Construction of Rigid Matrices from PCPs

Amey Bhangale
(UC Riverside, USA)

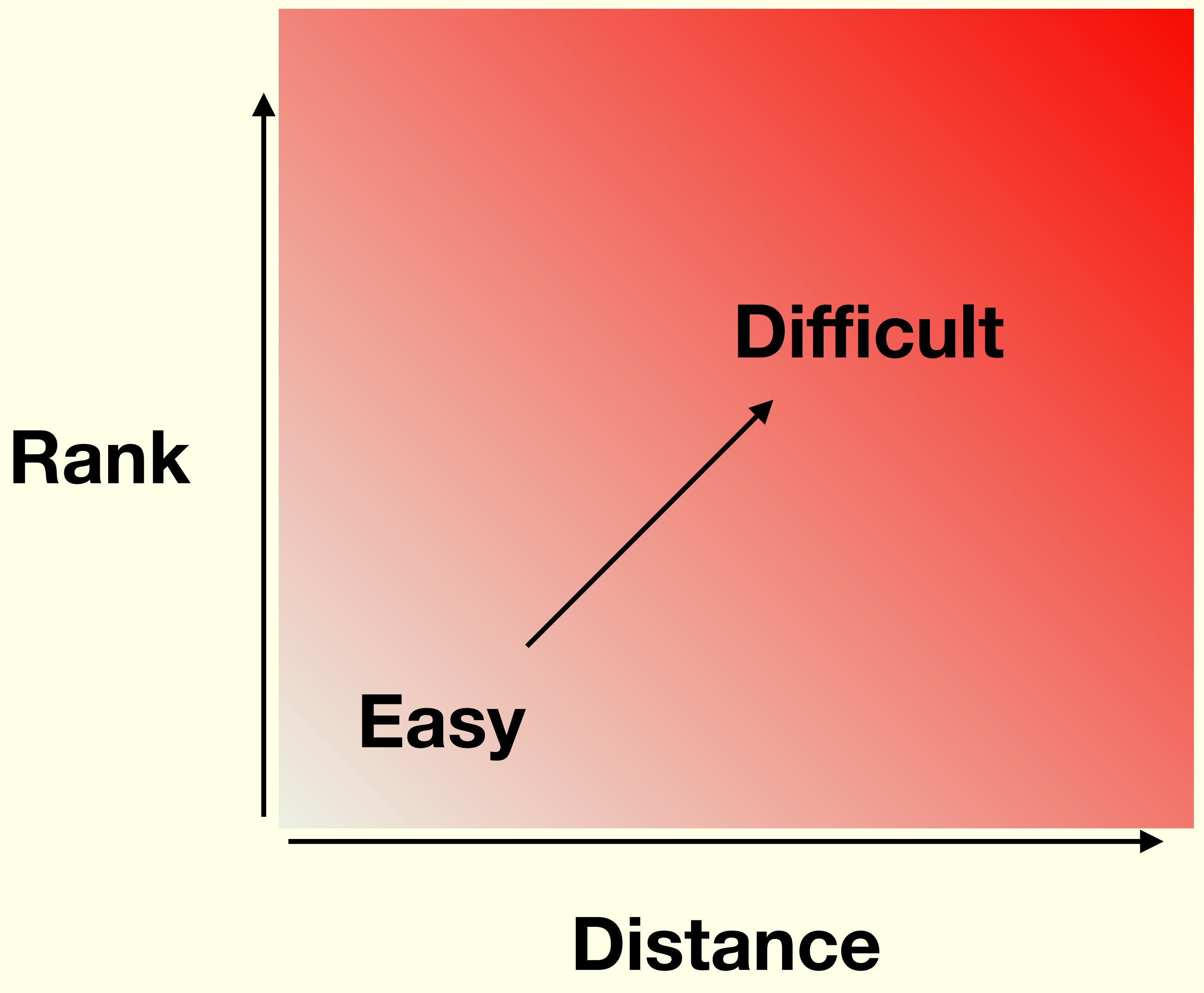
Workshop on Matrix Rigidity
FSTTCS, Dec 2020

What are rigid matrices?

- A matrix $A \in \mathbb{F}_2^{N \times N}$ is rigid if it is far from low rank matrices.

