FRI and Proximity Proofs: What they are, what they are for, and the future

Dan Boneh
Stanford University

Some papers we discuss in this module

- FRI (2018) and analysis (2018): Fast Reed–Solomon Interactive Oracle Proofs of Proximity
- DEEP-FRI (2019): Out of domain sampling improves soundness
- BCIKS (2020): Proximity Gaps for Reed–Solomon Codes
- CircleSTARK (2024): FRI using a Mersenne prime
- STIR (2024): Reed–Solomon proximity testing with fewer queries
- WHIR (2024): Proximity testing with a faster verifier

Beyond Reed-Solomon codes (a few recent results):

- Breakdown (2021), Orion (2022): Polynomial commitments with a fast prover
- BaseFold (2023): Polynomial commitments from foldable codes with shorter proofs
- Blaze (2024): Fast SNARKs from Interleaved RAA Codes

FRI: Fast Reed-Solomon IOPP

- Let $\mathbb{F}$ be a finite field (say, $\mathbb{F} = \{0,1,2,\dots,p-1\}$) and $\mathcal{L} \subseteq \mathbb{F}$.
- Let $\mathbb{y}: \mathcal{L} \to \mathbb{F}$ be a committed function (a vector of size $|\mathcal{L}|$)

FRI: a way to prove that $\mathbb{y}$ is "close" to a Reed-Solomon codeword

So what? Who cares? What does this even mean?

Let's get started … first some background

Background

(1) Codes

(2) IOP and IOPP

(3) Poly-IOP

(1) Linear codes

Def: an $[n,k,l]_p$ **linear code** $\mathcal{C}$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$ of dimension k (so $|\mathcal{C}| = p^k$) where $|u|_0 \geq l$ for all $0 \neq u \in \mathcal{C}$

Recall: For u, v in $\mathbb{F}^n$

(sum as integers)

$|u|_0 :=$ (Hamming weight of u) $= \sum_{i=0}^{n} (u_i)^0$ (where $0^0 = 0$)

$\Delta(u,v) :=$ (relative Hamming distance) = $\frac{1}{n}|u - v|_0$ $\in [0,1]$

example: $\Delta\big((1,\mathbf{5},9,\mathbf{4},\mathbf{1}),\ (1,\mathbf{2},9,\mathbf{7},\mathbf{4})\big) = 3/5$

$\mu = \mu(\mathcal{C}) := l/n =$ (relative min weight of $\mathcal{C}$) $= \frac{1}{n} \cdot \min_{0 \neq u \in \mathcal{C}} |u|_0$ $\in [0,1]$

(1) Linear codes

Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a $[n, k, l]_p$ linear code. Then:

Fact 1: For all distinct $u, v \in \mathcal{C}$ we have $\Delta(u, v) \geq \mu(\mathcal{C}) = l/n$

(otherwise $0 \neq |u - v|_0 < l$ and $u - v \in \mathcal{C}$)

Fact 2: $k \leq n - l + 1$ (i.e. $|\mathcal{C}| \leq p^{n-l+1}$) (the singleton bound)

Def: if $k = n - l + 1$ then $\mathcal{C}$ is called an **MDS Code**

The classic MDS code: the Reed-Solomon code (more in a bit)

Encoding a message as a codeword

Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a $[n, k, l]_p$ linear code.

Encoding: Let $\boldsymbol{c_1}, \dots, \boldsymbol{c_k} \in \mathbb{F}^n$ be a basis of $\mathcal{C}$.

A message $m = (m_1, \dots, m_k) \in \mathbb{F}^k$ can be encoded as a codeword

$$\boxed{m \in \mathbb{F}^k} \xrightarrow{\text{encode}} \boxed{m_1\boldsymbol{c_1} + \cdots + m_k\boldsymbol{c_k} \in \mathbb{F}^n}$$

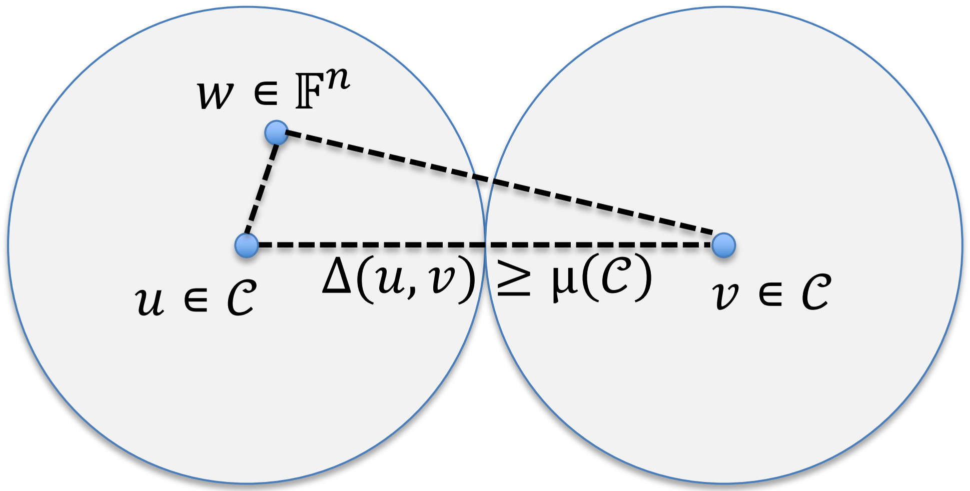
($1/\rho$ expansion)

