



TREE PCPs

Tamer Mour

Bocconi

Alon Rosen

Bocconi

Ron Rothblum

Succinct

Motivation

Civilization carries out a
never-ending computation over generations.

Every generation performs a new step
in the computation.

We want to verify that past generations
did not make mistakes.

Efficiently.

In a million years, LWE will be broken.



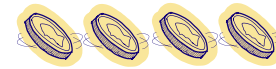
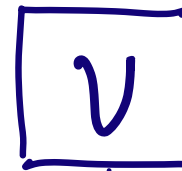
PCPs for NP [BFLS'91, FGLSS'96, AS'98, ALMSS'92, ...]

$$\exists w : C(x, w) = 1 ?$$

Proof π 

- Completeness:
if $x \in L$, $\exists \pi \Pr[V^\pi = 1] = 1$

- Soundness:
if $x \notin L$, $\forall \pi \Pr[V^\pi = 1] < 0.01$



Accept / Reject

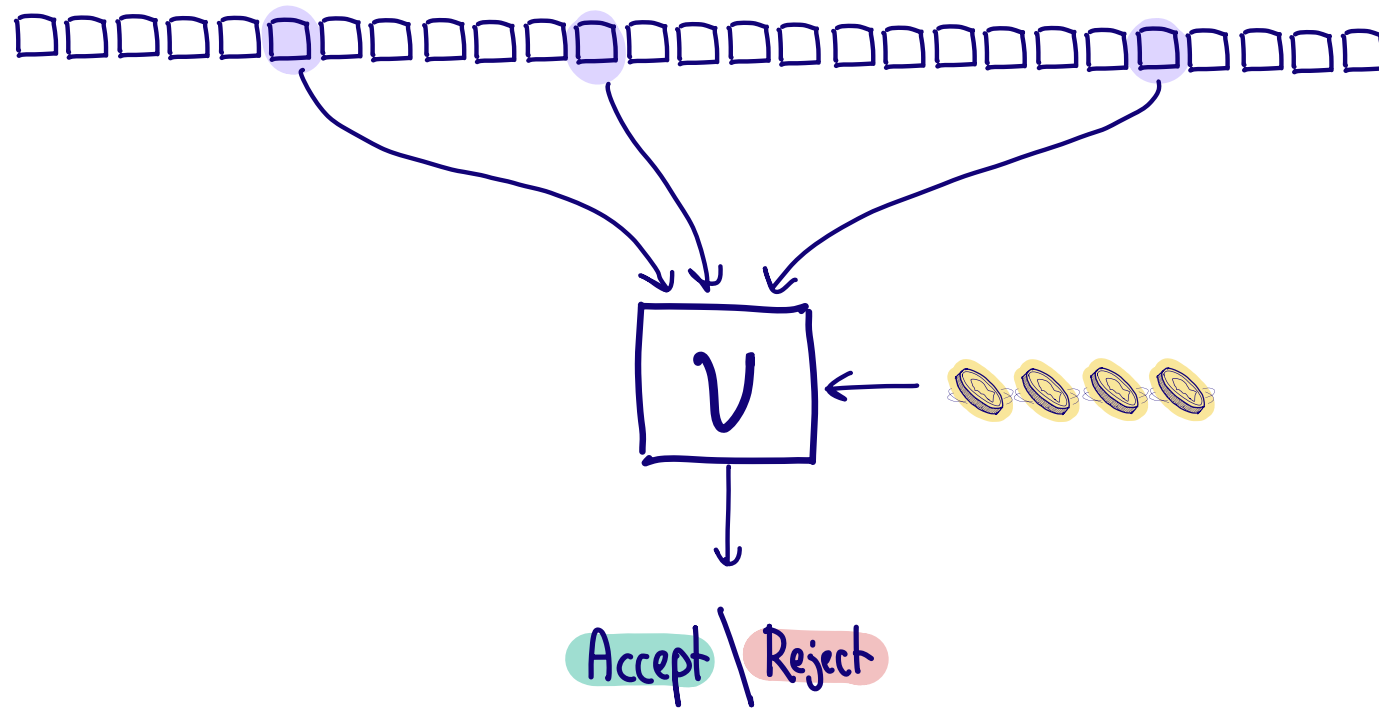
Proof length $|\pi| = \tilde{O}(|C|)$

Query complexity $O(1)$

[Ben Sasson-Sudan '08, Dinur '07]

PCPs for Incremental Computation?

PCP for t steps of the computation



PCPs for Incremental Computation?

PCP for t steps of the computation

π_t 

 "little effort"

PCP for $t+1$ steps

π_{t+1} 

Known PCP designs are inherently not "incremental"

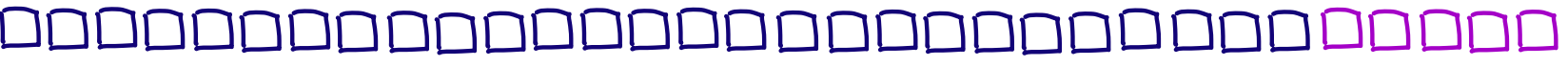
PCPs for Incremental Computation?

PCP for t steps of the computation

π_t 

 "little effort"

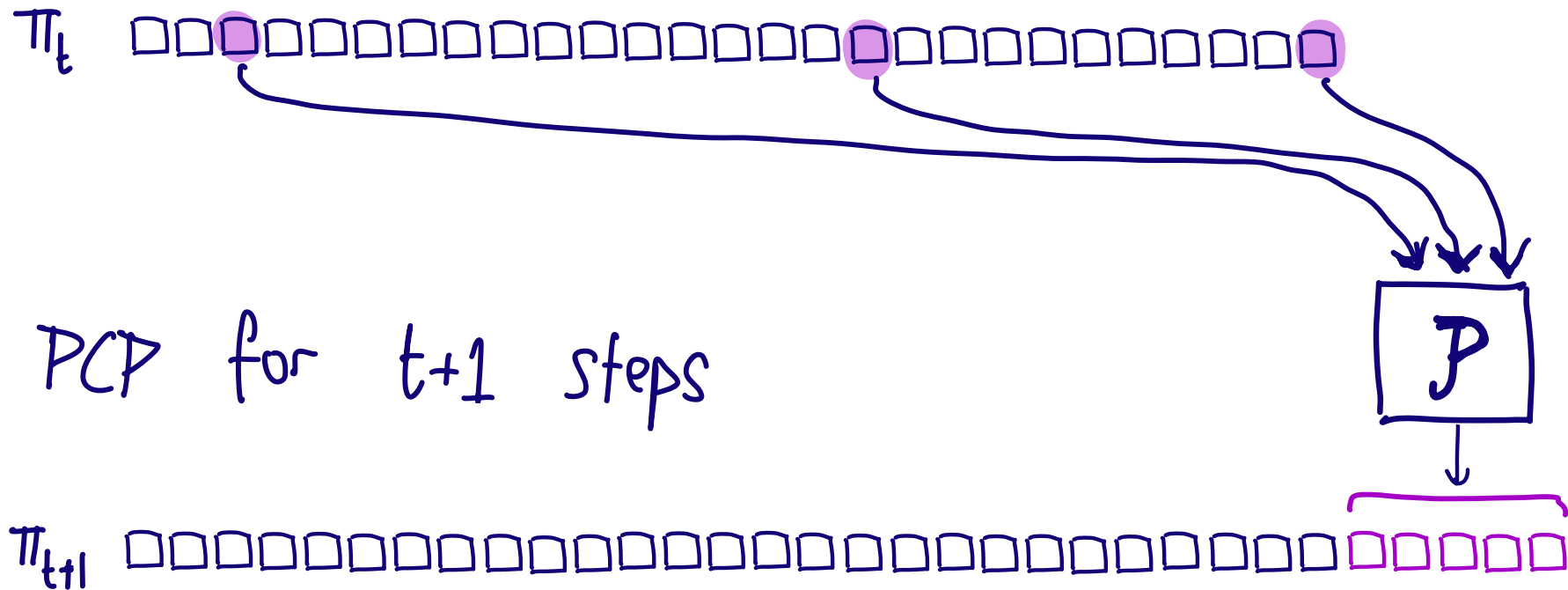
PCP for $t+1$ steps

π_{t+1} 

In Tree PCPs, $\pi_{t+1} = \pi_t + \text{short new string}$

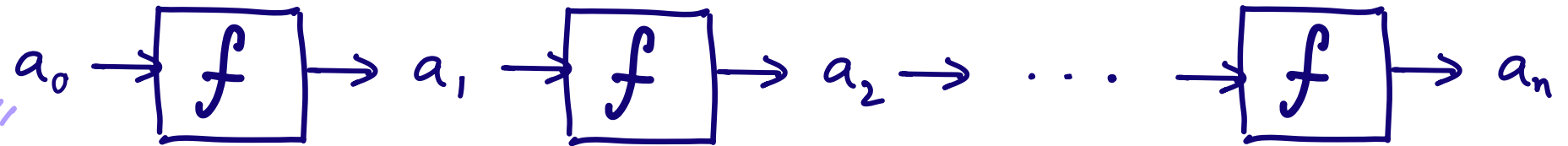
PCPs for Incremental Computation?