- Formally, a matrix A is (ρ, Δ)-rigid if

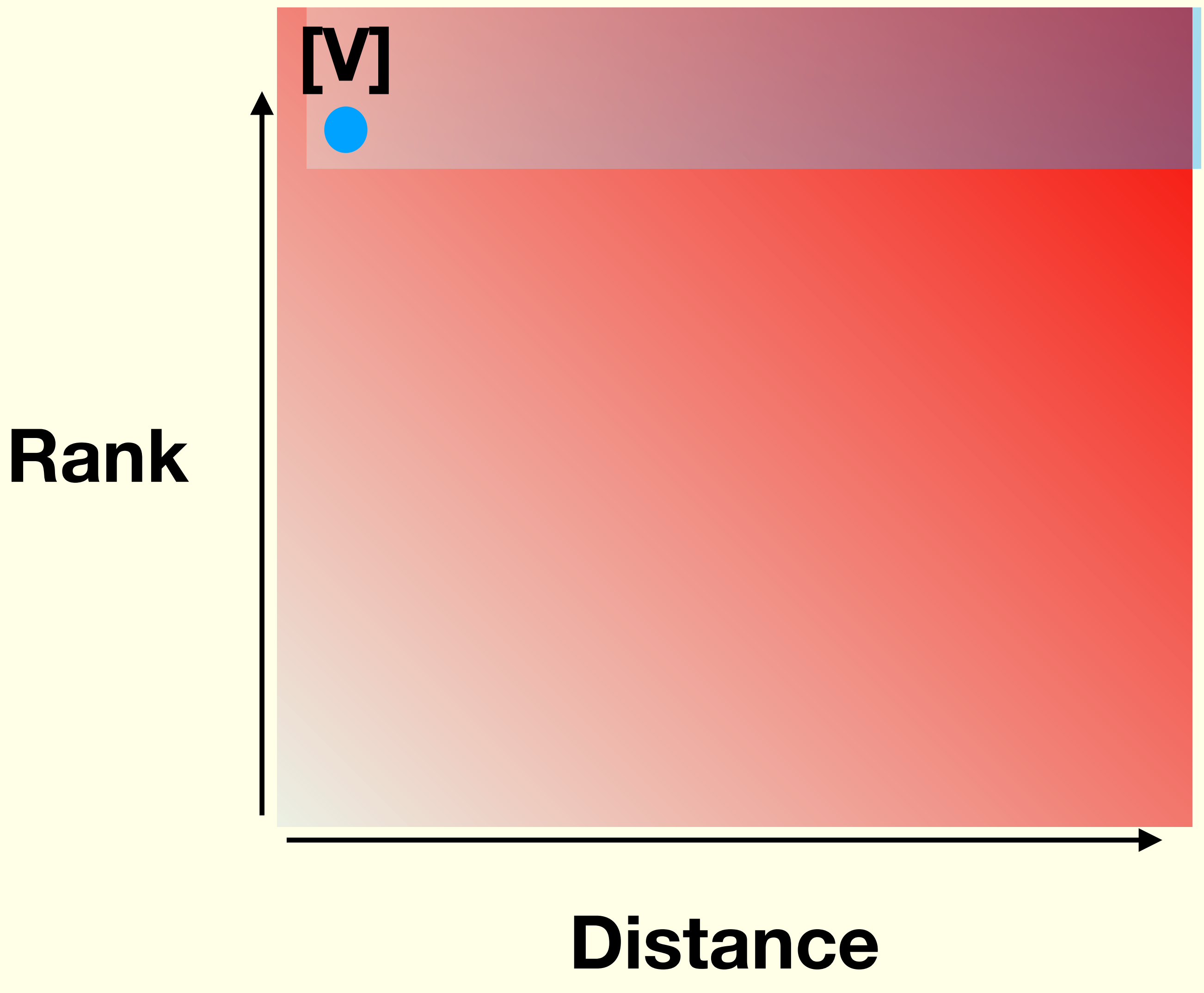
$$\min_{B: rank(B) = \rho} dist(A, B) \geq \Delta$$



Applications

- Circuit lower bound

[Valiant 77]: For any $\epsilon > 0$, if A is $\left(\frac{N}{\log\log N}, N^{1+\epsilon}\right)$-rigid, then $x \to Ax$ can't be computed by circuits of size $O(N)$ and depth $O(\log N)$



[FGHK 16] Current best known (explicit) circuit lower bound: 3.01 N

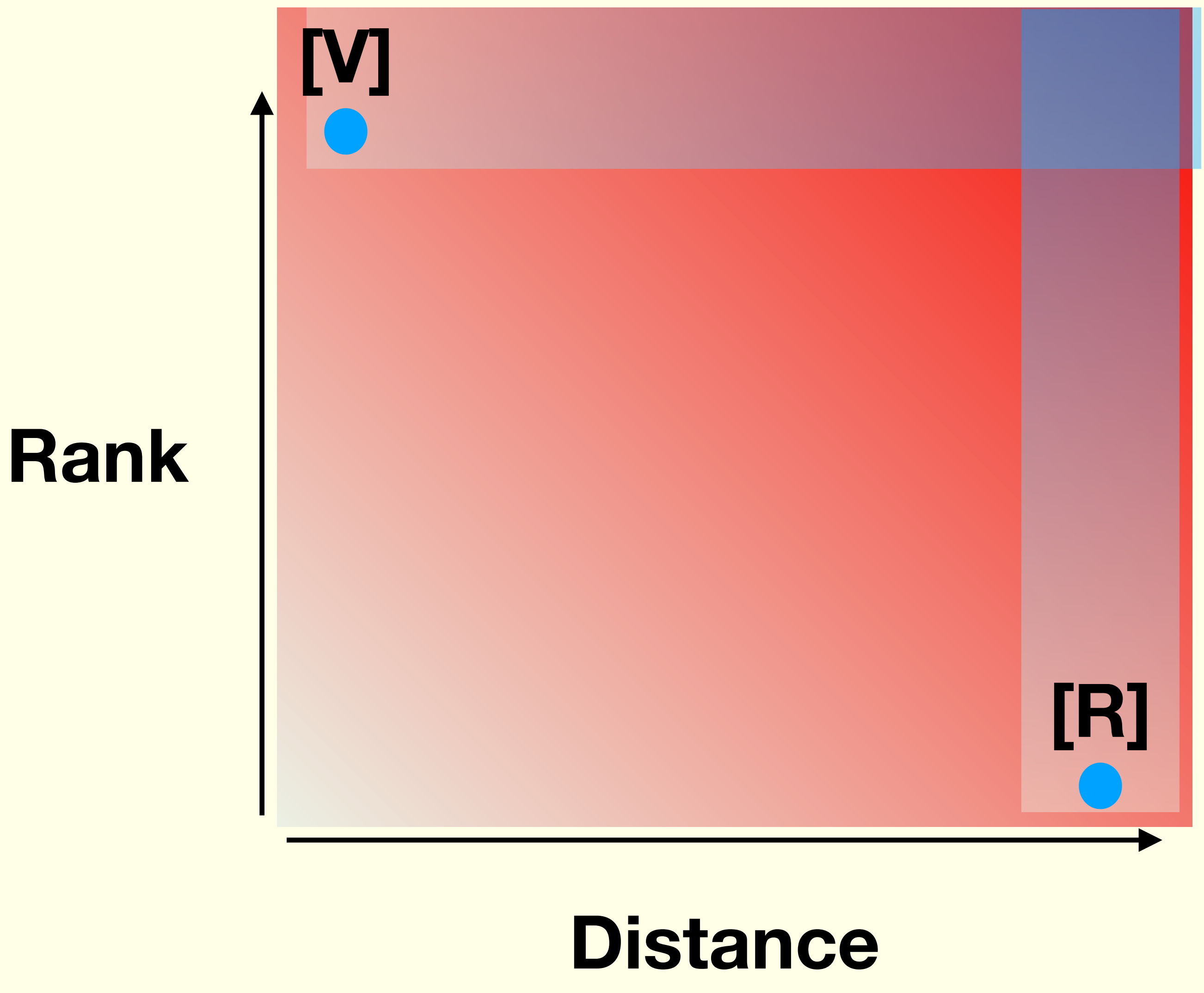
Applications

- Communication complexity

[Razborov 89] Let $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}$ be a function PH^{cc}, then for every $\epsilon > 0$, the communication matrix M_f is **not** (ρ, Δ)-rigid where