We can treat $\mathcal{C}$ as a linear map $\mathcal{C}: \mathbb{F}^k \rightarrow \mathbb{F}^n$ that encodes messages in $\mathbb{F}^k$

Def: The **rate** of a code is $\rho \coloneqq k/n \in [0,1]$ (e.g., $\rho = 0.5$)

In practice: for fast encoding, want ρ as large as possible (ρ=0.5 $\Rightarrow$ n=$2k$)

Unique decoding distance ($[n,k,l]_p$ linear code)



Fact 3: for every $w \in \mathbb{F}^n$ there is at most one codeword $u \in \mathcal{C}$ s.t. $\Delta(u,w) < \mu(\mathcal{C})/2$

(by triangular inequality)

Def: $\mu(\mathcal{C})/2$ in $[0,0.5]$ is called the **unique decoding distance** of $\mathcal{C}$

Most $w \in \mathbb{F}^n$ are not uniquely decodable

$$\sum_{u \in \mathcal{C}} B_0(u, l/2) = \sum_{u \in \mathcal{C}} \binom{n}{l/2} p^{l/2} \leq p^{n-l+1} \cdot \binom{n}{l/2} p^{l/2} < n^{l/2} \cdot p^{n-l/2+1} \overset{n<p}{\ll} p^n$$

List decoding

Def: For a $[n, k, l]_p$ linear code $\mathcal{C}$, $w \in \mathbb{F}^n$, and $\delta \in [0,1]$, let

$$\text{List}[w, \mathcal{C}, \delta] := \{ c \in \mathcal{C} \ \text{ s.t. } \ \Delta(c, w) \leq \delta \}$$

Then $\delta < \mu(\mathcal{C})/2 \quad \Rightarrow \quad |\text{List}[w, \mathcal{C}, \delta]| \leq 1$

(unique decoding distance)

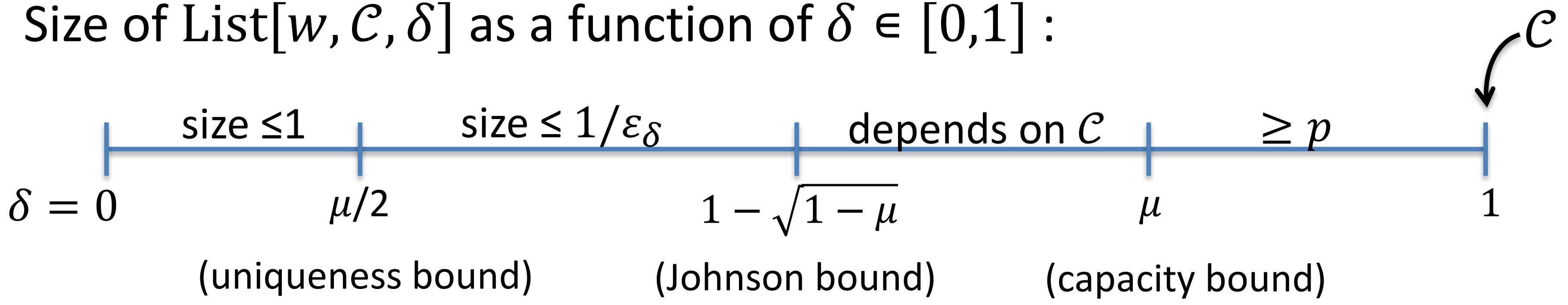
List decoding

The Johnson bound: For $\mathcal{C} \subseteq \mathbb{F}^n$, $w \in \mathbb{F}^n$, $0 < \delta < 1 - \sqrt{1-\mu}$

$$\left|\text{List}[w, \mathcal{C}, \delta]\right| \leq 1/\varepsilon_\delta \quad \text{where} \quad \varepsilon_\delta := 2\sqrt{1-\mu}\left(1 - \sqrt{1-\mu} - \delta\right)$$

(blows up as δ approaches $1 - \sqrt{1-\mu}$)

Size of $\text{List}[w, \mathcal{C}, \delta]$ as a function of $\delta \in [0,1]$:



Convenient terms: δ-close and δ-far

<u>Def:</u> We say that $w \in \mathbb{F}^n$ is **$\boldsymbol{\delta}$-close** to $\mathcal{C} \subseteq \mathbb{F}^n$

if there is some $c \in \mathcal{C}$ s.t. $\Delta(w, c) \leq \delta$

(i.e. $|\mathrm{List}[w, \mathcal{C}, \delta]| \geq 1$). We write $\Delta(w, \mathcal{C}) \leq \delta$.

<u>Def:</u> We say that $w \in \mathbb{F}^n$ is **$\boldsymbol{\delta}$-far** from $\mathcal{C} \subseteq \mathbb{F}^n$

if for all $c \in \mathcal{C}$ we have $\Delta(w, c) > \delta$

(i.e. $|\mathrm{List}[w, \mathcal{C}, \delta]| = 0$). We write $\Delta(w, \mathcal{C}) > \delta$.

