Insertion time of random walk cuckoo hashing

Alan Frieze
Tony Johansson

Department of Mathematical Sciences,

Carnegie Mellon University,

Pittsburgh, USA.

On-line Bipartite Matching

We are given a bipartite graph $G = (L + R, E)$ in an on-line manner.

$L = \{v_1, v_2, \ldots, v_n\}$ and $R = \{w_1, w_2, \ldots, w_m\}$.

Vertices of L are presented one at a time along with their edges to R.

On-line Bipartite Matching

We are given a bipartite graph $G = (L + R, E)$ in an on-line manner.

$L = \{v_1, v_2, \dots, v_n\}$ and $R = \{w_1, w_2, \dots, w_m\}$.

Vertices of L are presented one at a time along with their edges to R.

If $L_k = \{v_1, v_2, \dots, v_k\}$ then the aim is to maintain a matching M_k of L_k into R.

Assuming a matching of L into R exists we wish to control the lengths of the augmenting path needed to replace M_{k-1} by M_k.

On-line Bipartite Matching

Has applications in Streaming Content Delivery; Web Hosting; Job Scheduling; Hashing.

Chaudhuri, Daskalakis, Kleinberg, Lin (2009): Arbitrary G with $|L| = |R| = n$. If L arrives in random order and one always uses a shortest augmenting path then the total length of augmenting paths needed is at most $n \log n$.

Cuckoo Hashing

Each $v \in L$ chooses a set $N(v)$ of $d \geq 2$ random neighbors in R to create a graph Γ.

Aim is to construct a matching M_n from L into R.

Cuckoo Hashing

Each $v \in L$ chooses a set $N(v)$ of $d \geq 2$ random neighbors in R to create a graph Γ.

Aim is to construct a matching M_n from L into R.

Natural questions:
(a) How large should $m = |R|$ be to ensure the existence of a matching w.h.p.? Solved.
(b) How long does it take to construct M_n? Subject of talk.

Related to Cuckoo Hashing: $N(v)$ are the values of d different hash functions. Pagh and Rodler (2004)

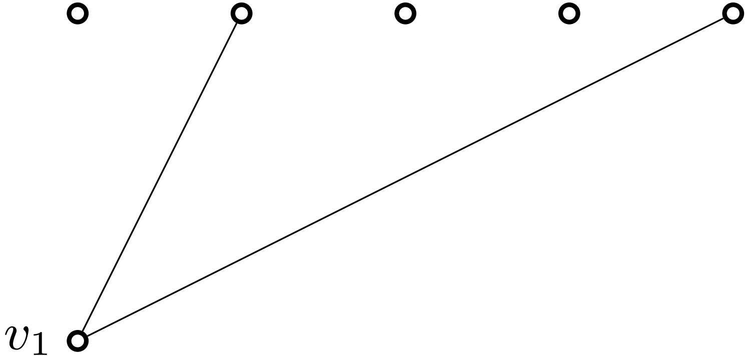
In Cuckoo Hashing, there are d different hash functions $f_1, f_2, \ldots, f_d$.

When a new item x is to be added to the table, we see if one of the hash functions maps it to an empty location. If so we place x into this location.

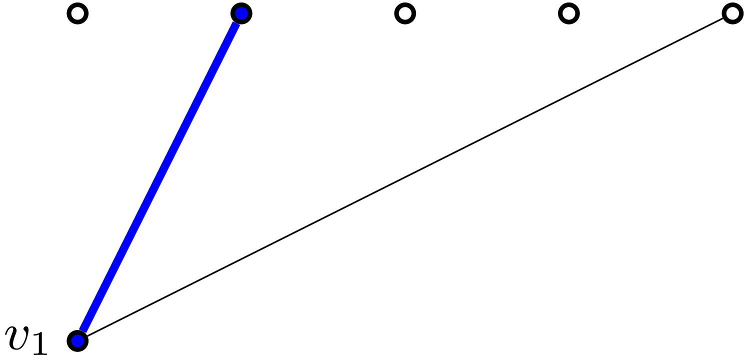
If not, x is mapped to $X = f_i(x)$ for some random $i \in [d]$, and the y which was previously mapped to X is mapped to $f_j(y) \neq X$ for some random $j \in [d]$. Repeat if necessary.

In other words we take a random walk to find an augmenting path.