$$\rho = 2^{poly\left(\frac{\log n}{\epsilon}\right)}, \Delta = \epsilon \cdot 4^n$$

$$M_f(x, y) = f(x, y)$$



Applications

- Approximate probabilistic $\mathbb{F}_2-$degree

If ϵ-approximate probabilistic degree of $f : \{0,1\}^{2n} \to \{0,1\}$ is at most ρ then M_f is **not** $(n^{O(\rho)}, \epsilon 4^n)$-rigid.

[Razborov,Smolensky 89] Approximating Majority needs probabilistic $\mathbb{F}_2-$degree at least $\sqrt{n}$.

Previous constructions

		Rank ρ	Distance Δ	Time
	Random Matrix	$O(N)$	$0.01N^2$	DTIME($2^{O(N^2)}$)
Cauchy Matrices DFT Matrix	**[Friedman, SSS 90's]**	Any	$\Omega\left(\frac{N^2}{\rho}\log\left(\frac{N}{\rho}\right)\right)$	DTIME($N^{O(1)}$) (Untouched minor argument)
	[Goldreich-Tal 15]	$\geq \sqrt{N}$	$\Omega\left(\frac{N^3}{\rho^2 \log N}\right)$	DTIME($2^{O(N)}$) (Random Toeplitz matrices ($\mathbb{F}_2$))

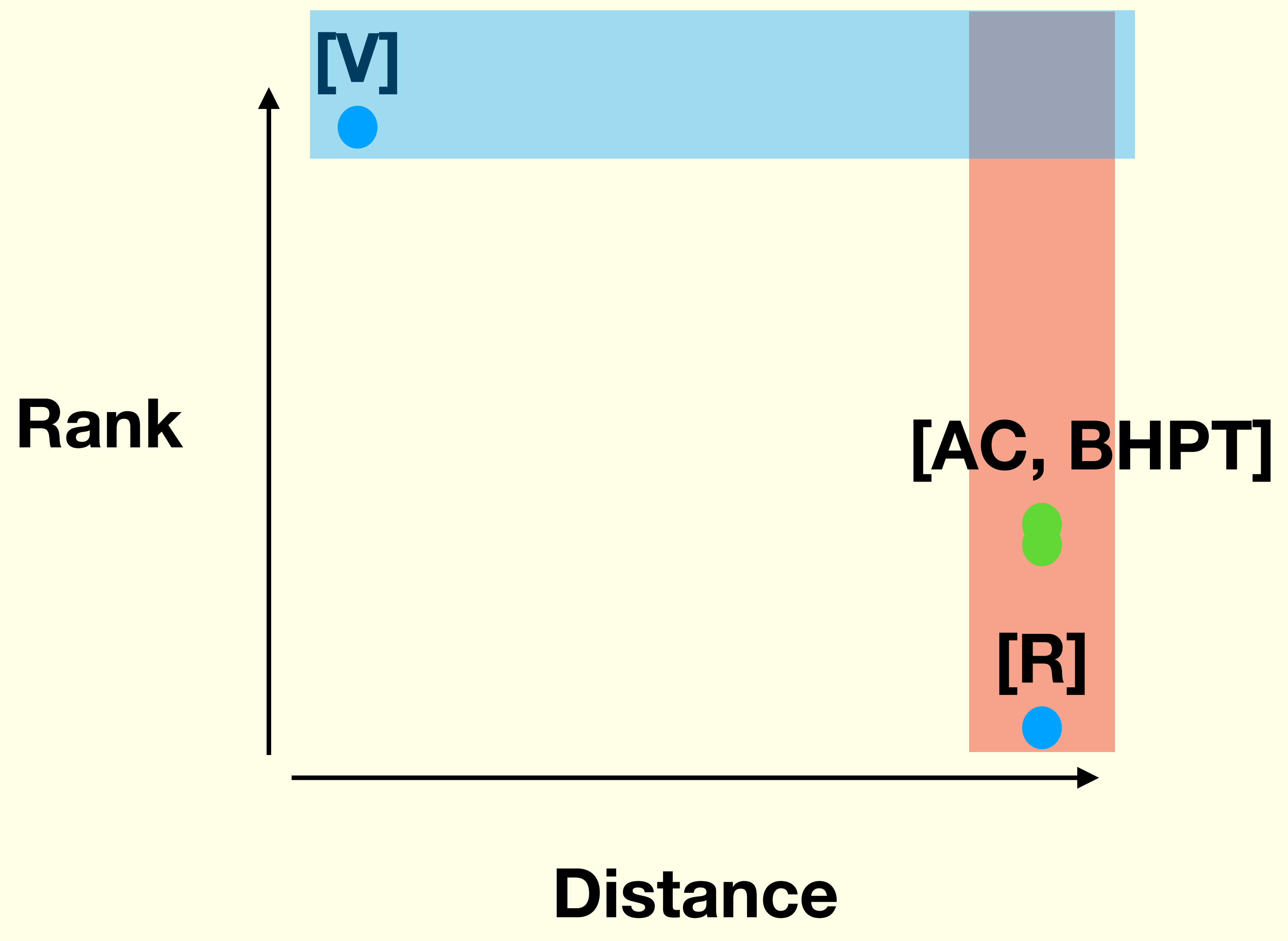
$$\begin{bmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ i & h & g & f & a \end{bmatrix}.$$

Recent constructions

	Rank ρ	Distance Δ	Time
[Alman-Chen 19]	$2^{O\left((\log N)^{1/4-\epsilon}\right)}$	$0.01 N^2$	NTIME($N^{O(1)}$) (PCPs !!)
[B., Harsha, Paradise, Tal 20]	$2^{O\left(\frac{\log N}{\log \log N}\right)}$	$0.01 N^2$	NTIME($N^{O(1)}$) (PCPs !!)