The classic MDS code: Reed-Solomon

First, polynomials over a field $\mathbb{F}$

- $\mathbb{F}^{<d}[X]$: set of all univariate polynomials over $\mathbb{F}$ of degree $< d$
- For a polynomial $f \in \mathbb{F}^{<d}[X]$ and $\mathcal{L} \subseteq \mathbb{F}$

  write $\bar{f}: \mathcal{L} \to \mathbb{F}$ for the restriction of f to the domain $\mathcal{L}$

A function $w: \mathcal{L} \to \mathbb{F}$, where $n := |\mathcal{L}|$, can be treated as a vector

$$\mathrm{vec}(w) := \big(w(a_1), \ldots, w(a_n)\big) \in \mathbb{F}^n$$

where $\mathcal{L} = \{a_1, \ldots, a_n\} \subseteq \mathbb{F}$ has a natural ordering

The classic MDS code: Reed-Solomon

Def: The **Reed-Solomon code** over the field $\mathbb{F}$, evaluation domain $\mathcal{L} \subseteq \mathbb{F}$, and degree d, is the linear code

$$\mathrm{RS}[\mathbb{F}, \mathcal{L}, d] := \{\ \bar{f}: \mathcal{L} \to \mathbb{F} \quad \text{where} \quad f \in \mathbb{F}^{<d}[X]\ \}$$

Fact: Let $d < n := |\mathcal{L}|$.

$\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ is a $[n,\ d,\ l = (n - d + 1)]_p$ linear code

$\Rightarrow$ $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ is an MDS code (has p^d codewords)

Def: The **rate** of $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ is $\rho := d/n \in [0,1]$ (e.g., $\rho = 0.5$)

$$\boxed{m \in \mathbb{F}^d} \xrightarrow{\text{encode}} \boxed{\mathrm{vec}(\bar{f}_m) \in \mathbb{F}^n}$$

($1/\rho$ expansion)

Unique decoding and list decoding

Def: For $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$, $w: \mathcal{L} \to \mathbb{F}$, and $\delta \in [0,1]$, let

$$\mathrm{List}[w, d, \delta] := \{ \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, \mathrm{d}] \text{ s.t. } \Delta(\bar{f}, w) \leq \delta \}$$

So: $\delta < \frac{\mu}{2} = \frac{l}{2n} = \frac{n-d+1}{2n} \approx \frac{1-\rho}{2}$ $\Rightarrow$ $|\mathrm{List}[w, d, \delta]| \leq 1$

(unique decoding distance)

Recall: $\rho := d/n \in [0,1]$ where $n := |\mathcal{L}|$. For MDS code: $\mu \approx 1 - \rho$.

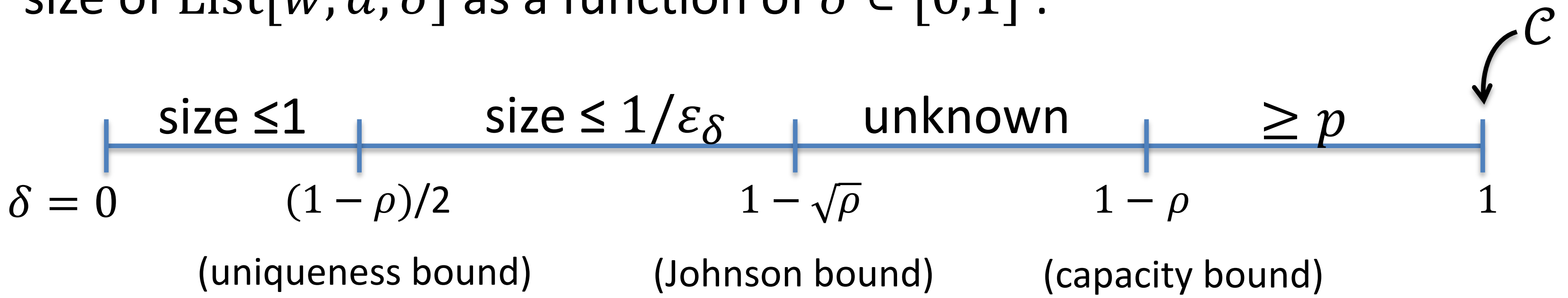
Unique decoding and list decoding

The Johnson bound: For $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$, $w: \mathcal{L} \to \mathbb{F}$, $\delta < 1 - \sqrt{\rho}$

$$\big|\mathrm{List}[w, d, \delta]\big| \le 1/\varepsilon_\delta \quad \text{where} \quad \varepsilon_\delta := 2\sqrt{\rho}(1 - \sqrt{\rho} - \delta) \in (0,1)$$

(blows up as δ approaches $1 - \sqrt{\rho}$)

size of $\mathrm{List}[w, d, \delta]$ as a function of $\delta \in [0,1]$:



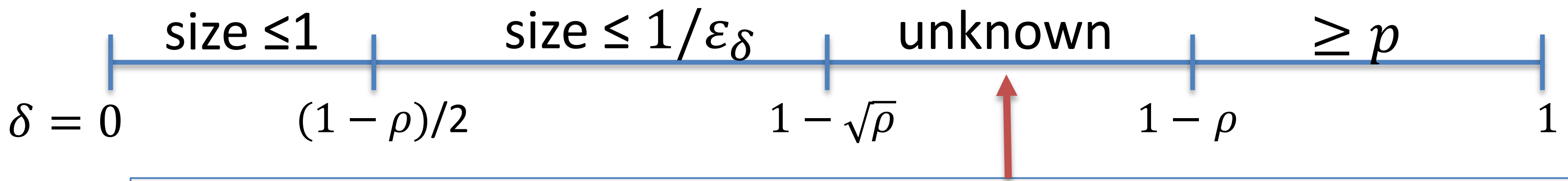
Unique decoding and list decoding

The Johnson bound: For $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$, $w\colon\mathcal{L}\to\mathbb{F}$, $\delta < 1-\sqrt{\rho}$

$$\left|\mathrm{List}[w,d,\delta]\right| \le 1/\varepsilon_\delta \quad \text{where} \quad \varepsilon_\delta := 2\sqrt{\rho}(1-\sqrt{\rho}-\delta) \in (0,1)$$

