}{2} - 1\right) n > n$.

We are done by a union bound as:

$$\underbrace{n^{-\chi\cdot(1-O(\varepsilon))}}_{\text{probability}} \cdot \underbrace{n^{(1+O(\varepsilon))}}_{\text{no. certificates}} \leq n^{-(\chi-n)(1-O(\varepsilon))} \ll 1$$

Questions?