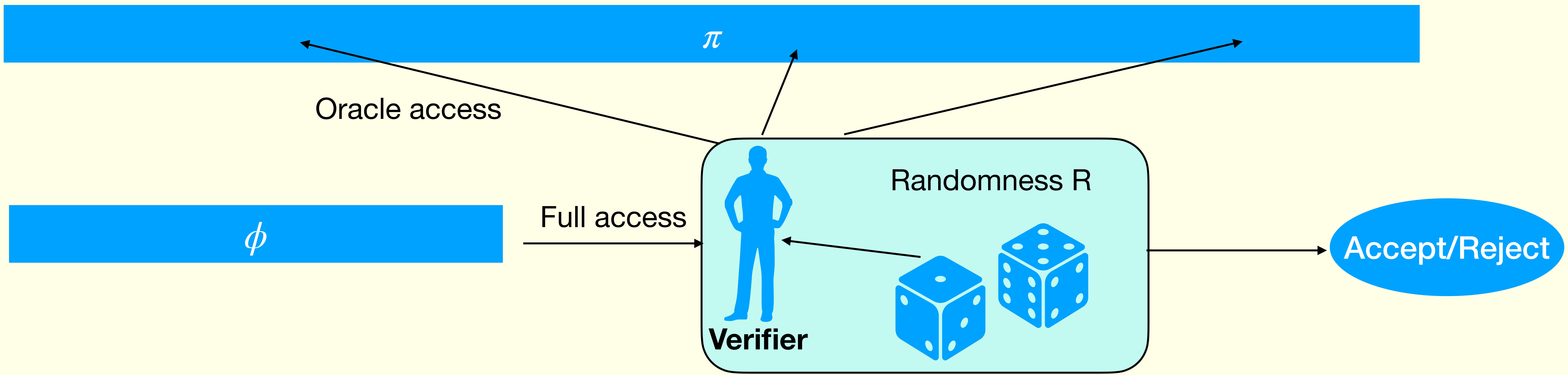
New implications

- [AC19] TIME$[2^{\log n^{\omega(1)}}]^{NP} \not\subseteq PH^{cc}$
- [AC 19] $E^{NP} \not\subseteq$ (restricted) $AC^0[p] \circ LTF \circ AC^0[p] \circ LTF$
- [BHPT 20, Viola 20] Approximate probabilistic $\mathbb{F}_2$-degree lower bound of $\Omega\left(\frac{n}{\log^2 n}\right)$ for a function in E^{NP}
- [BHPT 20] Simpler (PCPs $\rightarrow$Matrix rigidity)
- [BHPT 20] Tight w.r.t. the PCP parameters!



Probabilistically Checkable Proofs

- Suppose we want to prove a mathematical statement ϕ (Think of ϕ: a given instance of 3SAT is satisfiable)
- PCPs provide a way to verify the claim ϕ, by reading the proof at a few locations.
- The prover needs to write the proof in a specific format.



Probabilistically Checkable Proofs

- Completeness: "Correct claims can always be proven"

If ϕ is true, then there exists $\pi^\star$

$$\mathbf{Pr}_R[V^{\pi^\star}(\phi, R) = 1] \geq c$$

- Soundness: "Incorrect claims cannot be proven"

If ϕ is false, then for all π

$$\mathbf{Pr}_R[V^{\pi}(\phi, R) = 1] \leq s$$

Probabilistically Checkable Proofs

- Other important parameters
  - Size of the proof
  - Number of queries
  - Gap between the completeness (c) and soundness (s)
  - Verifier's running time $V(\phi, R)$
  - Smoothness: all locations from π are equally likely to be queried

Probabilistically Checkable Proofs

- For any language in NTIME($T(n)$), there exist PCPs with the following parameters

- ✓ Size of the proof — $T \cdot \mathtt{poly}(\log T)$ — [BGHSV 1]
- ✓ Number of queries — $q = O(1)$
- ✓ Gap between the completeness and soundness — (1 vs. ϵ) for any $\epsilon > 0$
- ✓ Verifier's running time $V(\phi, R)$ — $\mathtt{poly}(\log T)$ — [BGHSV 2]
- ✓ Smoothness: all locations from π are equally likely to be queried

PCPs to Rigid Matrix (Overview) [Alman-Chen 19]

- L be any unary language in NTIME(2^n)$\setminus$NTIME($2^n/n$). Given $x = 1^n$
- Let π be the proof of "$x \in L$" written in a matrix form
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then we will put L $\in$ NTIME($2^n/n$), a CONTRADICTION!

Overview

- L be any unary language in NTIME(2^n)\NTIME($2^n/n$). Given $x = 1^n$
- Let π be the proof of "$x \in L$" written in a matrix form
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then let A, B be the low rank decomposition

    > Since
    > $|\pi| = 2^n \cdot \texttt{poly}(n), |A| + |B| \ll 2^n/n$

  - (Guess A, B) Simulate verifier on $A\,.\,B$
    - Completeness: Accepted with probability $1 - q\delta$

      > Smoothness and $\pi^\star \approx_\delta A \cdot B$

    - Soundness : Accepted with probability < 0.0001
  - If the overall verification is done in time $< 2^n/n$, then CONTRADICTION!

Overview

- L be any unary language in NTIME(2^n)$\setminus$NTIME($2^n/n$). Given $x = 1^n$
- Let π be the proof of "$x \in L$" written in a matrix form
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then let A, B be the low rank decomposition
  - (Guess A, B) Simulate verifier on $A \,.\, B$
    - Completeness: Accepted with probability $1 - q\delta$
    - Soundness : Accepted with probability < 0.0001
  - If the overall verification is done in time $< 2^n/n$, then CONTRADICTION!