(blows up as δ approaches $1-\sqrt{\rho}$)

size of $\mathrm{List}[w,d,\delta]$ as a function of $\delta \in [0,1]$:



Conjectured to be poly(n) size (true for random $\mathcal{L} \subseteq \mathbb{F}$ [BGM'24])

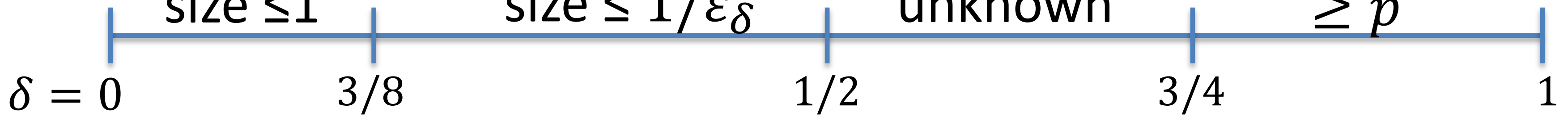
Unique decoding and list decoding

The Johnson bound: For $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$, $w:\mathcal{L}\to\mathbb{F}$, $\delta<1-\sqrt{\rho}$

$$\left|\mathrm{List}[w,d,\delta]\right|\le 1/\varepsilon_\delta \quad \text{where} \quad \varepsilon_\delta := 2\sqrt{\rho}(1-\sqrt{\rho}-\delta)\in(0,1)$$

(blows up as δ approaches $1-\sqrt{\rho}$)

size of $\mathrm{List}[w,d,\delta]$ as a function of $\delta\in[0,1]$:



An example: $\rho=1/4$

Background on IOPs

Review

(1) IOP and IOPP

(2) Poly-IOP

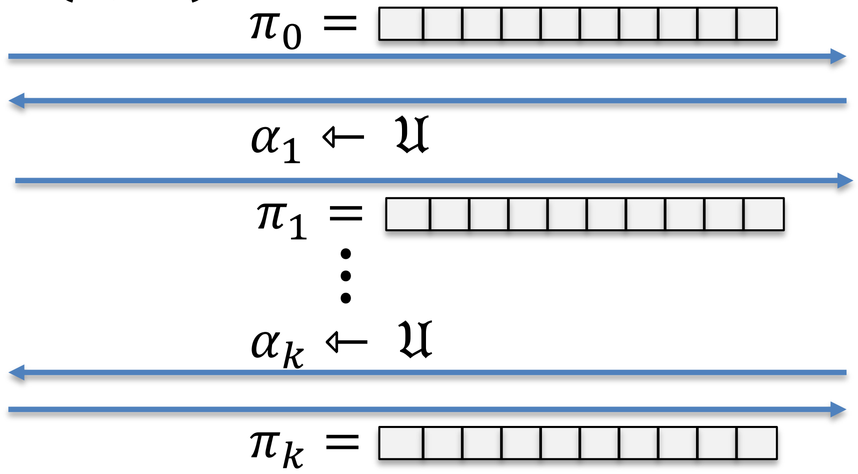
Interactive Oracle Proofs (IOP) [BCS'16, RRR'16]

Let $R = \{(\mathbb{x}, \mathbb{w})\}$ be a poly-time relation (e.g., $\mathbb{x} = \text{sha3}(\mathbb{w})$)

Def: an IOP for R is a pair of algorithms $(\boldsymbol{P}, \boldsymbol{V})$ s.t.:

Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{w})$

Verifier$(\mathbb{x})$



$\pi_0 =$
$\alpha_1 \leftarrow \mathfrak{U}$
$\pi_1 =$
$\vdots$
$\alpha_k \leftarrow \mathfrak{U}$
$\pi_k =$

α_i: short random challenges

π_i: poly-size strings (oracles)

$\boldsymbol{V}$ can query for cells of π_i

$\boldsymbol{V}^{\pi_0,\ldots,\pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) \rightarrow$ yes/no

Interactive Oracle Proofs (IOP) [BCS'16, RRR'16]

Let $R = \{(\mathbb{x}, \mathbb{w})\}$ be a poly-time relation (e.g., $\mathbb{x} = \mathrm{sha3}(\mathbb{w})$)

Def: an IOP $(\boldsymbol{P}, \boldsymbol{V})$ for R

is **complete** if for all $(\mathbb{x}, \mathbb{w}) \in R$, when $\boldsymbol{V}$ interacts with $\boldsymbol{P}$

$$\Pr[\ \boldsymbol{V}^{\pi_0,\dots,\pi_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) = \text{yes}\ ] = 1$$

is **sound** if for all $\boldsymbol{P}^*$ and $\mathbb{x} \notin L(R) := \{\mathbb{x} \mid \exists \mathbb{w}: (\mathbb{x}, \mathbb{w}) \in R\}$

$$\Pr[\ \boldsymbol{V}^{\pi_0,\dots,\pi_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) = \text{yes}\ ] < err \qquad (\approx 2^{-128})$$

is **knowledge sound** (informally) if for all $\boldsymbol{P}^*$,

$\boldsymbol{V}$ accepts $\mathbb{x}$ $\Rightarrow$ prover "knows" $\mathbb{w}$ s.t. $(\mathbb{x}, \mathbb{w}) \in R$