PCP for t steps of the computation



In Tree PCPs, $\pi_{t+1} = \pi_t + \text{short new string}$

Incremental Computations



//
(0,0,...,0)
w.l.o.g.

- Space-bounded

$$\forall t, a_t \in \{0,1\}^s$$

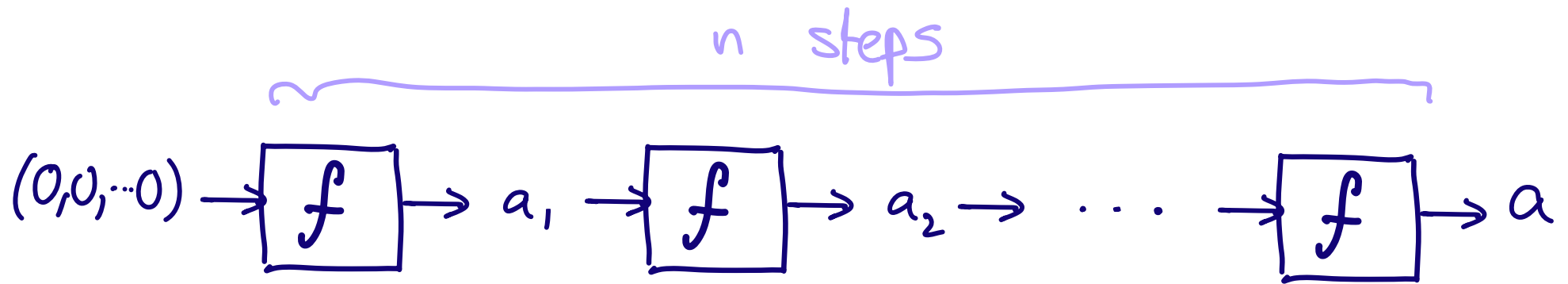
- Non-deterministic

Circuit $C : \{0,1\}^s \times \{0,1\}^s \times \{0,1\}^s \rightarrow \{0,1\}$

$$a_{t-1} \rightarrow \boxed{f} \rightarrow a_t \iff \exists w_t \in \{0,1\}^s : C(a_{t-1}, a_t, w_t) = 1$$

Circuit Reachability

$$(C, a, n) \in \text{CktReach}$$



$$\exists w_1, a_1, w_2, a_2, \dots, a_{n-1}, w_n:$$

$$C(a_{t-1}, a_t, w_t) = 1 \quad \text{for } t=1, \dots, n$$

$$(\text{where } a_0 = 0^s, a_n = a)$$

Tree PCPs: PCPs for CrktReach

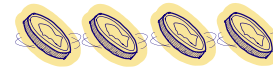
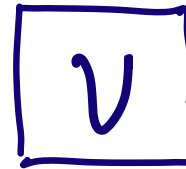
$$(C, a, n) \stackrel{?}{\in} \text{CrktReach}$$

Proof π 

- Completeness:
if $x \in L$, $\exists \pi \Pr[V^\pi(C, a, n) = 1] = 1$

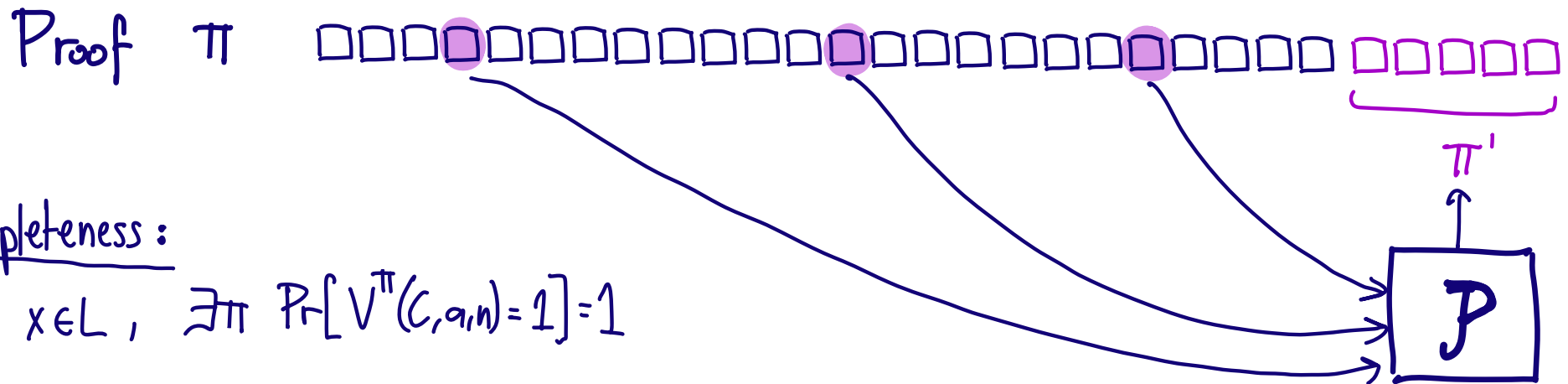
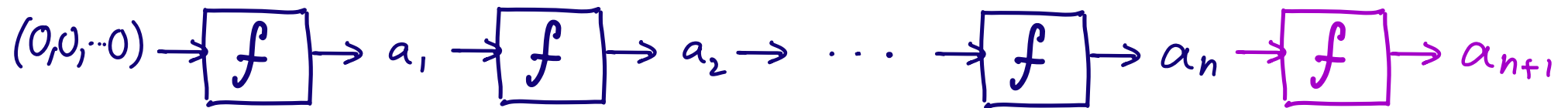
- Soundness:
if $x \notin L$, $\forall \pi \Pr[V^\pi(C, a, n) = 1] < \epsilon$

- "Incrementality": a PCP for $(C, a_{n+1}, n+1)$ can be obtained by appending a short, efficiently computable, string π' to π .



Accept / Reject

Tree PCPs: PCPs for CrktReach



- Completeness:

if $x \in L$, $\exists \pi \Pr[V^\pi(C, a_n) = 1] = 1$

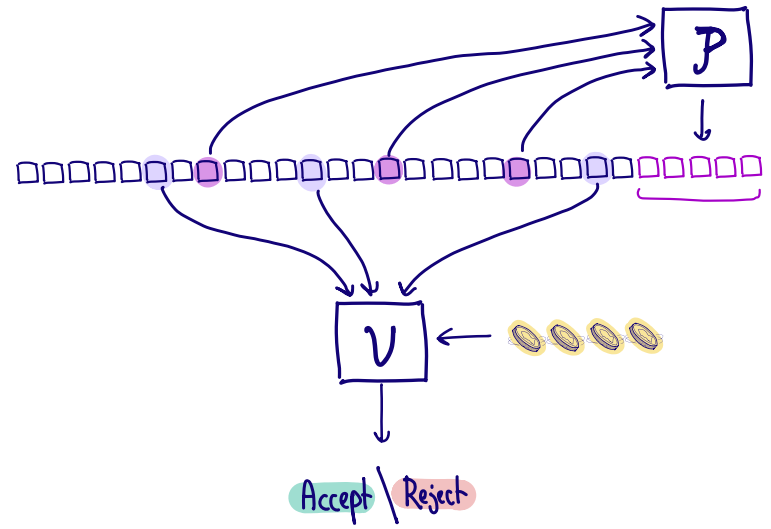
- Soundness:

if $x \notin L$, $\forall \pi \Pr[V^\pi(C, a_n) = 1] < \epsilon$

- "Incrementality": a PCP for $(C, a_{n+1}, n+1)$ can be obtained by appending a short, efficiently computable, string π' to π .