Must fail for infinitely many n

NTIME machine outputting rigid matrices

- On input 1^N, the machine does the following:
  - Let L be the language from the previous slide
  - Set $x = 1^n$ $\quad (N = 2^n)$

    can be done in $\texttt{poly}(N)$ non-deterministic time
  - Guess the "proof" π of the statement "$x \in L$" ($\pi \in \mathbb{F}_2^{\sim 2^{n/2} \times \sim 2^{n/2}}$)

    Recall $|\pi| \approx 2^n \cdot \texttt{poly}(n)$
  - Output the matrix π

Claim: For infinitely many N, the machine outputs a rigid matrix.

$\blacksquare$

Overview

- L be any unary language in NTIME(2^n)$\setminus$NTIME($2^n/n$). Given $x = 1^n$
- Let π be the proof of "$x \in L$" written in a matrix form
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then let A, B be the low rank decomposition
  - (Guess A, B) Simulate verifier on $A \,.\, B$ (needs to be done in $< 2^n/n$ time)
    - Completeness: Accepted with probability $1 - q\delta$
    - Soundness : Accepted with probability < 0.0001

(Calculate the acceptance prob. in $< 2^n/n$ time)

  - If the overall verification is done in time $< 2^n/n$, then CONTRADICTION!

Overview

- L be any unary language in NTIME(2^n)$\setminus$NTIME($2^n/n$). Given $x = 1^n$
- Let π be the proof of "$x \in L$" written in a matrix form
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then let A, B be the low rank decomposition
  - (Guess A, B) Simulate verifier on $A \, . \, B$ (needs to be done in $< 2^n/n$ time)
    - Completeness: Accepted with probability $1 - q\delta$
    - Soundness : Accepted with probability < 0.0001
  - If the overall verification is done in time $< 2^n/n$, then CONTRADICTION!

Simpler simulation using "rectangularity"

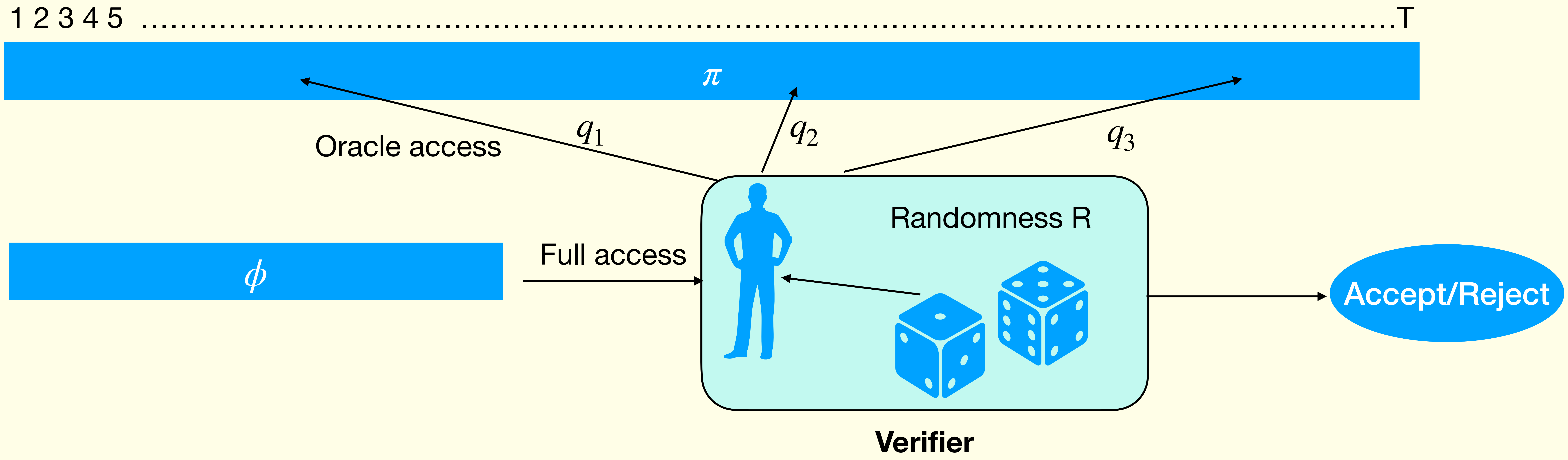
(Calculate the acceptance prob. in $< 2^n/n$ time)

[AC 19] Boils down to fast counting #1s in a product of low rank matrices

Rest of the talk

- Introduce rectangular PCPs
- Convince that the simulation can be done in $< 2^n/n$ non-deterministic time
- How the fast counting is used in this process