Interactive Oracle Proofs (IOP) [BCS'16, RRR'16]

Let $R = \{(\mathbb{x}, \mathbb{w})\}$ be a poly-time relation (e.g., $\mathbb{x} = \text{sha3}(\mathbb{w})$)

Def: an IOP $(\boldsymbol{P}, \boldsymbol{V})$ for R

is **complete** if for all $(\mathbb{x}, \mathbb{w}) \in R$, when $\boldsymbol{V}$ interacts with $\boldsymbol{P}$

$$\Pr[\ \boldsymbol{V}^{\pi_0,\dots,\pi_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) = \text{yes}\ ] = 1$$

is **sound** if for all $\boldsymbol{P}^*$ and $\mathbb{x} \notin L(R) := \{\mathbb{x} \mid \exists \mathbb{w}: (\mathbb{x}, \mathbb{w}) \in R\}$

$$\Pr[\ \boldsymbol{V}^{\pi_0,\dots,\pi_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) = \text{yes}\ ] < err \qquad (\approx 2^{-128})$$

is **succinct** if $\text{time}(\boldsymbol{V})$ is at most $\text{polylog}(\text{time}(R))$ and $O(|\mathbb{x}|, \log(1/err))$

$\Rightarrow$ k is small and $\boldsymbol{V}$ makes few queries to the oracles $\pi_0, \dots, \pi_k$

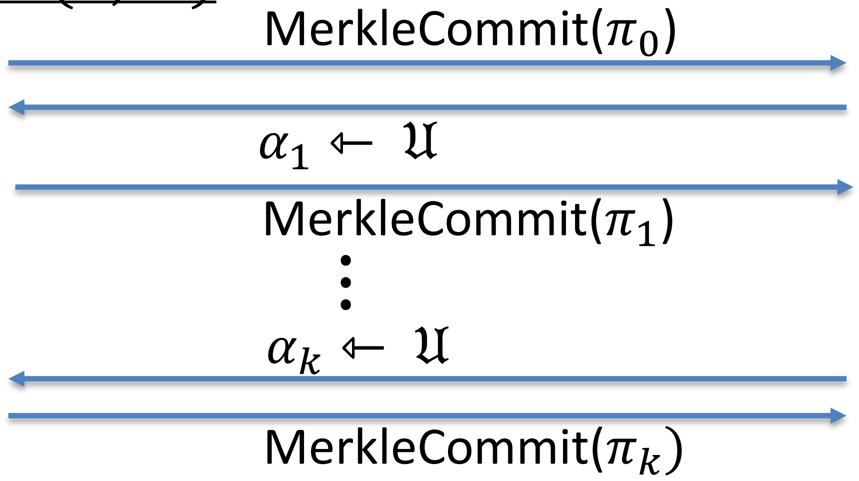
IOP for R $\Rightarrow$ SNARK for R (the BCS'16 compiler)

Step 1: replace $\pi_0, \ldots, \pi_k$ by Merkle commitments

We obtain an interactive proof (IP)

Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{w})$

Verifier$(\mathbb{x})$



Security now depends on collision resistance of Merkle hash function

$\boldsymbol{V}^{\pi_0,\ldots,\pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) \rightarrow$ yes/no

$\boldsymbol{V}$ queries π_i at cell j $\Rightarrow$ $\boldsymbol{P}$ responds with a Merkle proof for cell j

IOP for R $\Rightarrow$ SNARK for R (the BCS'16 compiler)

Step 2: Make non-interactive using the Fiat-Shamir transform

Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{w})$

$c_0 \coloneqq \text{MerkleCommit}(\pi_0)$

$\alpha_1 \leftarrow Hash(\mathbb{x}, c_0)$

$c_1 \coloneqq \text{MerkleCommit}(\pi_1)$

$\vdots$

$\alpha_k \leftarrow Hash(\mathbb{x}, c_0, \alpha_1, c_1 \ldots, c_{k-1})$

$c_k \coloneqq \text{MerkleCommit}(\pi_k)$

$\boxed{\boldsymbol{V}^{\pi_0,\ldots,\pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k)}$

MerkleProofs (one per $\boldsymbol{V}$ query)

SNARK Proof

IOP for R $\Rightarrow$ SNARK for R (the BCS'16 compiler)

"Thm" (BCS'16, CCH+'19, Hol'19):

the IOP has round-by-round soundness

$\Rightarrow$

the derived SNARG is secure in the random oracle model

(see also Chiesa-Yogev SNARK book)

Efficiency:

- To reduce prover work: minimize $|\pi_0| + \cdots + |\pi_k|$
- To reduce proof size: minimize $\underbrace{k}_{\Rightarrow \text{ Merkle Commitments}}$ and number of $\underbrace{\text{verifier queries}}_{\Rightarrow \text{ Merkle Proofs } (O_\lambda(\log|\pi_i|) \text{ size})}$