Main Theorem

There exists a Tree PCP



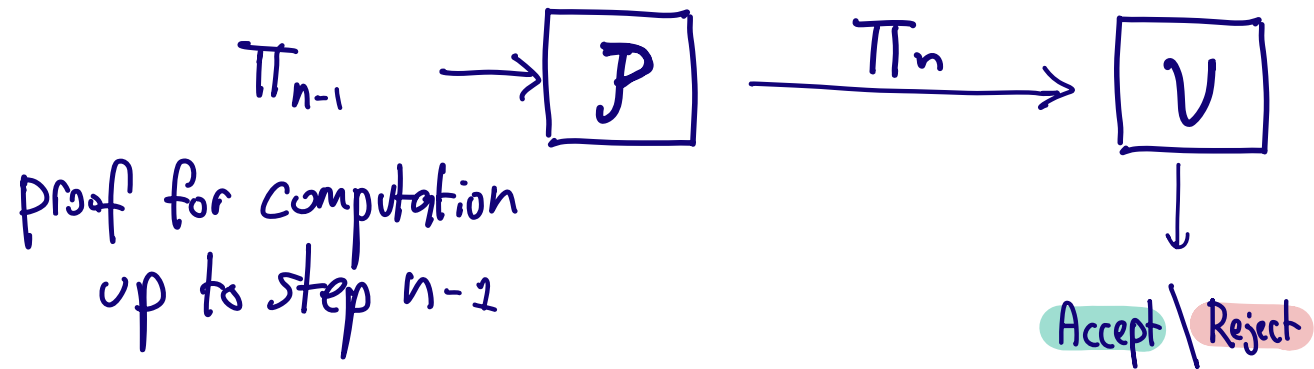
- Query complexity $\tilde{O}(n^\epsilon) \cdot \text{poly}(s)$
- At time n , proofs grows by $\tilde{O}(1) \cdot \text{poly}(s)$
- Extending the proof takes time $\tilde{O}(n^\epsilon) \cdot \text{poly}(s)$

Related Notions

- Incremental PCPs [Naor-Paneth-Rothblum '19]

Related Notions

- Incrementally Verifiable Computation (IVC) [Valiant '08]



- Cryptographically sound
- $|\pi_n| = \tilde{O}(1) \cdot \text{poly}(s)$

Tree PCPs + Kilian/Micali



"IVC" in the ROM

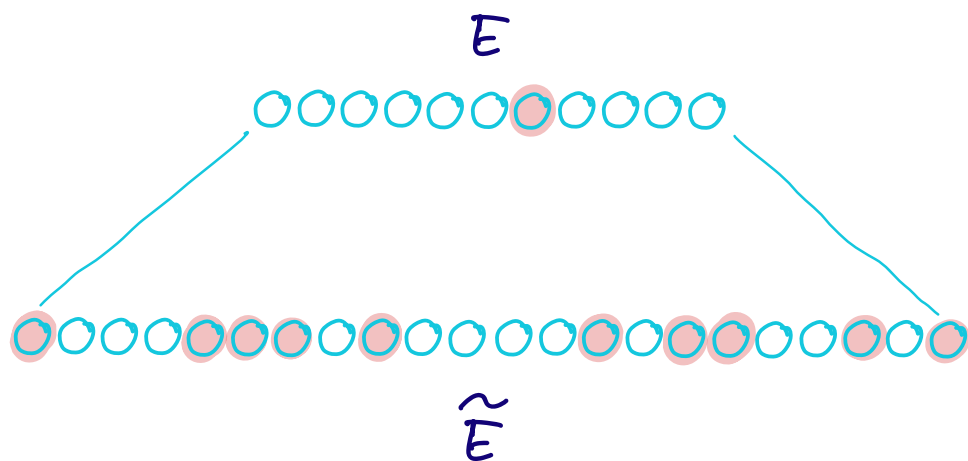
- "IVC" - Prover has access to a large state.
- Relaxation seems necessary [Hall-Andersen-Nielsen'23,
Barbara-Chiesa-Guan'24]

PCP as Redundant Encoding [Babai-Fortnow-Levin-Szegedy '91]

Coding Theory

Redundant encoding of information.

Test if codeword $\tilde{E}$ is all zeros by reading few random locations.



PCPs

Redundant encoding of computation.

Represent statement as $E \equiv 0$.

PCP: encoding $\tilde{E}$.

To Verify:

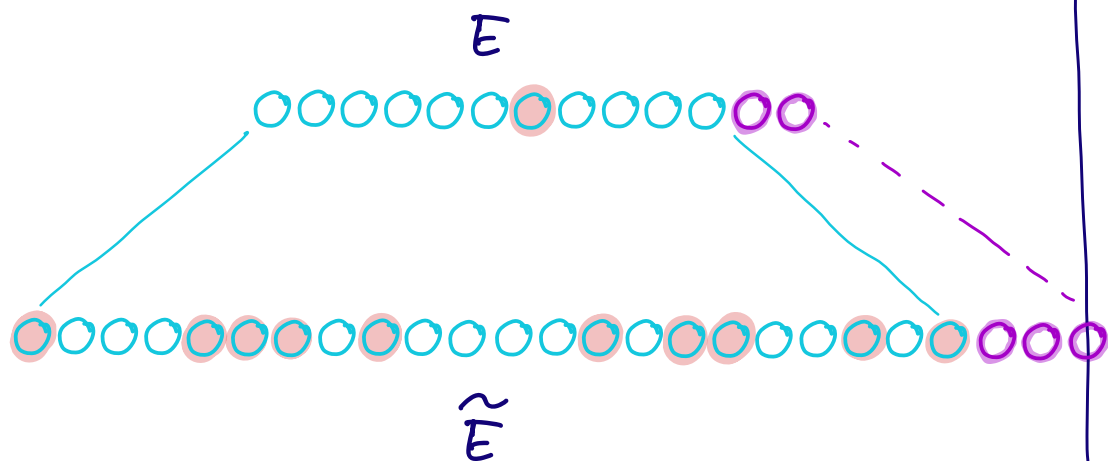
1. Test that PCP is codeword.
 - Local Testability
2. Test that it is all zeros.

PCP as Redundant Encoding [Babai-Fortnow-Levin-Szegedy '91]

Coding Theory

Redundant encoding of information.

Test if codeword $\tilde{E}$ is all zeros by reading few random locations.



PCPs

Redundant encoding of computation.

Represent statement as $E \equiv 0$.

PCP: encoding $\tilde{E}$.

To Verify:

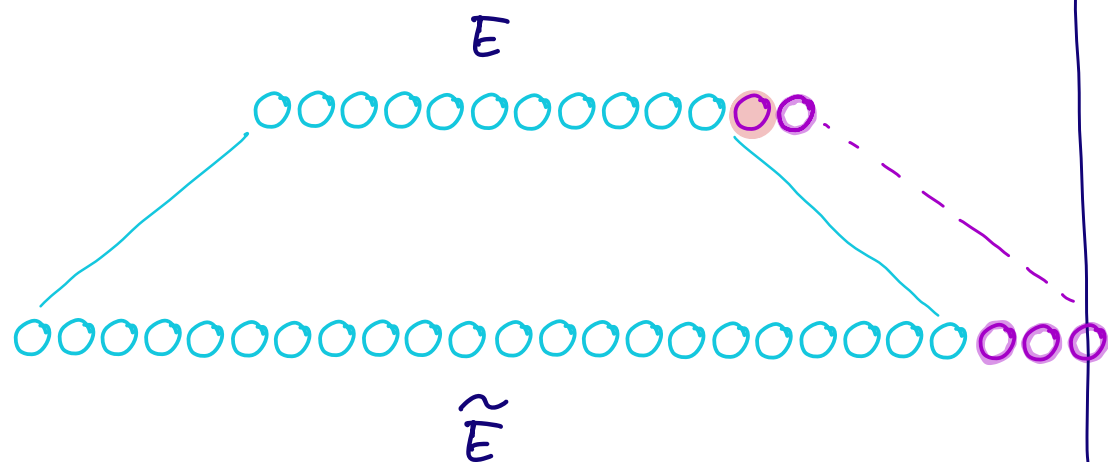
1. Test that PCP is codeword.
 - Local Testability
2. Test that it is all zeros.

PCP as Redundant Encoding [Babai-Fortnow-Levin-Szegedy '91]

Coding Theory

Redundant encoding of information.

Test if codeword $\tilde{E}$ is all zeros by reading few random locations.



PCPs

Redundant encoding of computation.

Represent statement as $E \equiv 0$.

PCP: encoding $\tilde{E}$.