PCPs

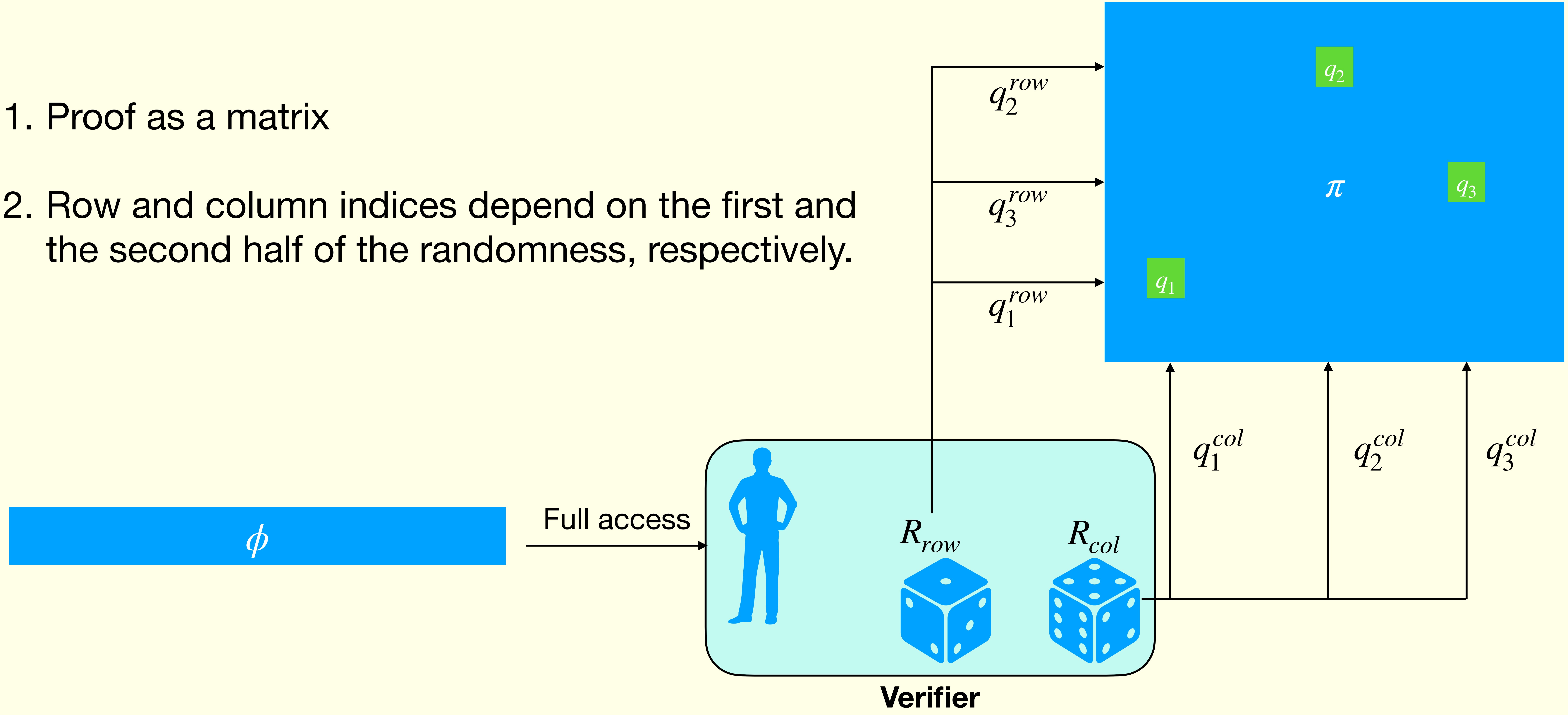


Each query location depends on the full randomness R

Rectangular PCPs

1. Proof as a matrix

2. Row and column indices depend on the first and the second half of the randomness, respectively.



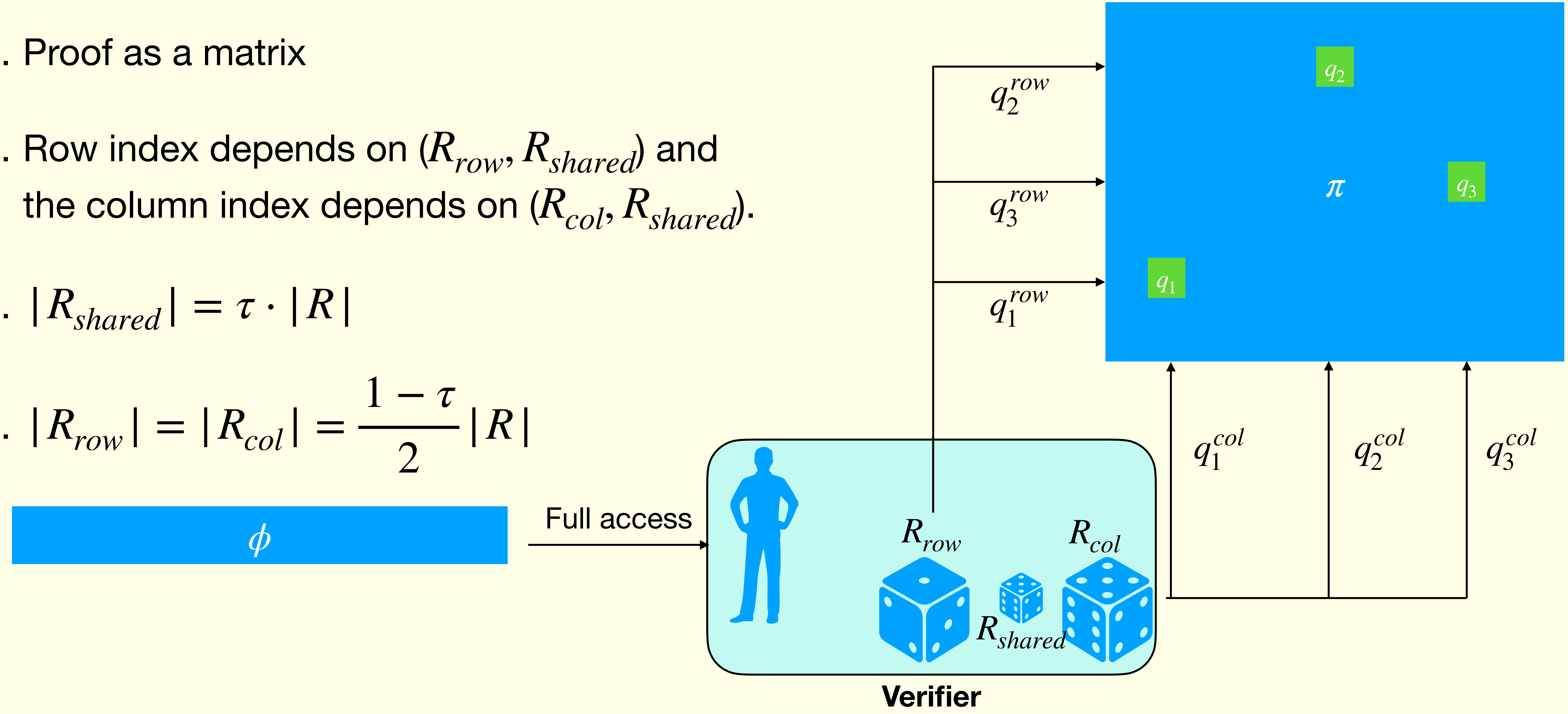
$\tau-$almost-rectangular PCPs

1. Proof as a matrix

2. Row index depends on (R_{row}, R_{shared}) and the column index depends on (R_{col}, R_{shared}).

3. $|R_{shared}| = \tau \cdot |R|$

4. $|R_{row}| = |R_{col}| = \dfrac{1-\tau}{2}|R|$



Main Theorem

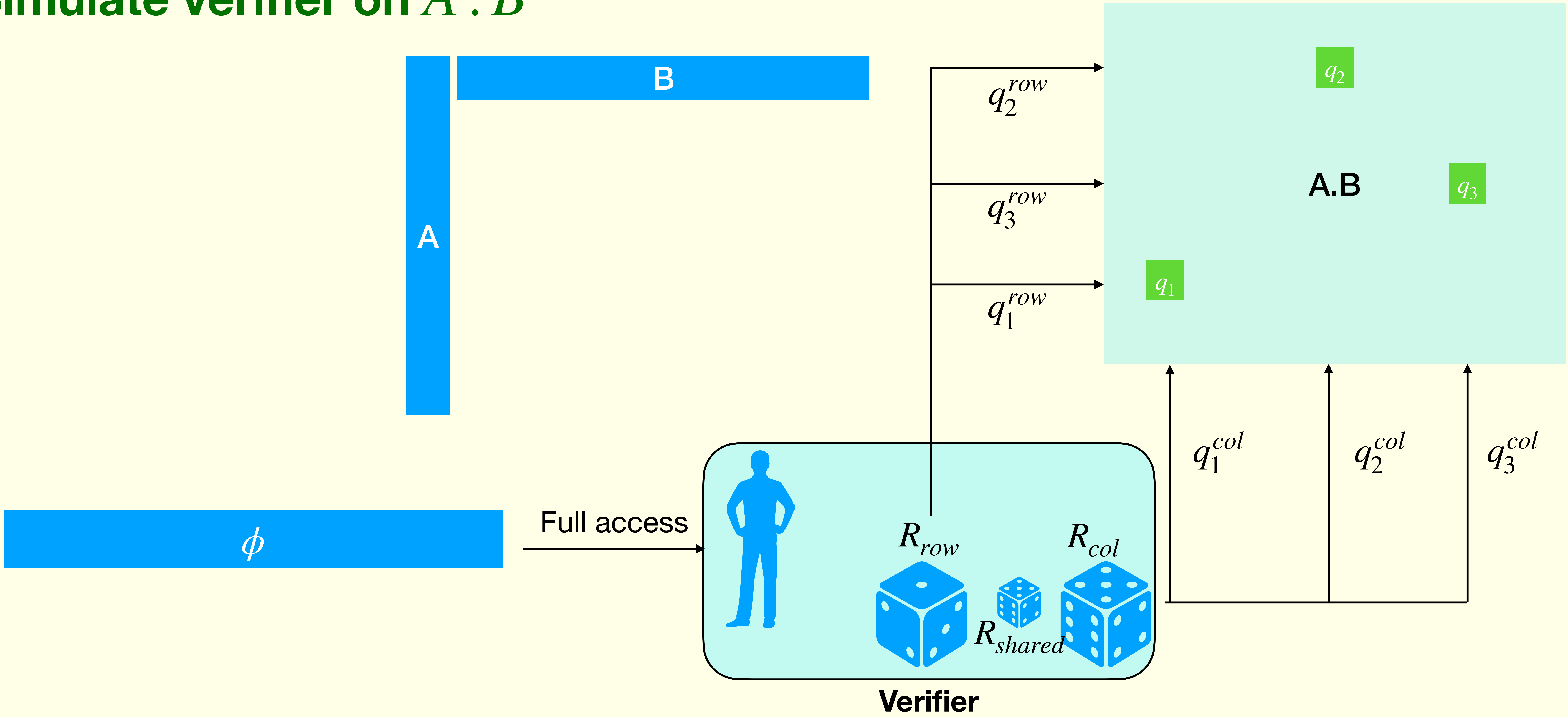
- [BHPT 20] Fix any $\epsilon, \tau > 0$. For every language $L \in$ NTIME(2^n), there exists a rectangular PCP with the following parameters:
  - Completeness 1 and soundness ϵ
  - Query complexity $O(1)$
  - Proof size $2^n \cdot \mathtt{poly}(n)$ (Randomness complexity = $n + O(\log n)$)
  - Verifier's run-time $2^{\epsilon n}$
  - Smooth and $\tau-$almost rectangular

Almost-rectangular PCP $\rightarrow$ Rigid Matrices

- L be any unary language in NTIME(2^n)$\setminus$NTIME($2^n/n$). Given $x = 1^n$
- Let π be the (almost-rectangular) proof.
- π cannot be $\delta-$approximated (hamming distance) by a low rank matrix
  - If it were then let A, B be the low rank decomposition of π
  - (Guess A, B) **Simulate verifier on $A \,.\, B$**
    - Completeness: Accepted with probability $1 - q\delta$
    - Soundness : Accepted with probability < 0.0001
  - If the overall verification is done in time $< 2^n/n$, then CONTRADICTION!