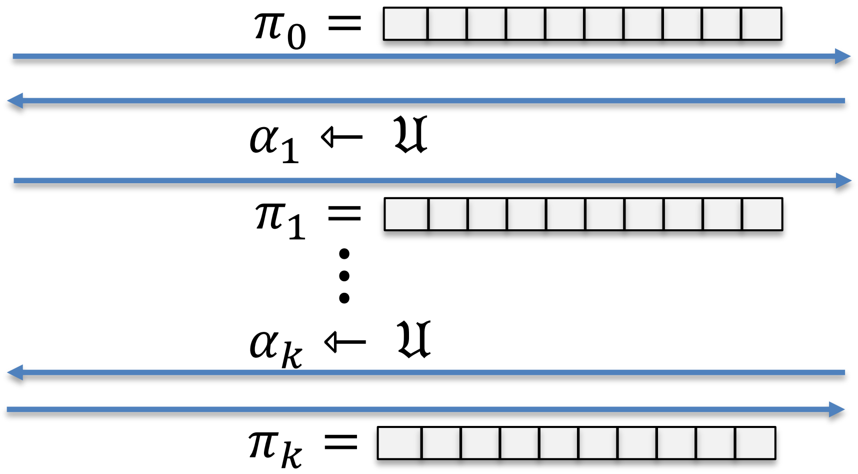
A generalization: IOP of Proximity (IOPP)

Let $R = \{(\mathbb{x}, \mathbb{y}, \mathbb{w})\}$ be a poly-time relation ($\mathbb{y} = \square\square\square\square\square\square\square\square\square\square$)

Def: an IOPP for R is a pair of algorithms $(\boldsymbol{P}, \boldsymbol{V})$ s.t.:

Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{y}, \mathbb{w})$

Verifier$^{\mathbb{y}}(\mathbb{x})$



$\pi_0 =$
$\alpha_1 \leftarrow \mathfrak{U}$
$\pi_1 =$
$\vdots$
$\alpha_k \leftarrow \mathfrak{U}$
$\pi_k =$

$\mathbb{y}, \pi_i$: poly-size strings (oracles)

$\boldsymbol{V}$ can query for cells of $\mathbb{y}, \pi_i$

The IOPP proves properties of $\mathbb{x}$ and a committed $\mathbb{y}$

$\boldsymbol{V}^{\mathbb{y}, \pi_0, \ldots, \pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) \rightarrow$ yes/no

Completeness and proximity soundness

Let $R = \{(\mathbb{x}, \mathbb{y}, \mathbb{w})\}$ be a poly-time relation ($\mathbb{y} =$)

Def: $(\mathbb{x}, \mathbb{y})$ is **δ-far from R**, if $(\mathbb{x}, \mathbb{y}', \mathbb{w}) \notin R$ for all $\mathbb{y}', \mathbb{w}$ with $\Delta(\mathbb{y}, \mathbb{y}') \leq \delta$

<u>Def</u>: an IOPP $(\boldsymbol{P}, \boldsymbol{V})$ for R

is **complete** if for all $(\mathbb{x}, \mathbb{y}, \mathbb{w}) \in R$ the Verifier $\boldsymbol{V}$ always accepts $\boldsymbol{P}$

is **δ-sound** if for all $(\mathbb{x}, \mathbb{y})$ that are δ-far from R:

$\forall \boldsymbol{P}^*$: $\Pr[\, \boldsymbol{V}^{\mathbb{y}, \pi_0, \ldots, \pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) = \text{yes} \,] < err$ $(\approx 2^{-128})$

if $(\mathbb{x}, \mathbb{y})$ is neither, then no guarantee on the output of $\boldsymbol{V}$

An important example: a Reed-Solomon IOPP

Let $\mathcal{C} = \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$, $u : \mathcal{L} \to \mathbb{F}$, and $\delta \in [0,1]$

Def: an IOPP for RS, a δ-**RS-IOPP**, is an IOPP $(\boldsymbol{P}, \boldsymbol{V})$ such that

$\boldsymbol{P}(\mathbb{x} = \mathcal{C}, \mathbb{y} = u, \mathbb{w} = \perp)$ Verifier$^{\mathbb{y}}(\mathbb{x} = \mathcal{C})$

complete: $u \in \mathcal{C} \quad \Rightarrow \quad$ for $\boldsymbol{P}$: $\Pr[\boldsymbol{V}^{u, \pi_0, \ldots, \pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) = \text{yes}\,] = 1$

δ-sound: $\Delta(u, \mathcal{C}) > \delta \;\Rightarrow\; \forall \boldsymbol{P}^* \colon \Pr[\boldsymbol{V}^{u, \pi_0, \ldots, \pi_k}(\mathbb{x}, \alpha_1, \ldots, \alpha_k) = \text{yes}\,] < err$

FRI is an efficient RS-IOPP. But why is this useful?

A special type of IOP: Poly-IOP

Let $R = \{(\mathbb{x}, \mathbb{w})\}$ be a poly-time relation

<u>Def</u>: a Poly-IOP for R is a pair of algorithms $(\boldsymbol{P}, \boldsymbol{V})$ s.t.:

<u>Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{w})$</u> <u>Verifier$(\mathbb{x})$</u>

$$\xrightarrow{f_0 \in \mathbb{F}^{<d}[X]}$$

$$\xleftarrow[\alpha_1 \leftarrow \mathbb{F}]{}$$

$$\xrightarrow[f_1 \in \mathbb{F}^{<d}[X]]{}$$

$\vdots$

$$\xleftarrow{\alpha_k \leftarrow \mathbb{F}}$$

$$\xrightarrow[f_k \in \mathbb{F}^{<d}[X]]{}$$

$f_0, \dots, f_k$: oracles for polynomials

$\boldsymbol{V}$ can eval f_i at any $x \in \mathbb{F}$

$\boldsymbol{V}^{f_0,\dots,f_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) \rightarrow$ yes/no