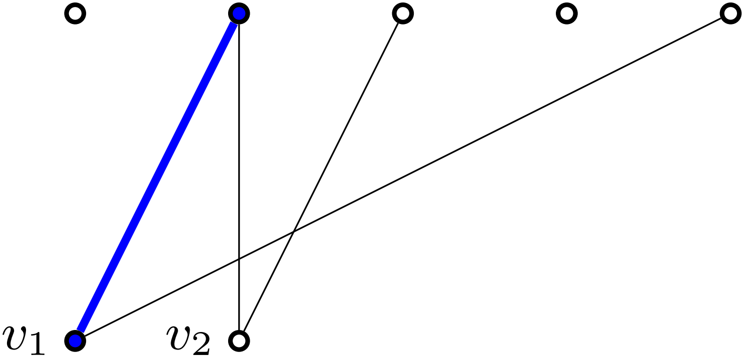
Cuckoo Hashing



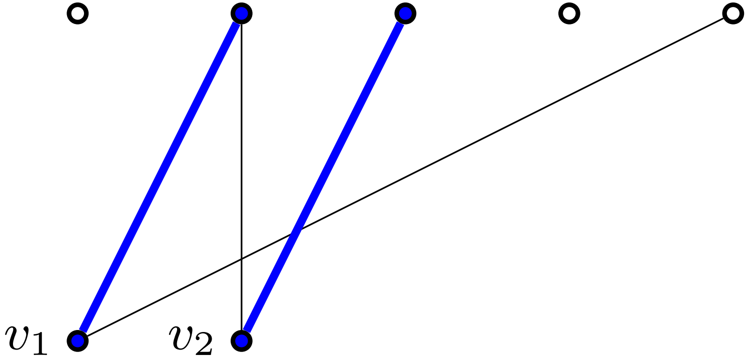
Cuckoo Hashing



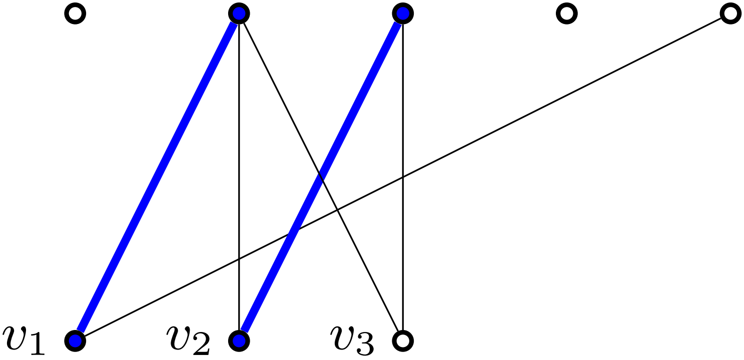
Cuckoo Hashing



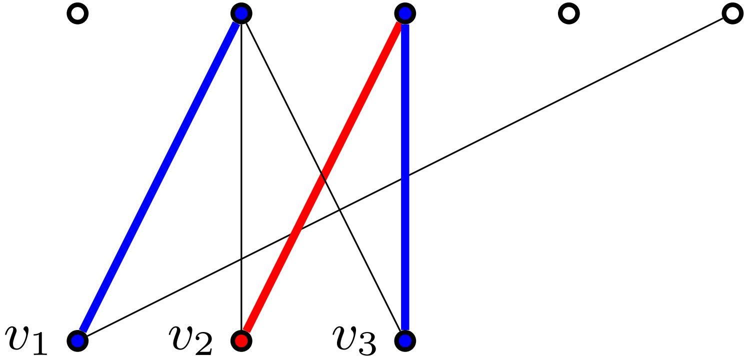
Cuckoo Hashing



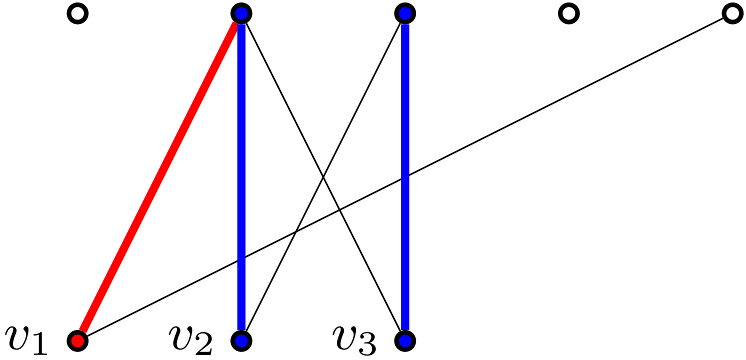
Cuckoo Hashing

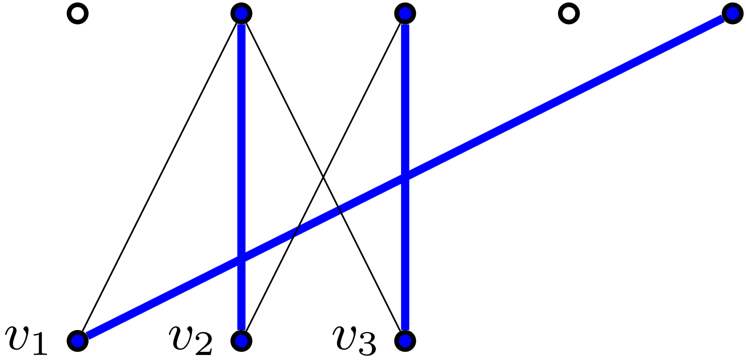


Cuckoo Hashing



Cuckoo Hashing




v_1
v_2
v_3

Cuckoo Hashing

What should $m = |R|$ and d be for a matching to exist?

Trivially $m \geq n = |L|$ is needed.

If $d = 2$ then we need $m > 2n$ (Not hard to prove).

Cuckoo Hashing

What should $m = |R|$ and d be for a matching to exist?

Trivially $m \geq n = |L|$ is needed.

If $d = 2$ then we need $m > 2n$ (Not hard to prove).

More complicated for $d \geq 3$. Solved by Frieze and Melsted (2012) and by Fountoulakis and Panagiotou (2012).

In general if $n = (1 - \varepsilon)m$ then we need $d \gg \log \frac{1}{\varepsilon}$

Cuckoo Hashing

We now consider the time to build M_n by using random walks to find augmenting paths.
M_{k-1} is a matching of $L_{k-1} = \{v_1, \ldots, v_{k-1}\}$ into $R_{k-1} \subseteq R$.
The edges of M_{k-1} are $\{\{v, \phi_{k-1}(v)\} : v \in L_{k-1}\}$.

We now consider the time to build M_n by using random walks to find augmenting paths.
M_{k-1} is a matching of $L_{k-1} = \{v_1, \ldots, v_{k-1}\}$ into $R_{k-1} \subseteq R$.
The edges of M_{k-1} are $\{\{v, \phi_{k-1}(v)\} : v \in L_{k-1}\}$.