Simulation

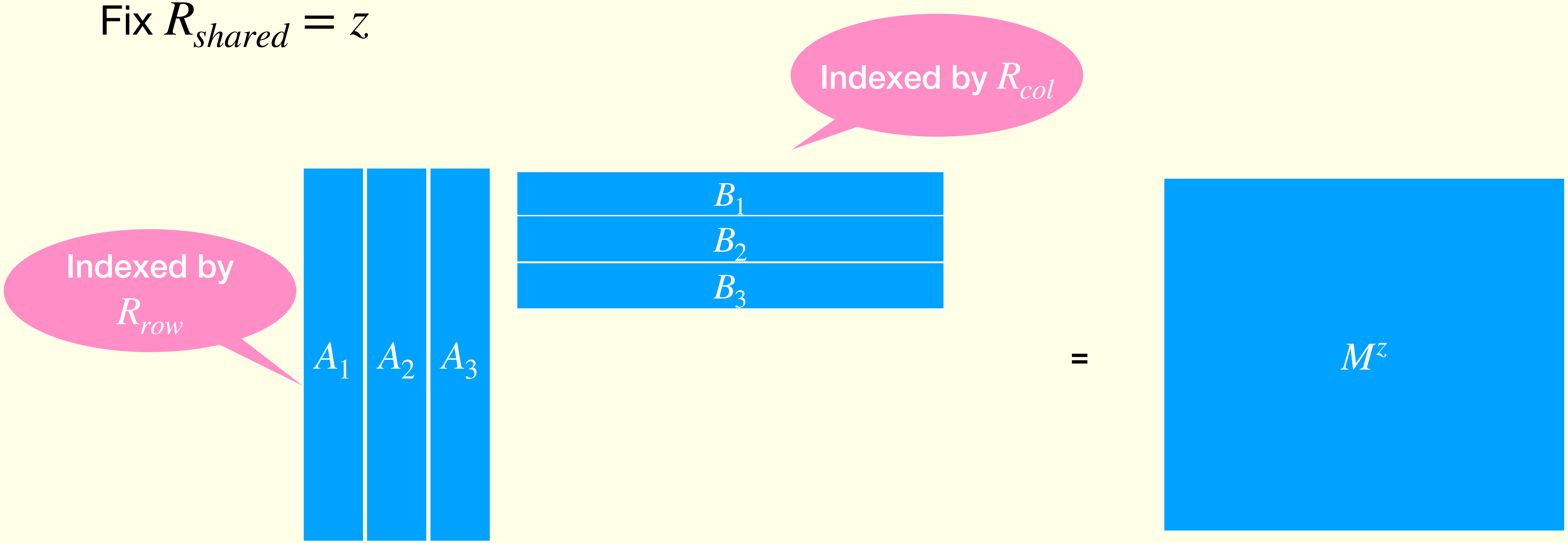
Simulate verifier on $A\,.\,B$



For simplicity, assume that the verifier is querying 3 bits and accepting iff the parity of the three bits is 1.

Counting #1s in a prod. of low rank matrices

Fix $R_{shared} = z$



A_k:= i-th row of A_k = $q_k^{row}(i, z)$-th row of A

B_k:= j-th col of B_k= $q_k^{col}(j, z)$-th col of B

$M^z(i, j)$ = parity of the 3 bits queried by the verifier on randomness (i, R_{shared}, j)

Simulation

Simulate verifier on $A . B$

- For every $z \in \{0,1\}^{|R_{shared}|}$
  - Calculate the fraction of 1s in M^z. Let the fraction be p_z
- Acceptance probability on the "proof" $A \cdot B$ is

$$\mathbf{E}_z[p_z]$$

Total running time of the simulation: $2^{\tau n} \cdot \left(2^{n/2-\tau/2} \cdot \rho \cdot 6 \cdot 2^{\epsilon n} + \text{calculate } p_z\right)$

Set of R_{shared}

Setting up matrices A_1, A_2, A_3 and B_1, B_2, B_3

The maps $(R_{row}, R_{shared}) \to q_k^{row}$ and $(R_{col}, R_{shared}) \to q_k^{col}$

Fast counting

Total running time of the simulation: $2^{\tau n} \cdot \left(2^{n/2-\tau/2} \cdot \rho \cdot 6 \cdot 2^{\epsilon n} + \text{calculate } p_z\right)$

Set of R_{shared}

Setting up matrices A_1, A_2, A_3 and B_1, B_2, B_3

The maps $(R_{row}, R_{shared}) \to q_k^{row}$ and $(R_{col}, R_{shared}) \to q_k^{col}$

Calculate p_z

- Given two matrices $X \in \mathbb{F}_2^{N\times r}$ and $Y \in \mathbb{F}_2^{r\times N}$, compute the number of 1s in $X \cdot Y$
    - [Chan-Williams 16] Can be done in time roughly $N^{2-\frac{1}{\log r}}$ (provided $r = N^{o(1)}$)

Fast counting

Total running time of the simulation: $2^{\tau n} \cdot \left(2^{n/2 - \tau/2} \cdot \rho \cdot 6 \cdot 2^{\epsilon n} + 2^{(1-\tau)n - \frac{n}{\log \rho}} \right)$

Set of R_{shared}

Setting up matrices A_1, A_2, A_3 **and** B_1, B_2, B_3

The maps $(R_{row}, R_{shared}) \to q_k^{row}$ **and** $(R_{col}, R_{shared}) \to q_k^{col}$

Calculate p_z

- Given two matrices $X \in \mathbb{F}_2^{N \times r}$ and $Y \in \mathbb{F}_2^{r \times N}$, compute the number of 1s in $X \cdot Y$
  - [Chan-Williams 16] Can be done in time roughly $N^{2 - \frac{1}{\log r}}$ (provided $r = N^{o(1)}$)

crucial saving

Finishing the proof

Total running time of the simulation: $\dfrac{2^n}{2^{\frac{n}{\log \rho}}} = \dfrac{2^n}{n}$ (if $\rho \approx 2^{\frac{n}{\log n}}$)

- Given two matrices $X \in \mathbb{F}_2^{N \times r}$ and $Y \in \mathbb{F}_2^{r \times N}$, calculate the number of 1s in $X \cdot Y$
  - [Chan-Williams 16] Can be done in time roughly $N^{2-\frac{1}{\log r}}$ (provided $r = N^{o(1)}$)

crucial saving

Open questions

- Even faster algorithm for counting #1s in a product of low rank matrices
  - Algorithm for higher ranks ($r = N^{\epsilon}$)
- Other complexity implications of this framework? e.g. Rectangular rigidity?