To Verify:

1. Test that PCP is codeword.
 - Local Testability
2. Test that it is all zeros.

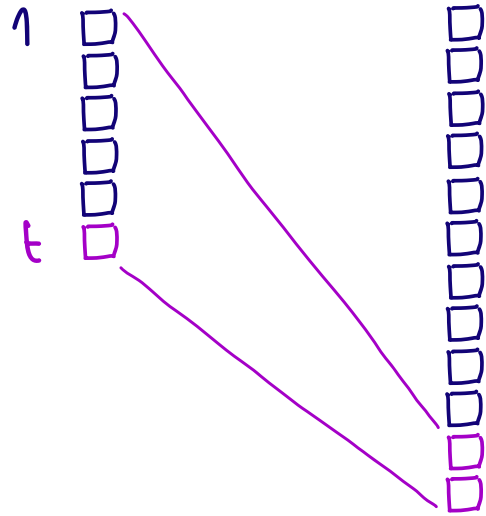
Locally Testable
Block Codes $\Rightarrow$ PCPs

Locally Testable
Tree Codes $\Rightarrow$ Tree PCPs

Tree Codes [Schulman'93]

message A

codeword $\tilde{A}$



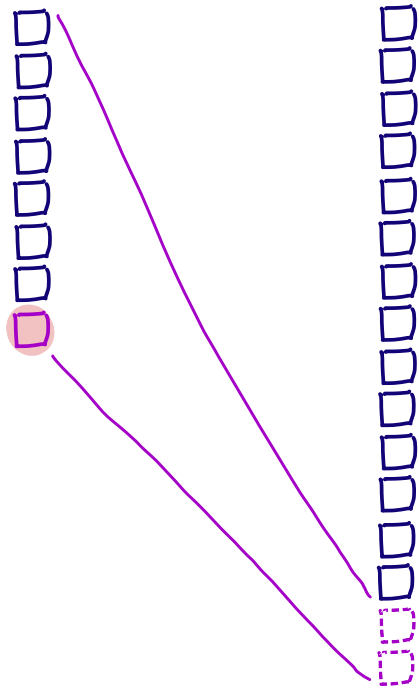
$$TC_t: \{0, 1\}^t \rightarrow \Sigma$$

Online encoding function

Tree Codes [Schulman'93]

message A

codeword $\tilde{A}$



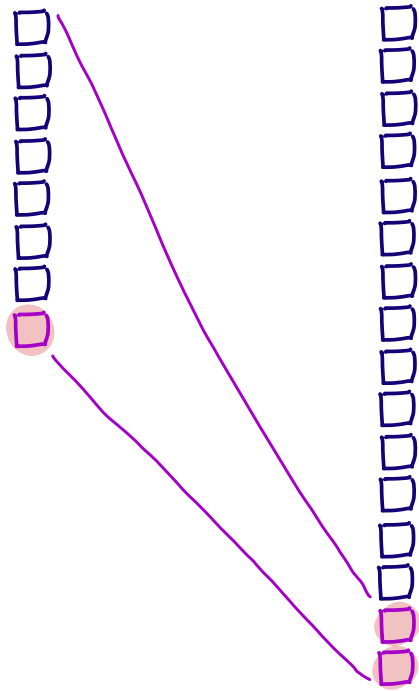
$$TC_t: \{0, 1\}^t \rightarrow \Sigma$$

Online encoding function

Tree Codes [Schulman'93]

message A

codeword $\tilde{A}$



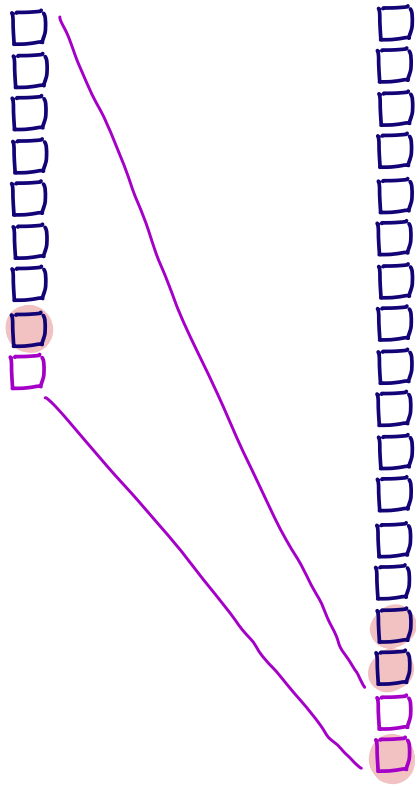
$$TC_t: \{0, 1\}^t \rightarrow \Sigma$$

Online encoding function

Tree Codes [Schulman'93]

message A

codeword $\tilde{A}$



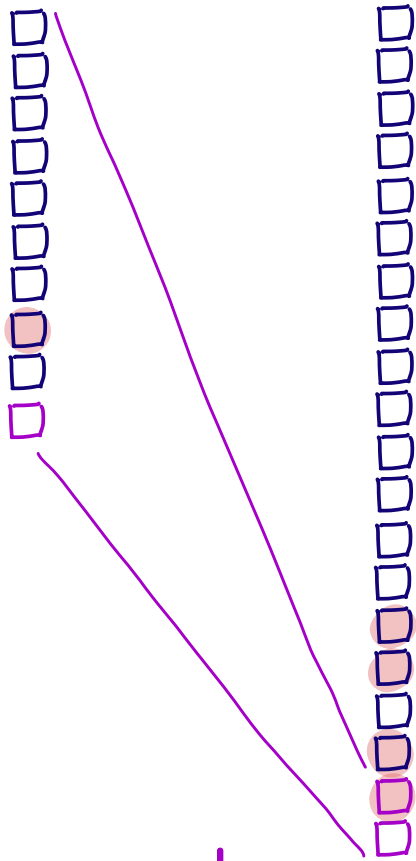
$$TC_t: \{0, 1\}^t \rightarrow \Sigma$$

Online encoding function

Tree Codes [Schulman '93]

message A

codeword $\tilde{A}$



Tree Distance

$\Delta_T(\tilde{A}, \tilde{B}) =$ fraction of disagreements starting first disagreement

minimum tree distance s :

$$\forall \tilde{A}, \tilde{B} \in TC, \Delta_T(\tilde{A}, \tilde{B}) \geq s$$

$$TC_t: \{0, 1\}^t \rightarrow \Sigma$$

Online encoding function

Trees?

Tree Codes

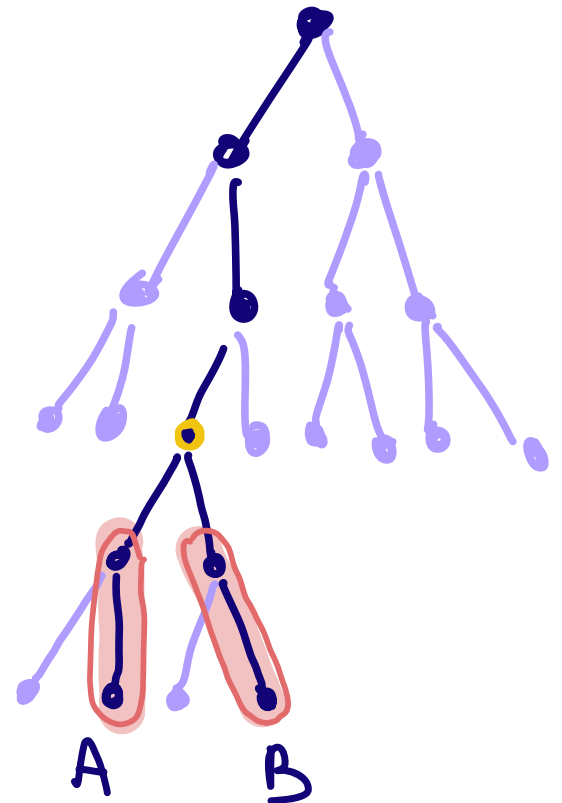
- Labels over edges
- Tree distance between any two paths $A, B \in \{0,1\}^*$.