Algorithm INSERT:

Step 1 $x \leftarrow v_k$; $M \leftarrow M_{k-1}$;

Step 2 **If** $S_k(x) = N(x) \cap \bar{R}_{k-1} \neq \emptyset$ **then** choose y randomly from $S_k(x)$ and let $M_k = M \cup \{\{x, y\}\}$, **else**

Step 3 Choose y randomly from $N(x)$;

Step 4 $M \leftarrow M \cup \{\{x, y\}\} \setminus \left\{y, \phi_{k-1}^{-1}(y)\right\}$; $x \leftarrow \phi_{k-1}^{-1}(y)$; **goto** Step 2.

Cuckoo Hashing

Let P_k denote the augmenting path that inserts v_k into the matching.

Earlier results: Frieze, Melsted and Mitzenmacher (2011) and Fountoulakis, Panagiotou and Steger (2013):
The expected time for INSERT to reach $\bar{R}_{k-1}$ is $O((\log n)^{2+\varepsilon_d})$.

Let P_k denote the augmenting path that inserts v_k into the matching.

Earlier results: Frieze, Melsted and Mitzenmacher (2011) and Fountoulakis, Panagiotou and Steger (2013):
The expected time for INSERT to reach $\bar{R}_{k-1}$ is $O((\log n)^{2+\varepsilon_d})$.

Theorem

Assume that $n = (1-\varepsilon)m$ *and* $5d^2(1-\varepsilon)^{d/2} \leq (1-\alpha)(d-1)$ *then*

$$\mathbf{E}(|P_k|) \leq 1 + \frac{2}{\alpha} \text{ for } k = 1, 2, \ldots, n.$$

This forces $d \gg \frac{1}{\varepsilon} \log \frac{1}{\varepsilon}$ (compare to $d \gg \log \frac{1}{\varepsilon}$)

Let

$$B = \{ v \in L : N(v) \cap \bar{R}_{k-1} = \emptyset \} .$$

If $x \notin B$ in Step 2 of INSERT then we will have found P_k.

