# Balls and Bins: Smaller Hash Families and Faster Evaluation

**Elisa Celis**

China Theory Week 2011

*Joint work with*

Omer Reingold Gil Segev Udi Wieder

# The Setup

- Throw n balls into n bins uniformly at random.
- Equivalently, consider a family F containing all functions $f : [n] \rightarrow [n]$. Choose f uniformly at random.
- In fact, consider a family F containing all functions $f : U \rightarrow [n]$. Let an adversary specify a subset of U of size n. Choose f uniformly at random.
- With high probability, the maximum load is at most $O(\log n / \log\log n)$.
- In fact, the best high probability guarantee on the maximum load must be $\Omega(\log n / \log\log n)$.
- However, in most applications the description length and/or evaluation time of the functions of F are ignored in the analysis.

# The Problem

- Define an explicit family F of hash functions $f : U \rightarrow [n]$ such that $\forall\ V \subseteq U$ of size n, when f is chosen uniformly at random from F, the maximum load of a bin is O(log n / loglog n) with high probability.
- Want to minimize the *description length* and *evaluation time* of a family F that satisfies the above condition.
  - e.g. for the complete family F, the description length is O(|U| log n), and evaluation time is O(1)

# Using k-wise Independence

- A family F with $f : U \to V$ is k-wise independent if for any $x_1, x_2, \ldots, x_k \in U$, when f is chosen uniformly from F, the distribution of $f(x_1), f(x_2), \ldots, f(x_k)$ is the uniform distribution on $V^k$.

- Given a family of O(log n / loglog n)-wise independent functions the maximum load is O(log n / loglog n) with high probability.

  Classical results show such the size is $\Omega(\log^2 n/\log\log n)$ and constructions attain evaluation time $O(\log n/\log\log n)$.

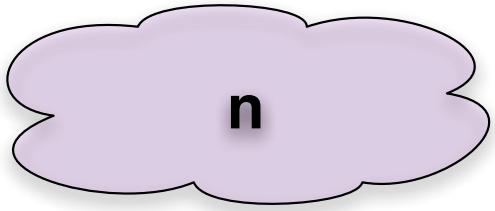
- Siegel's space-time tradeoff says a k-wise independent family either has evaluation time $k \leq T$ or size $n^{1/T} \leq S$.

- We show a lower bound on the size of $\Omega(\log n)$, which gives $O(\log n/\log\log n) \leq T$.

# Previous Results

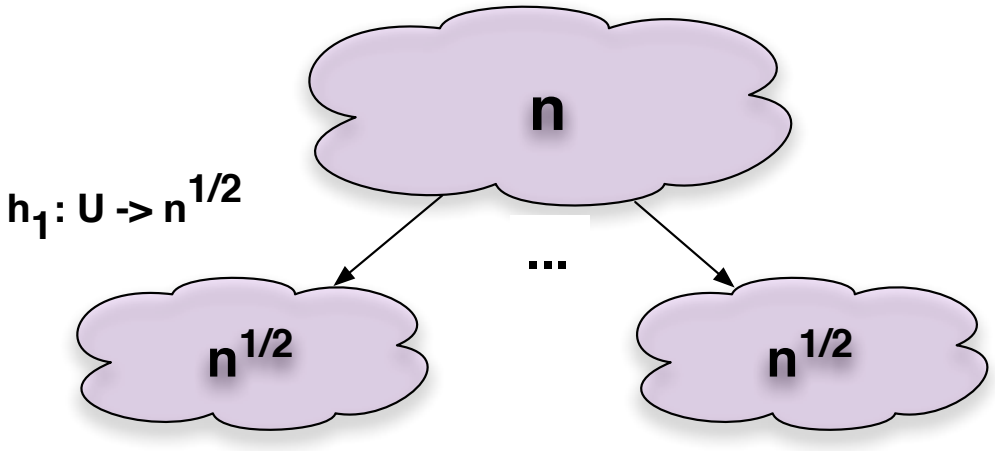
| | Size (bits) | Evaluation Time |
|---|---|---|
| Lower Bound<br>(Celis et. al. '11) | $\Omega(\log n)$ | $\Omega(\log n/\log\log n)$ |
| Full Independence | $\Omega(\lvert U\rvert \log n)$ | $O(1)$ |
| $n^{\varepsilon}$-wise independence<br>(Siegel '98, Tabluation Hashing '11, etc) | $O(n^{\varepsilon'})$ | $O(1)$ |
| log n/loglog n -wise independence<br>(classical construction) | $O(\log^2 n/\log\log n)$ | $O(\log n/\log\log n)$ |