Tree PCPs

- Paths of non-deterministic computation.

$$A = 0 \rightarrow [f] \rightarrow a_1 \rightarrow [f] \rightarrow a_2 \rightarrow \dots$$

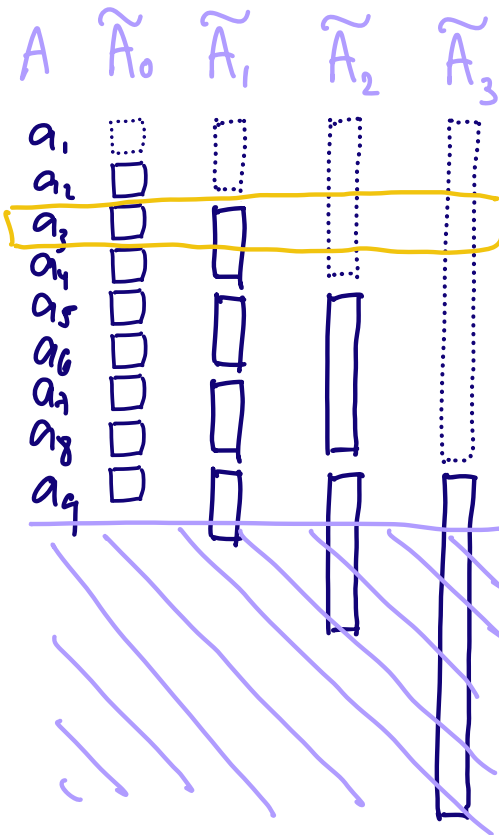
$$B = 0 \rightarrow [f] \rightarrow b_1 \rightarrow [f] \rightarrow b_2 \rightarrow \dots$$



A Tree Code [Schulman '94]

message A

a_1 ☐
 a_2 ☐
 a_3 ☐
 a_4 ☐
 a_5 ☐
 a_6 ☐
 a_7 ☐
 a_8 ☐
 a_9 ☐

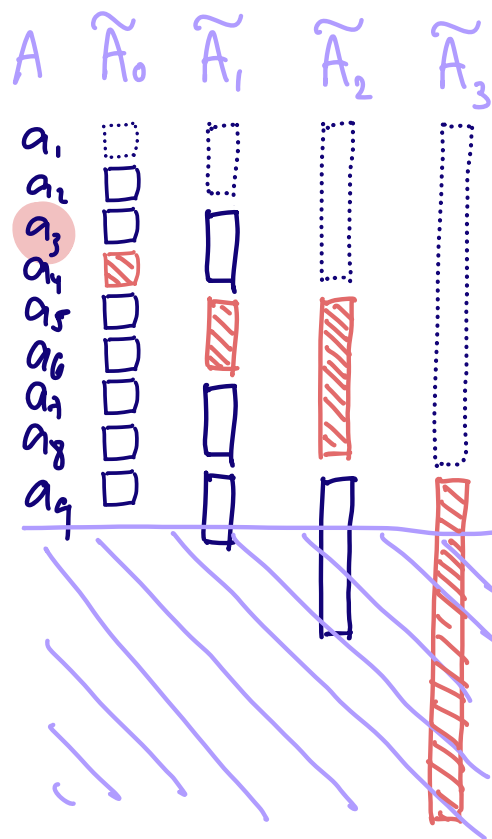


- Error-correction code C (Reed-Solomon).
 - Codeword $\tilde{A} = TC(A)$ has $\lceil \log(n) \rceil + 1$ "threads".
- Thread $\tilde{A}_i$ encodes chunks of size 2^i .
- Start encoding chunk after it ends.
- Split the block codewords to symbols.

A Tree Code [Schulman '94]

message A

a_1 ☐
 a_2 ☐
 a_3 ☒
 a_4 ☐
 a_5 ☐
 a_6 ☐
 a_7 ☐
 a_8 ☐
 a_9 ☐



- Error-correction code C (Reed-Solomon).
- Codeword $\tilde{A} = TC(A)$ has $\lceil \log(n) \rceil + 1$ "threads".

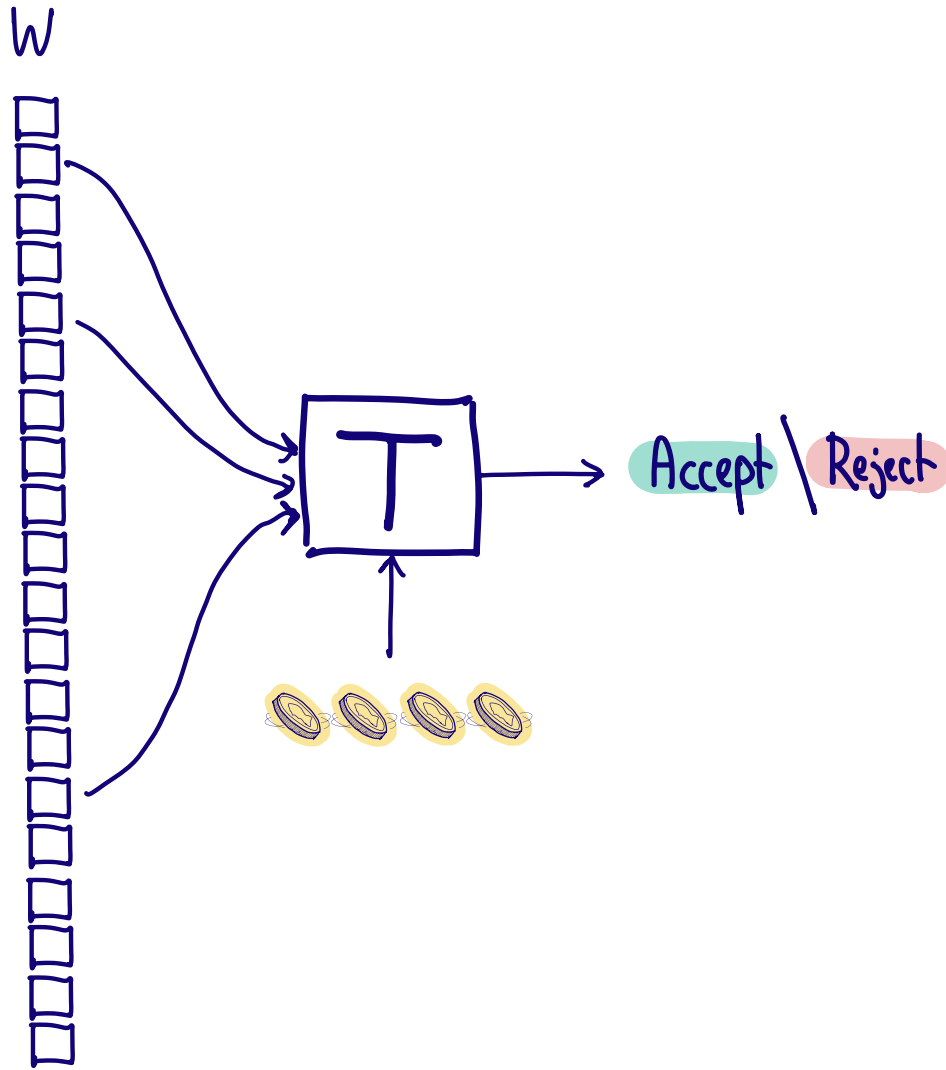
Thread $\tilde{A}_i$ encodes chunks of size 2^i .

Start encoding chunk after it ends.

- Split the block codewords to symbols.
- Tree distance from Hamming distance.
- Rate = $\frac{1}{\log(n)} \times \text{Rate of } C$

Fancier constructions with higher rate [Cohen-Haapler-Schulman '18, ...]

Locally Testable (Tree) Codes



- Completeness:

$$\forall W \in TC, \Pr[T^W = 1] = 1.$$

- Soundness:

$$\Pr[T^W = 0] > \alpha \cdot \Delta_T(W, TC).$$

- Locality:

T reads $q \ll n$ locations.

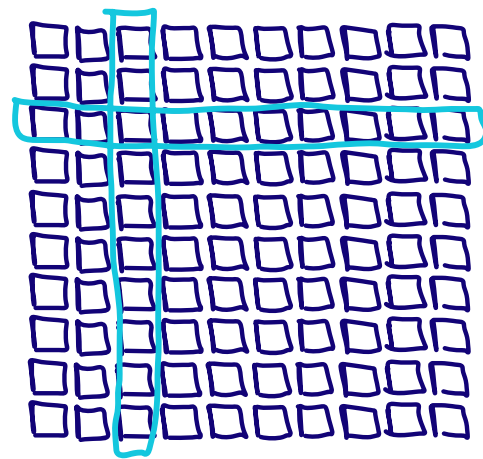
LTCs via Tensoring [Ben-Sasson-Sudan'04, Viderman'15, ...]

Linear Code $C \subset \mathbb{F}^n$

□□□□□□□□□□

$c \in C$

2-fold Tensor $C^2 \subset \mathbb{F}^{n^2}$



$c \in C^2$

iff every row, column $\in C$

Rate $\rho \mapsto \rho^2$

(Relative) Distance $\delta \mapsto \delta^2$

LTCs via Tensoring [Ben-Sasson-Sudan'04, Viderman'15, ...]