Let $P = (x_1, y_1, x_2, y_2, \ldots, x_\ell)$ be a path in Γ, where $x_1, x_2, \ldots, x_\ell \in L$ and $y_1, y_2, \ldots, y_{\ell-1} \in R$. We say that P is interesting if $x_1, x_2, \ldots, x_\ell \in B$.

Let

$$B = \{v \in L : N(v) \cap \bar{R}_{k-1} = \emptyset\}.$$

If $x \notin B$ in Step 2 of INSERT then we will have found P_k.

Let $P = (x_1, y_1, x_2, y_2, \ldots, x_\ell)$ be a path in Γ, where $x_1, x_2, \ldots, x_\ell \in L$ and $y_1, y_2, \ldots, y_{\ell-1} \in R$. We say that P is interesting if $x_1, x_2, \ldots, x_\ell \in B$.

Removing the last two edges from an augmenting path chosen by our algorithm will always yield an interesting path.

Cuckoo Hashing

Let

$$B = \left\{ v \in L : N(v) \cap \bar{R}_{k-1} = \emptyset \right\}.$$

If $x \notin B$ in Step 2 of INSERT then we will have found P_k.

Let $P = (x_1, y_1, x_2, y_2, \ldots, x_\ell)$ be a path in Γ, where $x_1, x_2, \ldots, x_\ell \in L$ and $y_1, y_2, \ldots, y_{\ell-1} \in R$. We say that P is interesting if $x_1, x_2, \ldots, x_\ell \in B$.

Removing the last two edges from an augmenting path chosen by our algorithm will always yield an interesting path.

If there are few long interesting alternating paths, then a short randomly chosen alternating path is likely to be augmenting.

Cuckoo Hashing

Two approaches:

A: It is not too hard to estimate the number of interesting paths of length $2\ell - 1$.

B: It is harder to estimate the number of interesting alternating paths, because this depends on the matching M_{k-1}.

We pay a price in d because we can do A but not B.

Cuckoo Hashing

Let $\nu_{k,\ell}$ denote the number of interesting paths with $2\ell - 1$ vertices. Given A_0 and d sufficiently large, $2 \leq \ell \leq A_0 \log\log n$,

Claim 1:

$$\mathbf{Pr}\left(\nu_{k,\ell} \geq (1+\alpha)(5d^2(1-\varepsilon)^{d/2})^{\ell-1}k\right) = o(n^{-2}).$$

Compare to number of possible random walks: $(d-1)^\ell k$, and recall $5d^2(1-\varepsilon)^{d/2} \leq (1-\alpha)(d-1)$.

Let $p_{k,\ell}$ denote the probability that INSERT requires at least ℓ rounds to insert v_k.

Claim 2:

$$\mathbf{E}(|P_k|) = 1 + 2\sum_{\ell=2}^{\infty} p_{k,\ell} \le 1 + \frac{2}{\alpha}.$$

Cuckoo Hashing

Let $p_{k,\ell}$ denote the probability that INSERT requires at least ℓ rounds to insert v_k.

Claim 2:

$$\mathbf{E}(|P_k|) = 1 + 2\sum_{\ell=2}^{\infty} p_{k,\ell} \leq 1 + \frac{2}{\alpha}.$$

We show Claim 2 first, using Claim 1. From previous papers:

$$\sum_{\ell=A_0 \log\log n}^{\infty} p_{k,\ell} = o(1)$$

Cuckoo Hashing

Recall $5d^2(1-\varepsilon)^{d/2} \leq (1-\alpha)(d-1)$.

$$\sum_{\ell=2}^{A_0 \log\log n} p_{k,\ell} \leq o(1) + \sum_{\ell=2}^{A_0 \log\log n} \frac{\nu_{k,\ell}}{(d-1)^\ell k}$$

$$[\text{Claim 1}] \leq o(1) + \sum_{\ell=2}^{A_0 \log\log n} \frac{(1+\alpha)(5d^2(1-\varepsilon)^{d/2})^{\ell-1} k}{(d-1)^\ell k}$$

$$\leq o(1) + (1+\alpha)\sum_{\ell=2}^{\infty}(1-\alpha)^{\ell-1}$$

$$= o(1) + \frac{1-\alpha^2}{\alpha}.$$

Need to verify Claim 1:

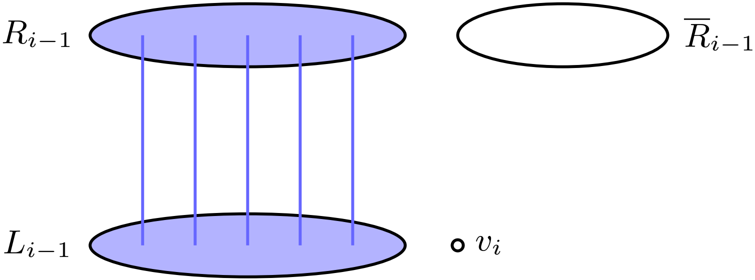
$$\mathbf{Pr}\left(\nu_{k,\ell} \geq (1+\alpha)(5d^2(1-\varepsilon)^{d/2})^{\ell-1}k\right) = o(n^{-2}).$$

First we show that $|B|$ is small w.h.p.:

$$\mathbf{Pr}(|B| \geq 5(1-\varepsilon)^{d/2}k) = O(e^{-\Omega(n^{1/2})}).$$

Cuckoo Hashing

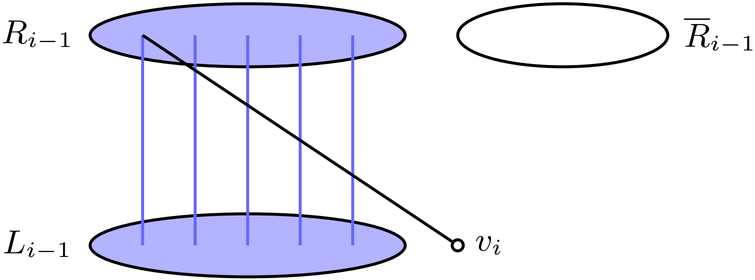
Expose edges one by one until $\overline{R}_{i-1}$ reached or the d edges are exhausted.



Partition B by counting the number of exposures before $\overline{R}_{i-1}$ is reached, if it is.

Cuckoo Hashing

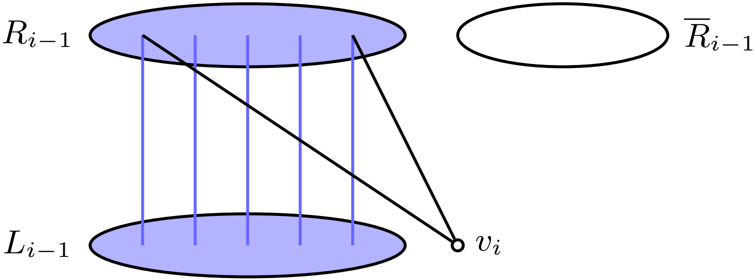
Expose edges one by one until $\overline{R}_{i-1}$ reached or the d edges are exhausted.



Partition B by counting the number of exposures before $\overline{R}_{i-1}$ is reached, if it is.

Cuckoo Hashing

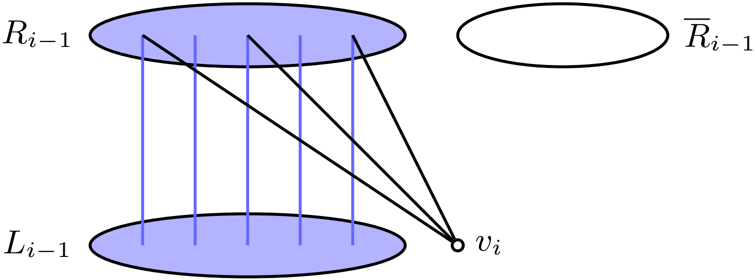
Expose edges one by one until $\overline{R}_{i-1}$ reached or the d edges are exhausted.



Partition B by counting the number of exposures before $\overline{R}_{i-1}$ is reached, if it is.