# Split-Hashing: Construction 1

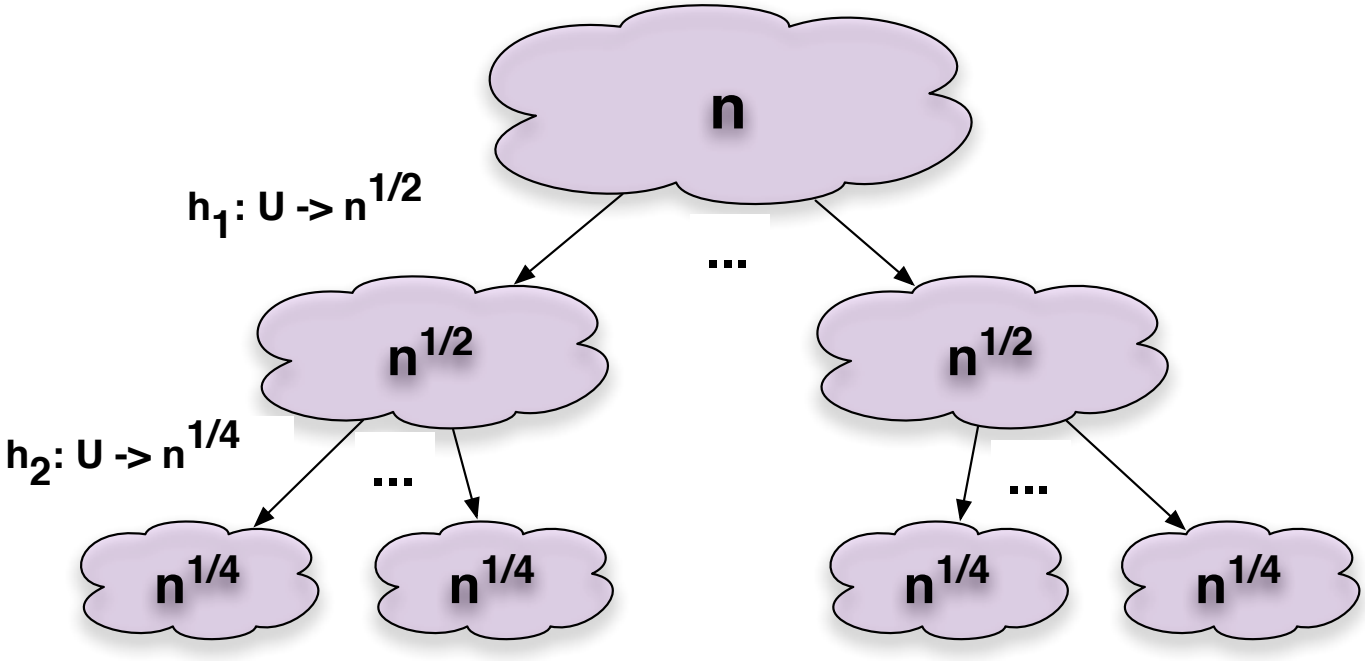



$h_1 : U \rightarrow n^{1/2}$

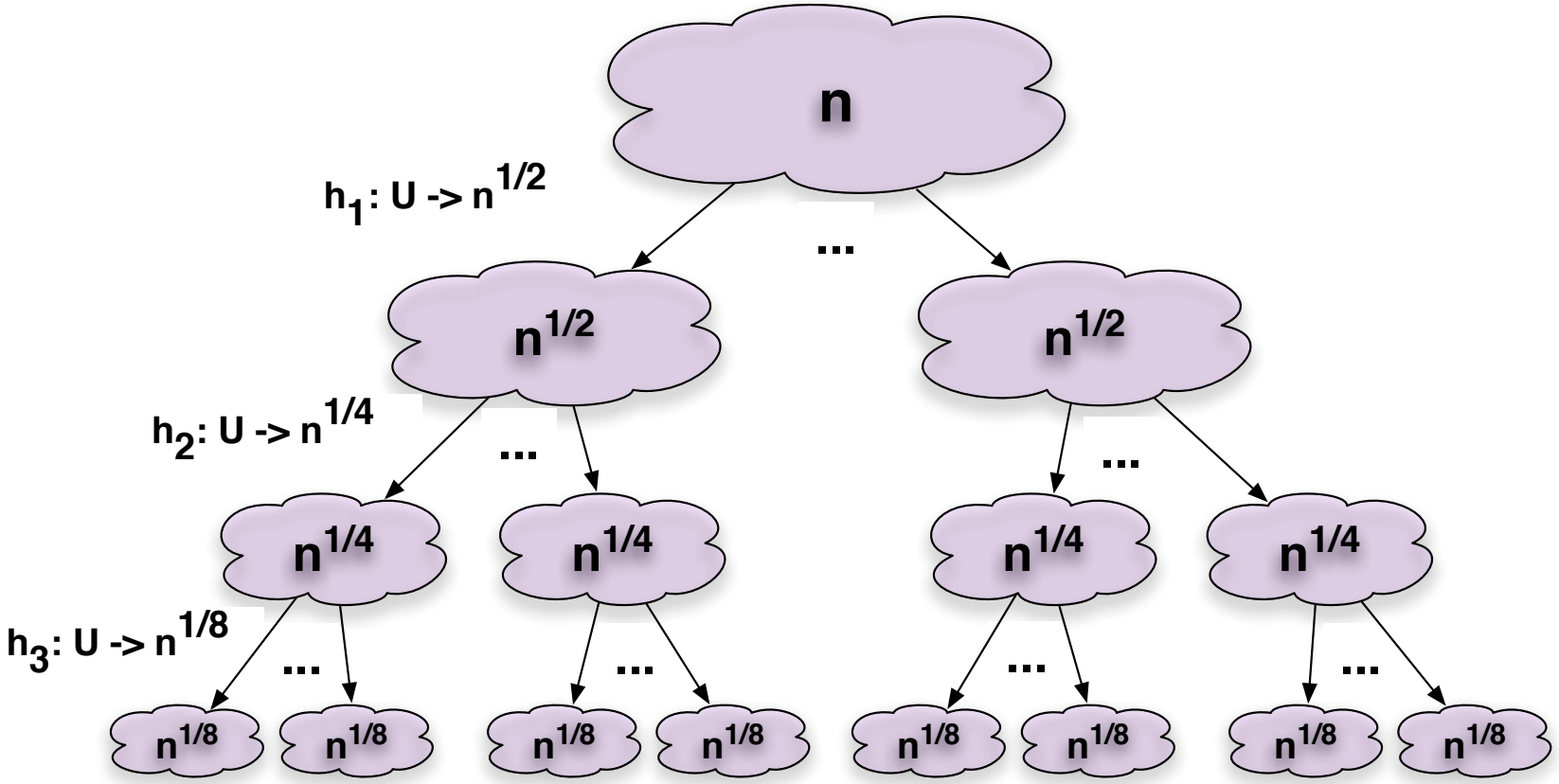
# Splashing: Construction 1



# Splashing: Construction 1



# Splashing: Construction 1



# Limited Independence

A family of functions F with $f : U \to V$ is

- k-wise independent if for any $x_1, x_2, \ldots, x_k \in U$, when f is chosen uniformly from F, the distribution of $f(x_1), f(x_2), \ldots, f(x_k)$ is the uniform distribution on $V^k$.
- k-wise δ-dependent if for any $x_1, x_2, \ldots, x_k \in U$, when f is chosen uniformly from F, the total variation distance between $f(x_1), f(x_2), \ldots, f(x_k)$ and the uniform distribution on $V^k$ is at most δ.