A special type of IOP: Poly-IOP

Let $R = \{(\mathbb{x}, \mathbb{w})\}$ be a poly-time relation

Def: a Poly-IOP for R is a pair of algorithms $(\boldsymbol{P}, \boldsymbol{V})$ s.t.:

Prover $\boldsymbol{P}(\mathbb{x}, \mathbb{w})$

Verifier$(\mathbb{x})$

Completeness and soundness as for an IOP

$$\xleftarrow{\alpha_k \leftarrow \mathfrak{U}}$$

$$\xrightarrow[f_k \in \mathbb{F}^{<d}[X]]{}$$

$\boldsymbol{V}^{f_0,\dots,f_k}(\mathbb{x}, \alpha_1, \dots, \alpha_k) \to$ yes/no

Why Poly-IOP?

Many SNARKs are derived from Poly-IOPs:

either univariate (e.g., Plonk) or multilinear (e.g., HyperPlonk)

Can we compile a Poly-IOP to a SNARK?

Yes!   And the key ingredient is an IOPP

Compiling a Poly-IOP to a SNARK Using a Reed-Solomon IOP of Proximity

An important application of an RS-IOPP

Poly-IOP for R $\Rightarrow$ SNARK for R

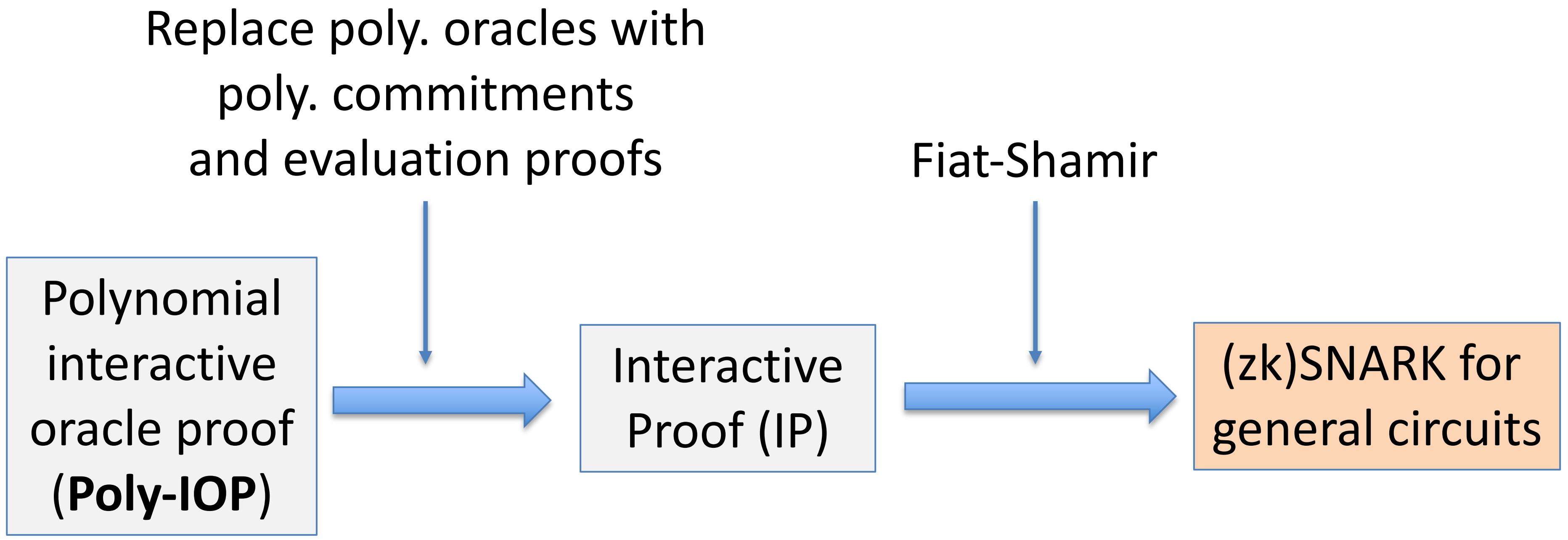
Derive a SNARK in two steps:

- Replace $f_0, \ldots, f_k$ by a **polynomial commitment** to same;

  now queries from $\boldsymbol{V}$ are replaced by polynomial evaluations.

- Apply the Fiat-Shamir transformation.