Linear Code $C \subset \mathbb{F}^n$

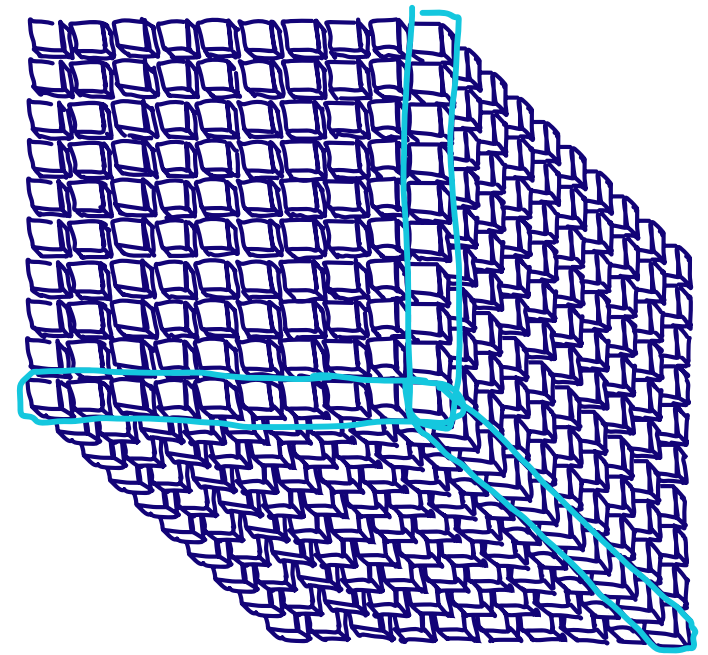


$c \in C$

Rate $\rho \mapsto \rho^m$

(Relative) Distance $\delta \mapsto \delta^m$

m-fold Tensor $C^m \subset \mathbb{F}^{n^m}$



$c \in C^m$

iff every axis-parallel line $\in C$

LTCs via Tensoring [Ben-Sasson-Sudan'04, Viderman'15, ...]

Plane Tester

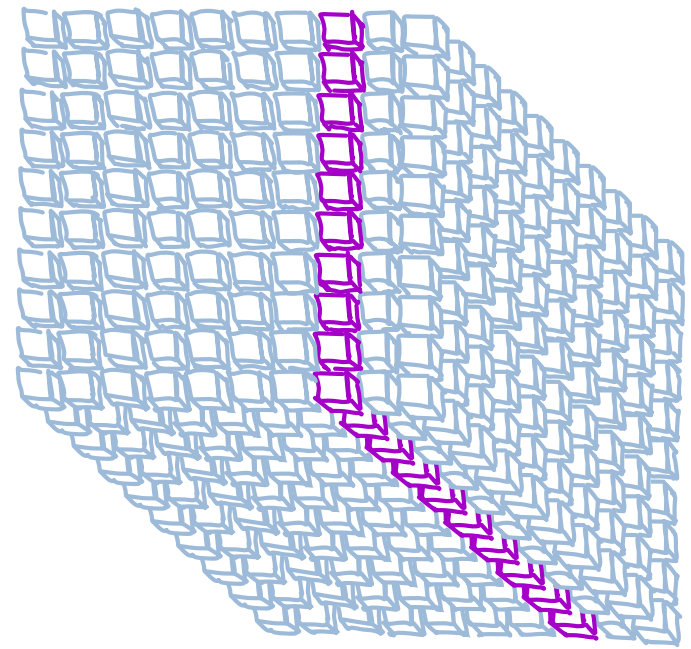
- 1- Sample a uniform plane
- 2- Test that it is in C^2

[Viderman'15]:

$$\Pr[T^c = 0] > \Omega(1) \cdot \Delta(c, C)$$

3-fold Tensor $C^3 \subset \mathbb{F}^{n^3}$

w.p. $1/3n$



$$c \in C^3$$

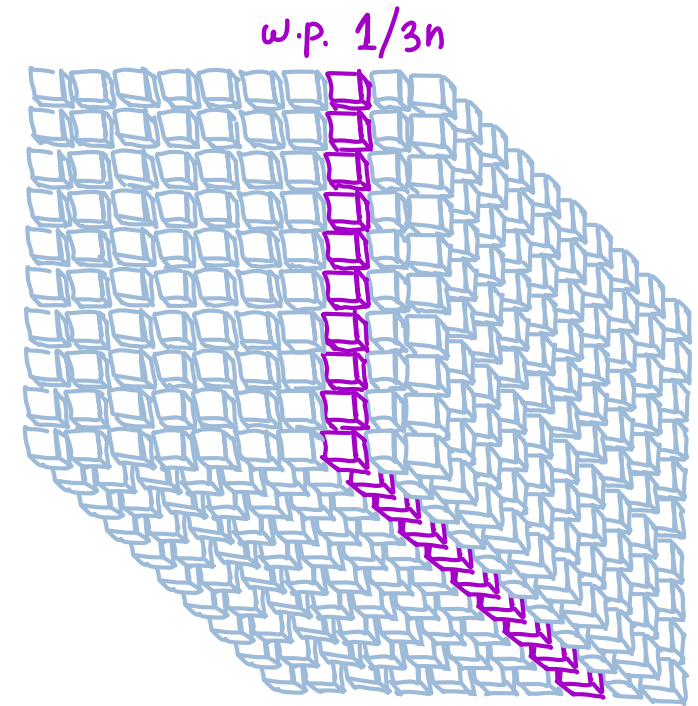
iff every axis-parallel line $\in C$

Testing Tree Code Tensors [M-Rosen-Rothblum'25]

Plane Tester

- 1- Sample a uniform plane
- 2- Test that it is in C^2

3-fold Tensor $T \in C^3 \subset \mathbb{F}^{n^3}$



$$c \in C^3$$

iff every axis-parallel line $\in C$

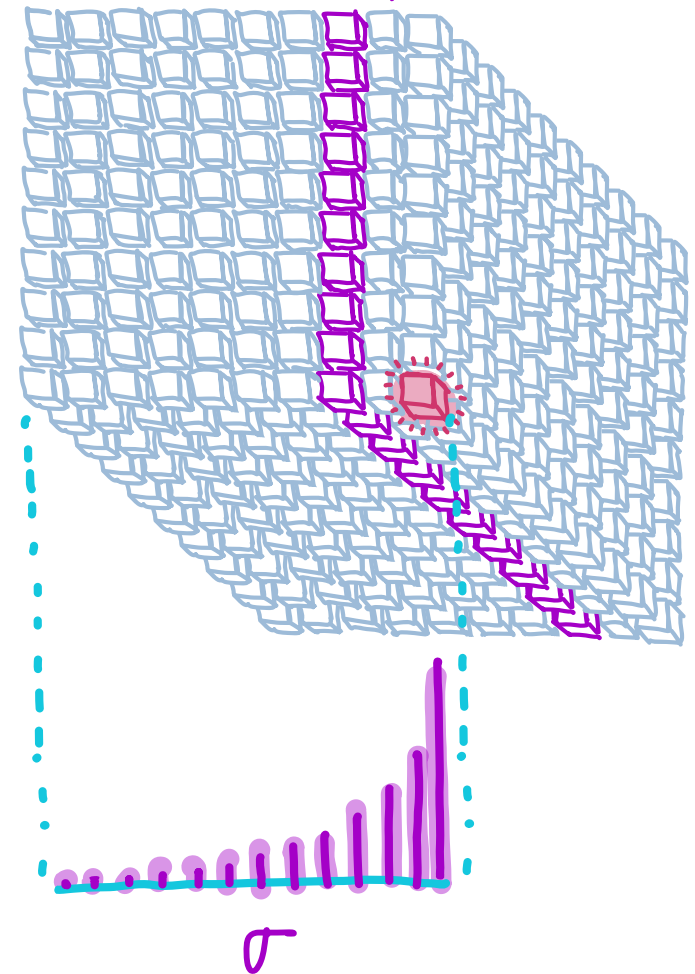
Testing Tree Code Tensors [M-Rosen-Rothblum'25]