Cuckoo Hashing

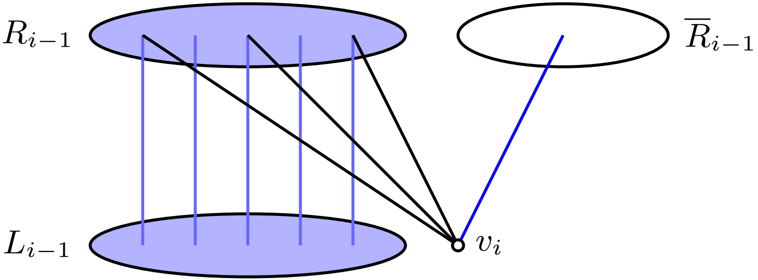
Expose edges one by one until $\overline{R}_{i-1}$ reached or the d edges are exhausted.



Partition B by counting the number of exposures before $\overline{R}_{i-1}$ is reached, if it is.

Cuckoo Hashing

Expose edges one by one until $\overline{R}_{i-1}$ reached or the d edges are exhausted.



Partition B by counting the number of exposures before $\overline{R}_{i-1}$ is reached, if it is.

Cuckoo Hashing

$B = B_1 \cup B_2 \cup B_3 \cup B_4$ where

$B_1 = \{v_i \in B : \text{round } i \text{ exposes at least } d/2 \text{ edges incident with } v_i\}$.

Chernoff bounds show
$\mathbf{Pr}\left(|B_1| \geq 2k(1-\varepsilon)^{d/2}\right) = O(e^{-\Omega(n^{1/2})})$.

Cuckoo Hashing

$B = B_1 \cup B_2 \cup B_3 \cup B_4$ where

$B_1 = \{v_i \in B : \text{round } i \text{ exposes at least } d/2 \text{ edges incident with } v_i\}$.

Chernoff bounds show
$\mathbf{Pr}\left(|B_1| \geq 2k(1-\varepsilon)^{d/2}\right) = O(e^{-\Omega(n^{1/2})})$.

$B_2 = \{v_i \in B : \text{ round } i \text{ does not end in Step 2 with } x = v_i.\}$

We have $\mathbf{E}(|B_2|) \leq \frac{k^{d+1}}{(d+1)m^d}$ and we can use Hoeffding inequality.

Cuckoo Hashing

$$B_3 = \{v_i \in B : \exists \ell \leq k, \ell \neq i \text{ s.t. round } \ell \text{ ends with } x = v_i\}$$

We have $|B_3| \leq |B_2|$ since then $\ell \in B_2$.

$$B_3 = \{v_i \in B : \exists \ell \leq k, \ell \neq i \text{ s.t. round } \ell \text{ ends with } x = v_i\}$$

We have $|B_3| \leq |B_2|$ since then $\ell \in B_2$.

$B_4 = B \setminus (B_1 \cup B_2 \cup B_3)$. If $v_i \in B_4$ then

- round i reached $\overline{R}_{i-1}$ in $\leq d/2$ queries,
- no other round $\ell \leq k$ ended at v_i, and
- all neighbors of v_i are in R_{k-1}.

The probability of this is at most $(1-\varepsilon)^{d/2}$, so
$E(|B_4|) \leq (1-\varepsilon)^{d/2} k$. Apply Chernoff.

Cuckoo Hashing

Let $\gamma = 5(1-\varepsilon)^{d/2}$ so that $|B| \leq \gamma k$ w.h.p.

$$\begin{aligned}\mathbf{E}(\nu_{k,\ell}) = \mathbf{E}(\nu_{k,\ell} \mid |B| \leq \gamma k)\mathbf{Pr}(|B| \leq \gamma k) + \\ \mathbf{E}(\nu_{k,\ell} \mid |B| > \gamma k)\mathbf{Pr}(|B| > \gamma k) \\ \leq k^{\ell}\gamma^{\ell}k^{\ell-1} \cdot \left((1+o(1))\frac{d}{k}\right)^{2\ell-2} + O(k^{2\ell-1} \cdot e^{-\Omega(n^{1/4})}), \\ \leq (1+o(1))k\gamma(d^2\gamma)^{\ell-1} + o(1).\end{aligned}$$

Middle inequality needs some care.
Now we use Azuma-Hoeffding to show concentration.

It follows that whp,

$$\nu_{k,\ell} < (1+\alpha)\gamma^{\ell}d^{2\ell-2}k$$

Open Questions:

- Find the correct dependence of d on ε.
- Do an analysis of $d = 3$ and $1/2 < \varepsilon < 1$.

THANK YOU