There is a construction for k-wise δ-dependent functions with $O(k \log|V|)$ bits and $O(\log U + k \log v + \log(1/\delta))$ evaluation time [Alon et al '92].

- Goal: Carefully balance amount of limited independence used to get small error and small representation size.

# Construction

Consider d families of functions where each $h_i : U \to n_i$ is $k_i$-wise δ-dependent. Let $h : U \to [n]$ be the concatenation the above functions where each $h_i$ is chosen uniformly from its family.

$$h(x) = h_1(x), h_2(x), \ldots, h_d(x)$$

- $n_0 = n$
  $n_i = (n_{i-1})^{3/4}$ for every $i \in [d-1]$ and
  $\log(n_d) = \log n - \Sigma_{i=1:d-1} \log n_i$.
- $k_i \log(n_i) = \Theta(\log n)$ for every $i \in [d-1]$, and
  $k_d = \Theta(\log n / \log \log n)$.
- $\delta = \text{poly}(1/n)$.
- $d = O(\text{loglog } n)$.

Note: We gradually increase independence.

# Representation Size & Evaluation Time

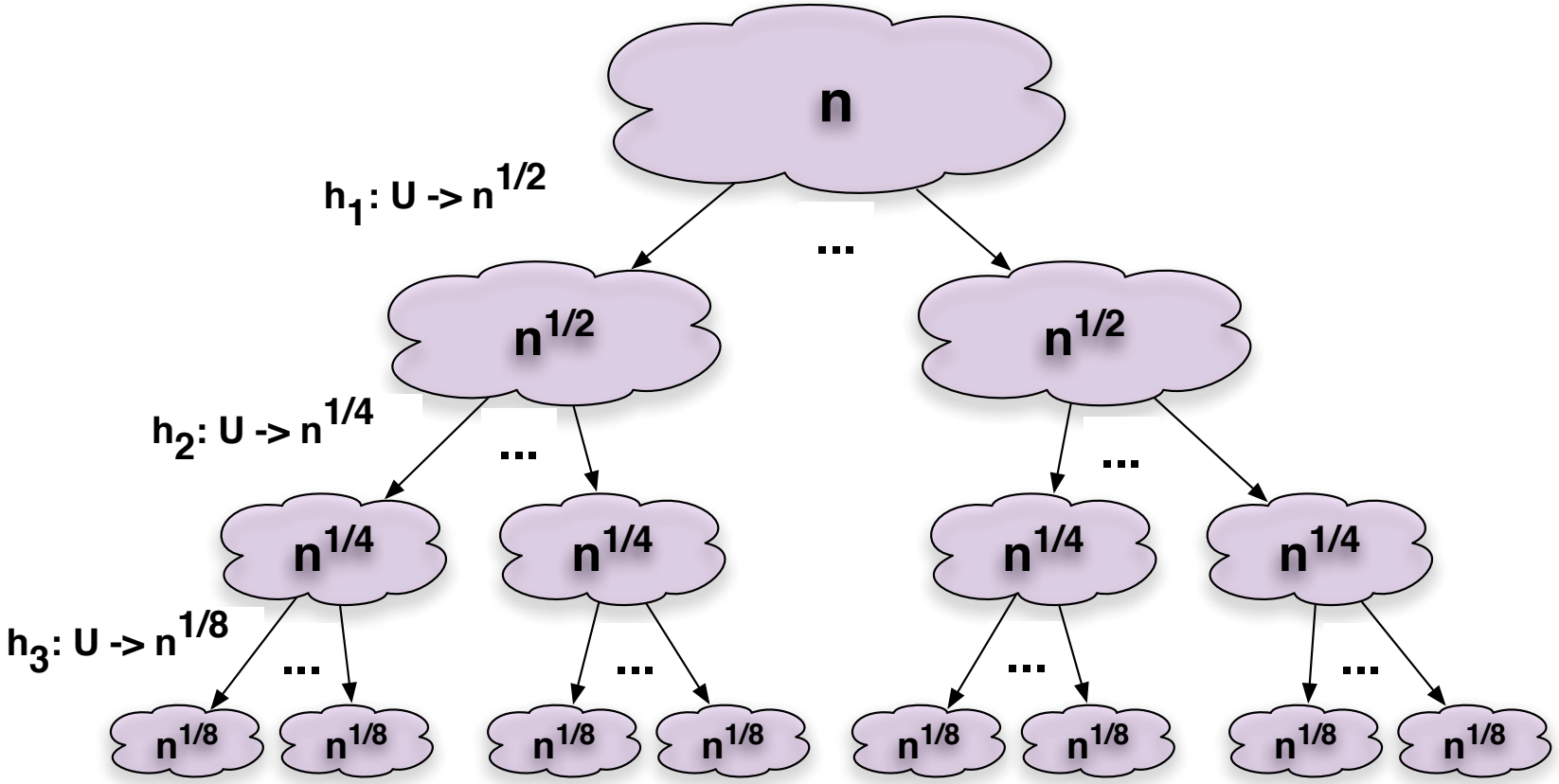
Construction:

- $n_0 = n$, and $n_i = (n_{i-1})^{3/4}$ for every $i \in [d\text{-}1]$ and $\log(n_d) = \log n - \Sigma_{i=1:d-1} \log n_i$.
- $k_i \log(n_i) = \Theta(\log n)$ for every $i \in [d-1]$, and $k_d = \Theta(\log n / \log \log n)$.
- $\delta = \text{poly}(1/n)$.
- $d = O(\text{loglog } n)$.

Recall the construction for $k_i$-wise $\delta$-dependence requires $O(k_i \log n_i)$ bits and $O(\log U + k_i \log n_i + \log(1/\delta))$ evaluation time. We use $O(\text{loglog } n)$ such functions.

This gives $O(\log n \text{ loglog } n)$ overall for both space and time (with some careful accounting for level d)!

# Splashing: Construction 1



# Correctness

Proof sketch:

- First show that for $\alpha \in \Omega(1/\text{loglog } n)$ and $0 \le \alpha_i \le 1$, any set $|S_i| \le (1 + \alpha_i)\, n_i$ has
  $$\Pr_{hi}[\text{ max load} \le (1+\alpha)(1+\alpha_i)\, n_{i+1}] > 1-1/n^c$$
- Now, show by induction that with high probability the maximal load at level d-1 is
  $$(1+\alpha)^{d-1} n_{d-1} \le 2n_{d-1} \le 2 \log n.$$
- Bound the probability that more than log n / loglog n balls are hashed into a bin by $h_d$, and apply a union bound to complete.

# A Useful Tail Bound

**Lemma:**

Let $X_1, \ldots, X_n \in \{0, 1\}$ be $2k$-wise $\delta$-dependent random variables for $k \in \mathbb{N}$ and $0 \leq \delta < 1$. Let $X = \Sigma_{i=1:n} X_i$, $\mu = E[X]$. For any $t > 0$ :

$$\Pr[|X - \mu| > t] \leq 2(2nk/t^2)^k + \delta(n/t)^{2k}$$

**Proof sketch:**

Apply Markov's inequality and expand to get:

$$\Pr[|X - \mu| > t] \leq \Pr[(X - \mu)^{2k}] / t^{2k}$$
$$\leq \Sigma_{S \subseteq [n],\, |S|=2k} \Pr[\Pi_{i \in S}(X_i - \mu_i)] / t^{2k}$$

By definition, $\Pi_{i \in S}(X_i - \mu_i) \leq \Pi_{i \in S}(X'_i - \mu_i) + \delta$ where the $X'_i$s are *fully* independent with the *same marginals* as $X_i$s, which gives:

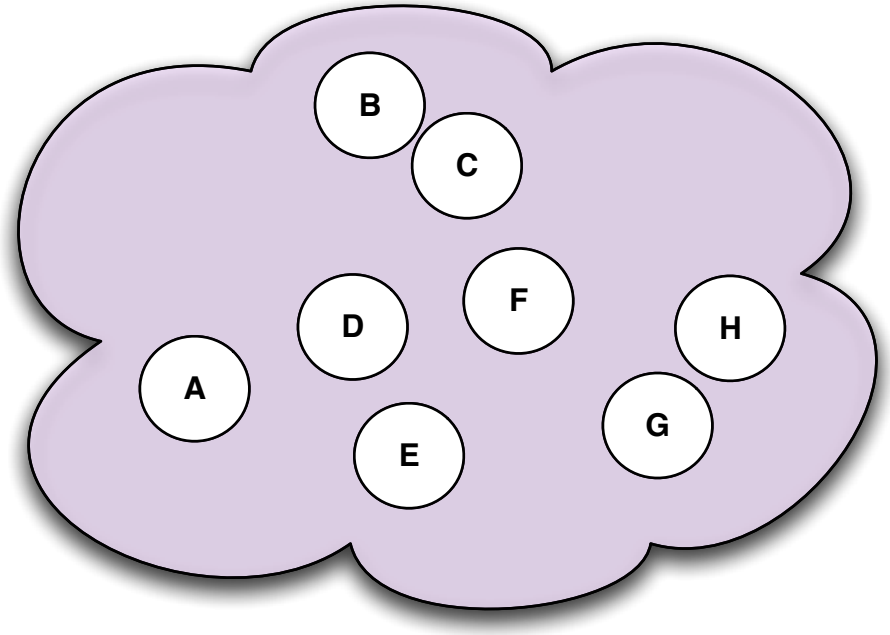
$$\leq (\Pr[(X' - \mu)] + \delta n^{2k}) / t^{2k}$$

Analyze $\Pr[(X' - \mu)^{2k}] / t^{2k}$ (now independent) to get the bound.

# Comparison of Results

| | Size (bits) | Evaluation Time |
|---|---|---|
| Lower Bound | $\Omega(\log n)$ | $\Omega(\log n/\log\log n)$ |
| Full Independence | $\Omega(\lvert U\rvert \log n)$ | $O(1)$ |
| $n^{\varepsilon}$-wise independence | $O(n^{\varepsilon'})$ | $O(1)$ |
| log n/loglog n -wise independence | $O(\log^2 n/\log\log n)$ | $O(\log n/\log\log n)$ |
| Construction 1 | $O(\log n \log\log n)$ | $O(\log n \log\log n)$ |