"Skewed" Plane Tester

- 1- Sample a plane from σ
- 2- Test that it is in C^2

3-fold Tensor $TC^3 \subset \mathbb{F}^{n^3}$

w.p. $\propto 1/i$

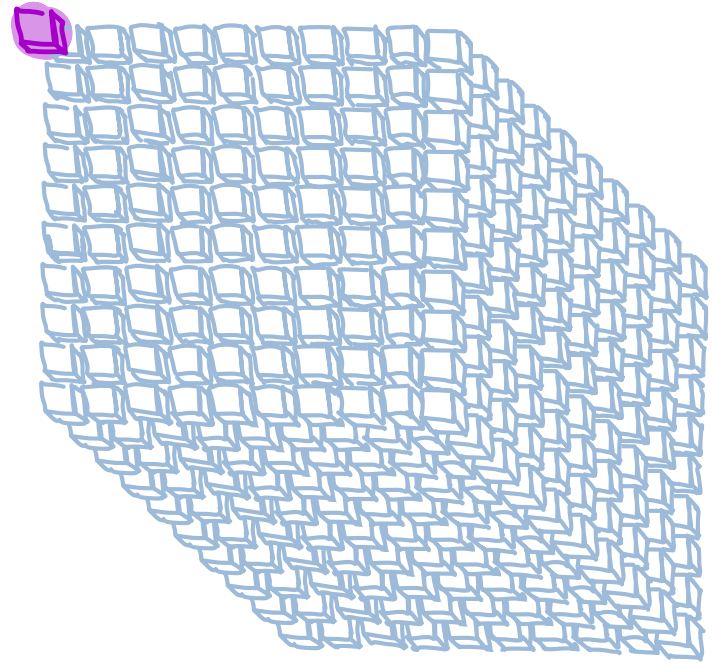


Testing Tree Code Tensors [M-Rosen-Rothblum'25]

"Skewed" Plane Tester

- 1- Sample a plane from $\mathcal{T}$
- 2- Test that it is in $\mathcal{C}^2$

3-fold Tensor $\mathcal{T} \in \mathbb{C}^3 \subset \mathbb{F}^{n^3}$

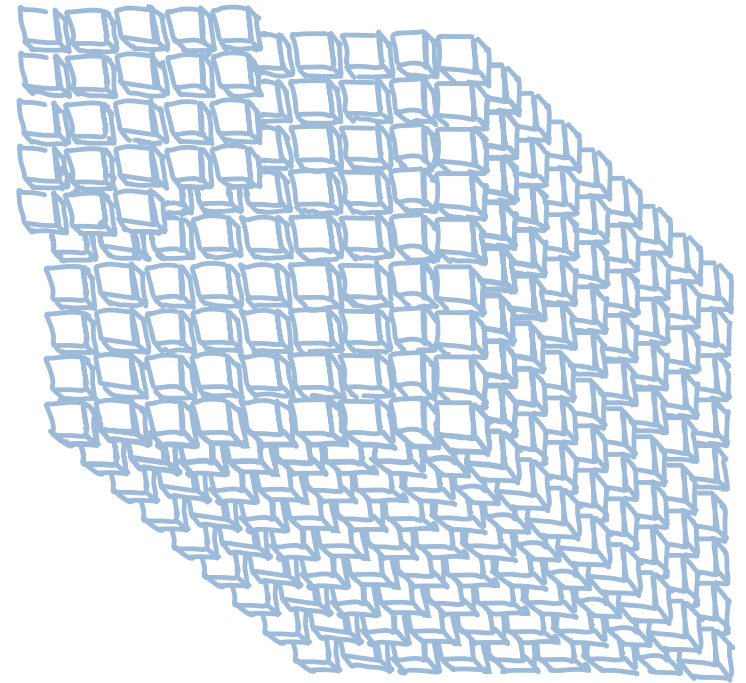


Testing Tree Code Tensors [M-Rosen-Rothblum'25]

"Skewed" Plane Tester

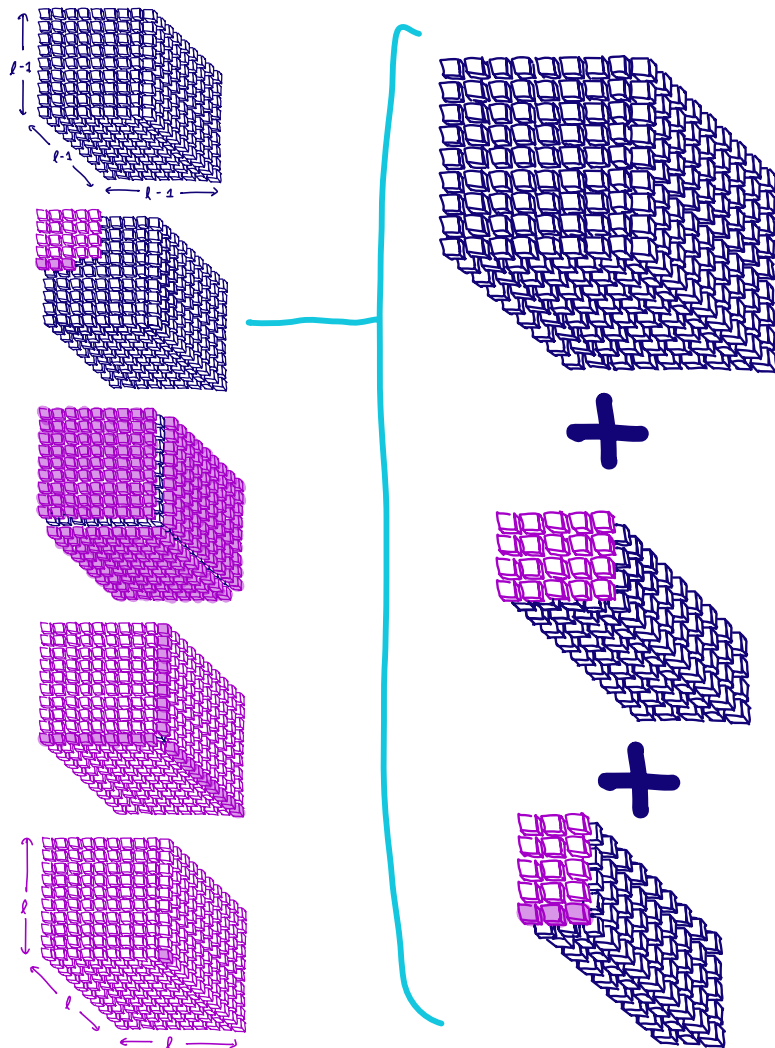
- 1- Sample a plane from $\mathcal{T}$
- 2- Test that it is in $\mathcal{C}^2$

A "Partial" Tensor

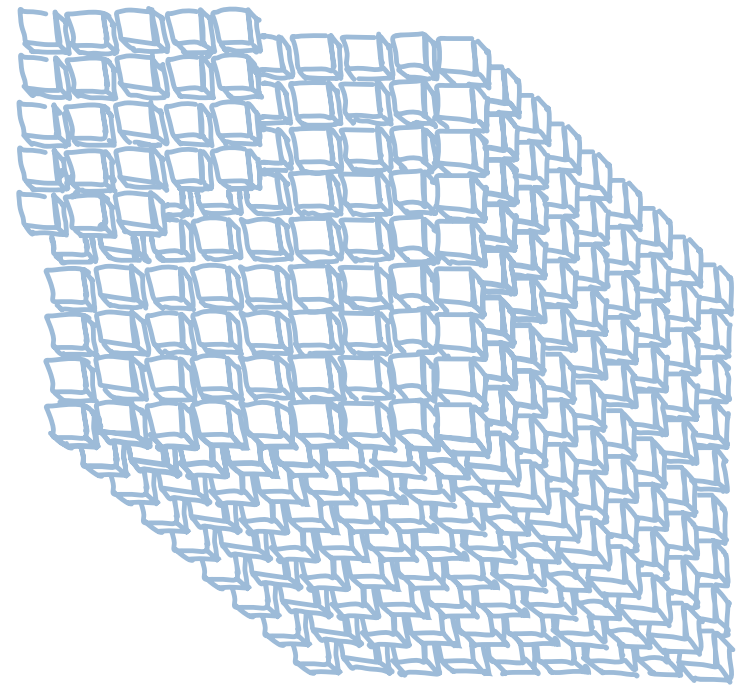


Testing Tree Code Tensors [M-Rosen-Rothblum'25]

Monotone embedding $\mathbb{N} \rightarrow \mathbb{N}^3$



A "Partial" Tensor



Locally Testable
Block Codes $\Rightarrow$ PCPs

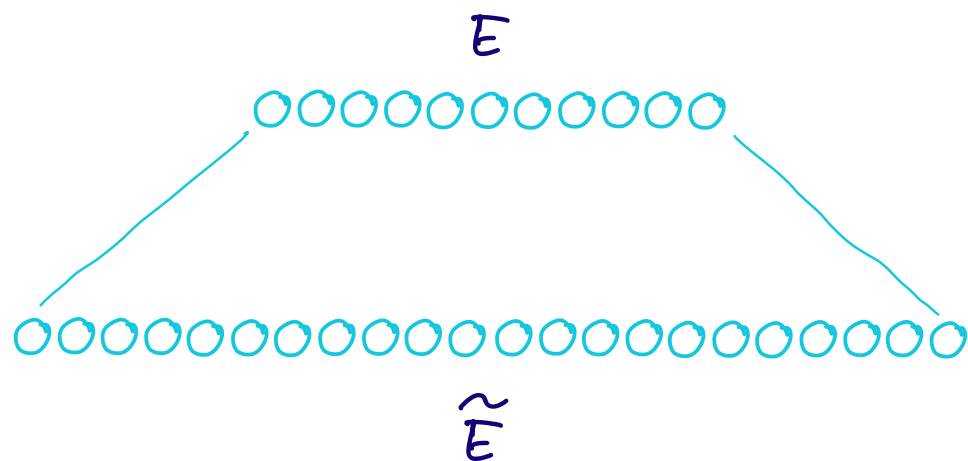
Locally Testable
Tree Codes $\Rightarrow$ Tree PCPs

PCP as Redundant Encoding [Babai-Fortnow-Levin-Szegedy '91]

Coding Theory

Redundant encoding of information.

Test if codeword $\tilde{E}$ is all zeros by reading few random locations.



PCPs

Redundant encoding of computation.

Represent statement as $E \equiv 0$.

PCP: encoding $\tilde{E}$.

To Verify:

1. Test that PCP is codeword.
 - Local Testability
2. Test that it is all zeros.

Tree PCP Outline

Witness for $(C, a, n) \in \text{CrktReach}$

$A = (a_0, w_1, a_1, w_2, a_2, \dots, w_n, a_n)$

□ □ □ □ □ □ □ □ □ □

THE PCP

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

$\tilde{A} = \text{TC}(A)$

Locally Testable



$E \neq 0 \iff \forall t, C(a_{t-1}, a_t, w_t) = 1$

E

○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

○ ○

$\tilde{E} = \text{TC}(E)$

Eval

"Tree Sumcheck"

1. Test that $\tilde{A}$ is a codeword.

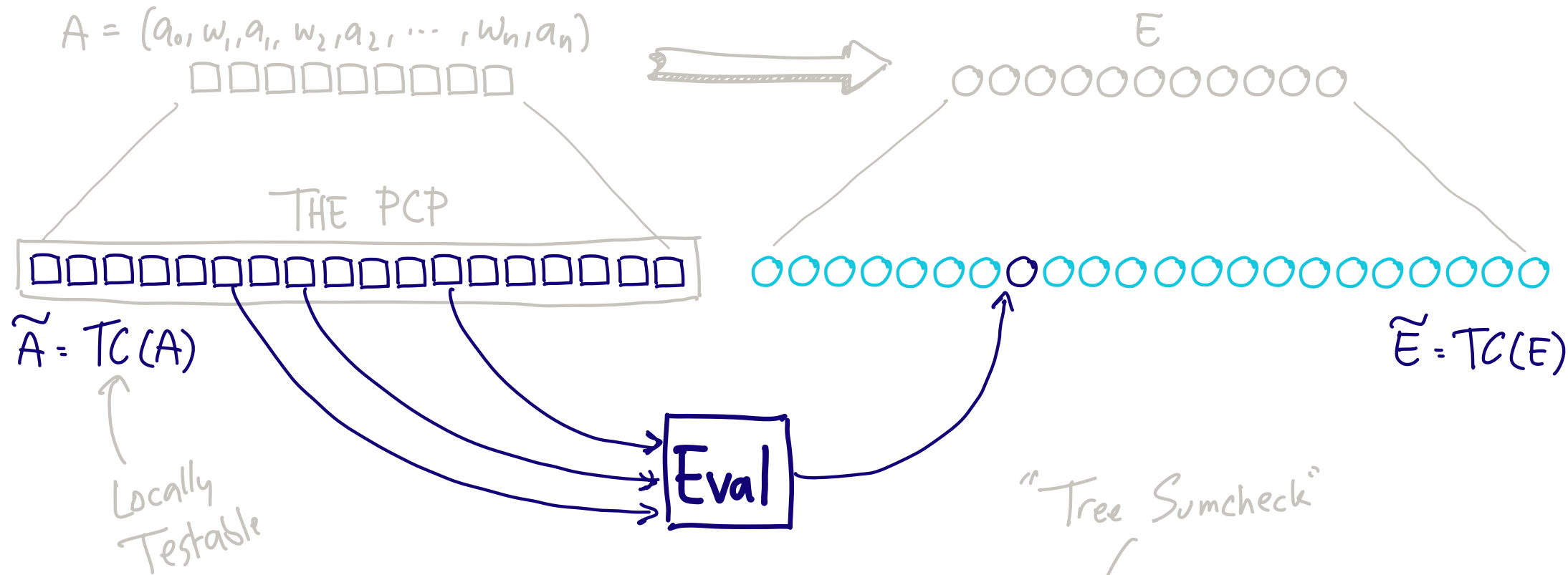
2. Test that $\tilde{E}$ encodes zeros.

- Simulate $\tilde{E}$ given $\tilde{A}$.

Tree PCP Outline

Witness for $(C, a, n) \in \text{CrktReach}$

$$E \neq 0 \iff \forall t, C(a_{t-1}, a_t, w_t) = 1$$

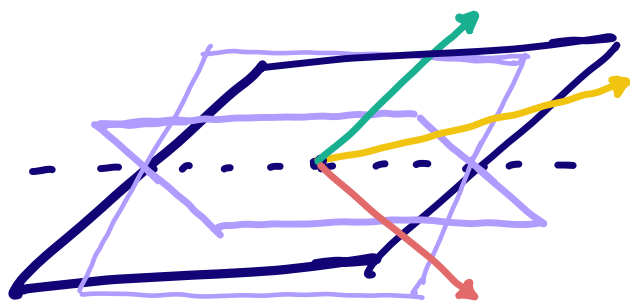


1. Test that $\tilde{A}$ is a codeword.

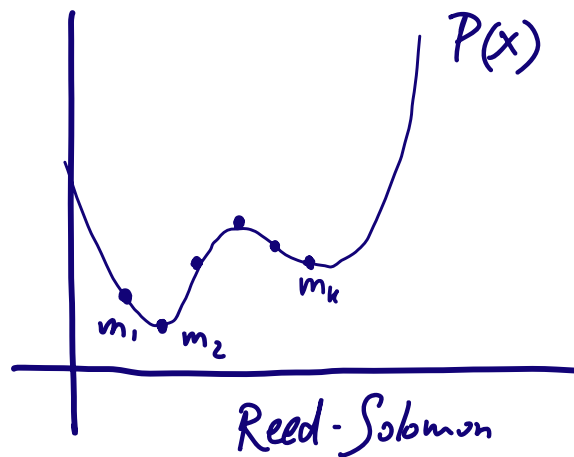
2. Test that $\tilde{E}$ encodes zeros.

- Simulate $\tilde{E}$ given $\tilde{A}$.

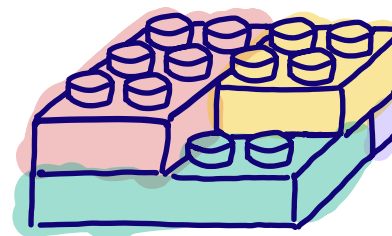
Block Codes



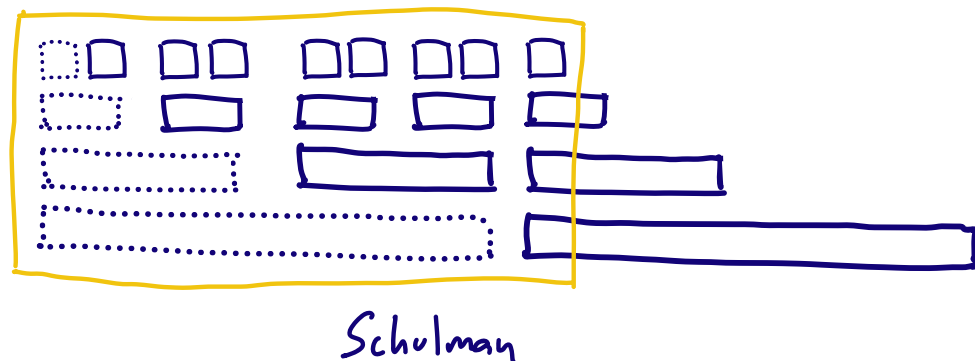
algebraic wonderland



Tree Codes



all usable codes involve
"combinatorial designs"



Open Questions

- Beyond bounded-space computation? (random access to history)
- Reduce query complexity?
- More "straight-forward" $\Rightarrow$ efficient constructions?
 - 100% algebraic tree codes?
 - "Tree IOPs"?
- "DAG PCPs"?

Thanks!