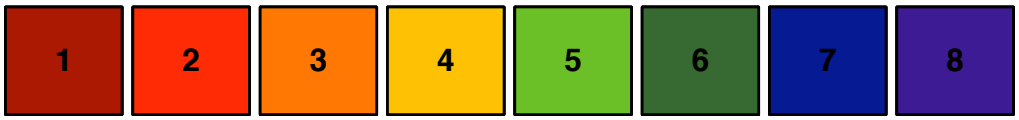
# Split-Hashing: Revisited



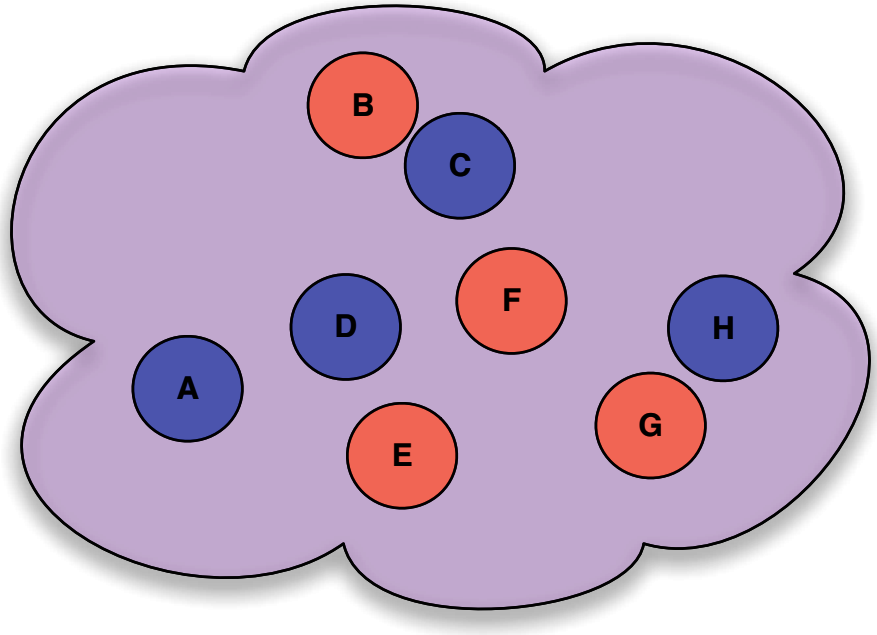

Define h : U → ■■ and

$g_{■}$, $g_{■}$ : U → ■■■■■■■■

Define $f(x) = g_{h(x)}(x)$



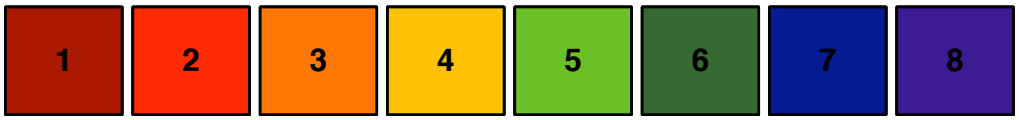
# Splishing: Construction 2



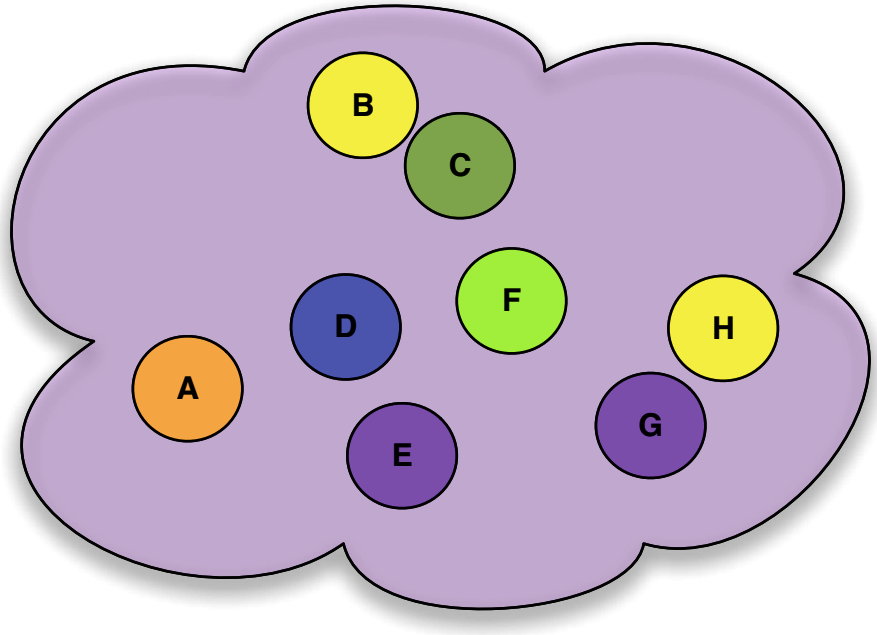

Define h : U → ■■ and

$g_{■}$, $g_{■}$ : U → ■■■■■■■■

Define $f(x) = g_{h(x)}(x)$



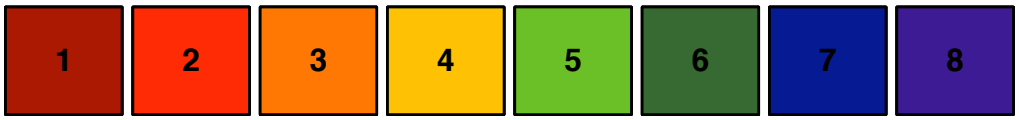
# Splishing: Construction 2




Define h : U → ■■ and

$g_{■}, g_{■}$ : U → ■■■■■■■■

Define $f(x) = g_{h(x)}(x)$



# Construction

Define a function $h : U \to \ell$ for $\ell = O(2^{\sqrt{\log n}})$ that is c-wise independent, and functions $g_1, \ldots, g_\ell$ that are $O(\log^{1/2} n)$-wise independent. For any $x \in U$ define

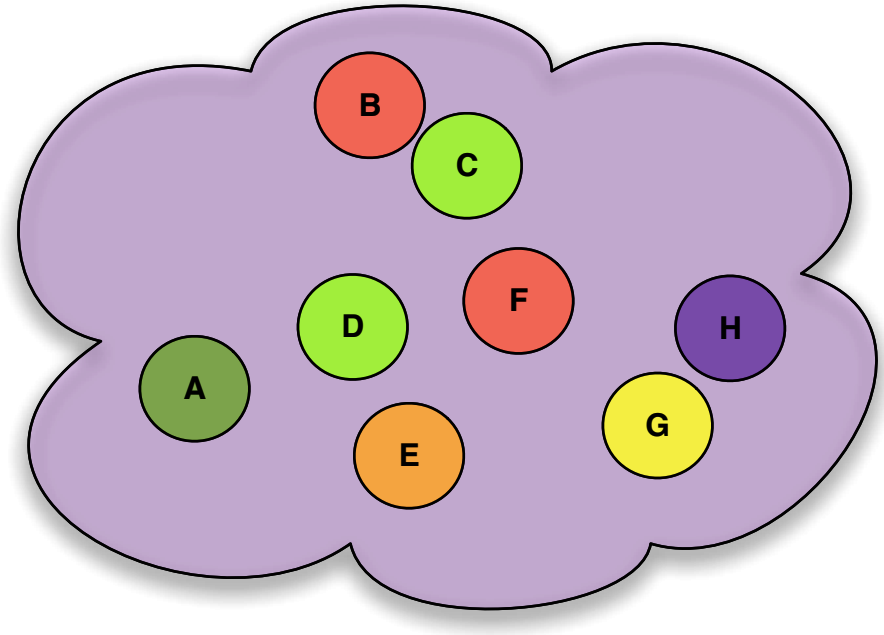
$$f(x) = g_{h(x)}(x) .$$

- By sampling independently this has size $O(\ell \log^{3/2} n)$
- Instead, obtain as the output of an explicit pseudorandom generator for space-bounded computations using a seed of length $O(\log^{3/2} n)$.
- We give a new construction with evaluation time $O(\log^{1/2} n)$ inspired by a pseudorandom generator for combinatorial rectangles by Lu '02.
- Siegel's lower bound would imply size $\Omega(2^{\sqrt{\log n}})$ bits for this evaluation time!

# Comparison of Results

| | Size (bits) | Evaluation Time |
|---|---|---|
| Lower Bound | $\Omega(\log n)$ | $\Omega(\log n/\log\log n)$ |
| Full Independence | $\Omega(\lvert U\rvert \log n)$ | $O(1)$ |
| $n^{\varepsilon}$-wise independence | $O(n^{\varepsilon})$ | $O(1)$ |
| $\log n/\log\log n$ -wise independence | $O(\log^2 n/\log\log n)$ | $O(\log n/\log\log n)$ |
| Construction 1 | $O(\log n \log\log n)$ | $O(\log n \log\log n)$ |
| Construction 2 | $O(\log^{3/2} n)$ | $O(\log^{1/2} n)$ |

# Open Questions




- Can we improve to match optimal bounds?
- Can we show these hash functions perform well for use in cuckoo hashing, linear probing, etc.
- A general time-space lower bound.

# Thanks!