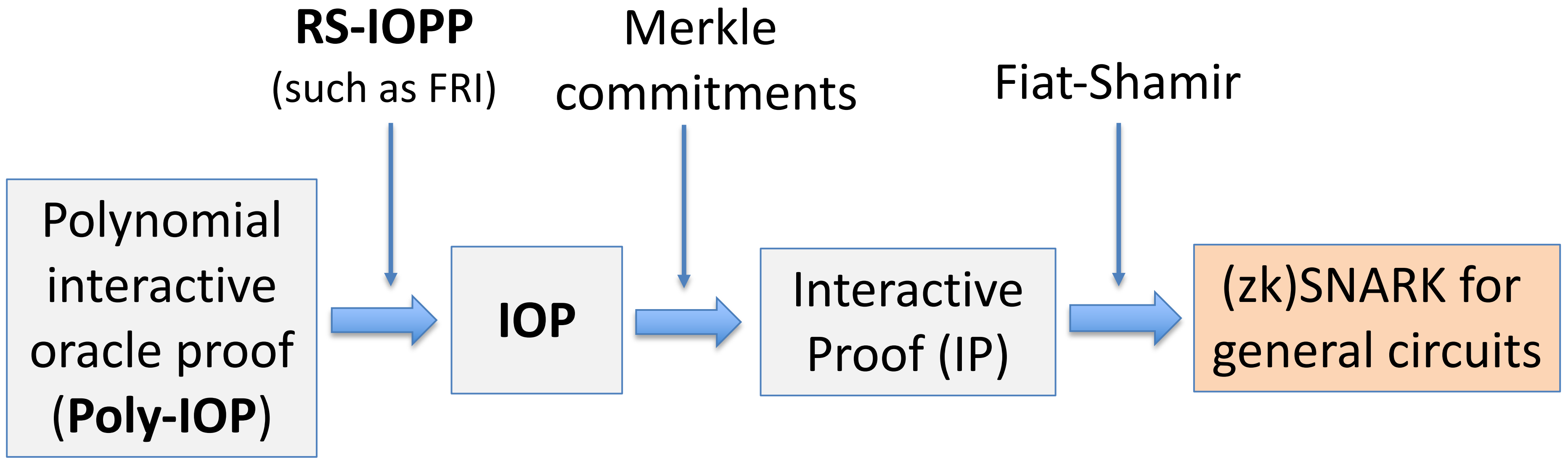
Poly-IOP for R $\Rightarrow$ SNARK for R


Replace poly. oracles with poly. commitments and evaluation proofs
Fiat-Shamir
Polynomial interactive oracle proof (**Poly-IOP**)
Interactive Proof (IP)
(zk)SNARK for general circuits

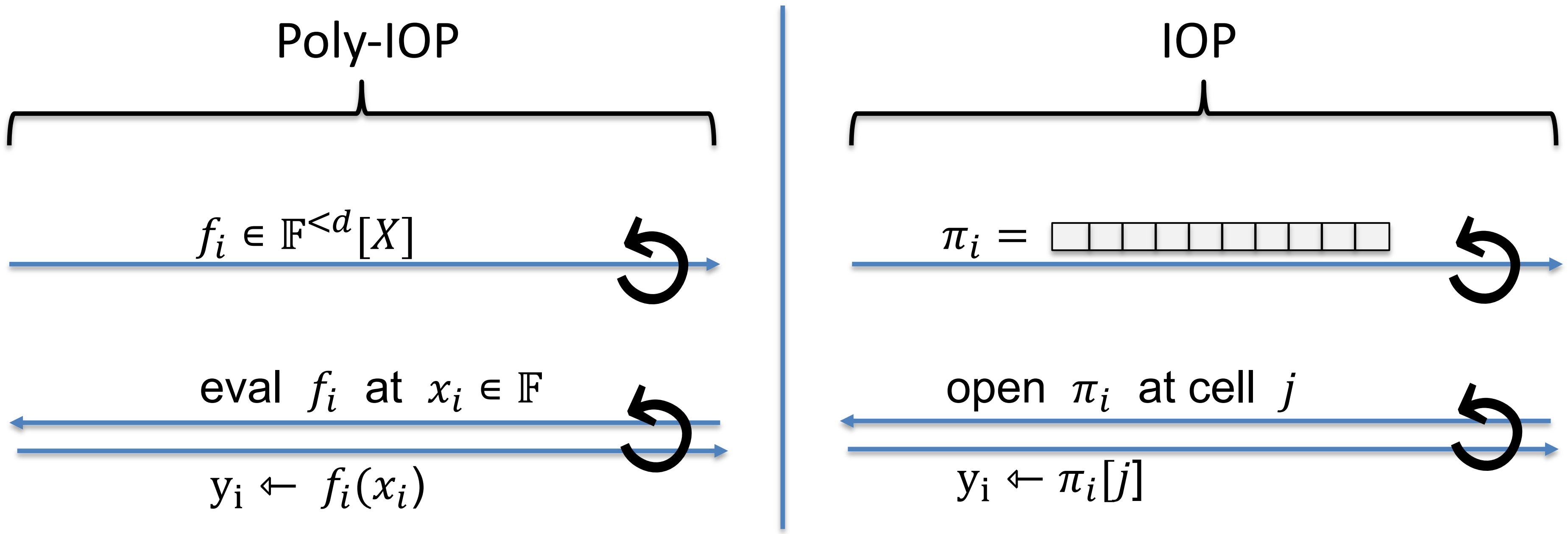
We will show: RS-IOPP $\Rightarrow$ poly. commitment scheme (PCS)

Poly-IOP ⇒ IOP ⇒ SNARK

A direct SNARK construction:



The interesting step: Poly-IOP $\Rightarrow$ IOP


Poly-IOP
$f_i \in \mathbb{F}^{<d}[X]$
eval f_i at $x_i \in \mathbb{F}$
$y_i \leftarrow f_i(x_i)$
IOP
$\pi_i =$
open π_i at cell j
$y_i \leftarrow \pi_i[j]$

Challenge: how to build a polynomial eval oracle from a list lookup oracle??

Representing a polynomial as an IOP oracle

The problem: $f \in \mathbb{F}^{<d}[X] \quad \rightarrow \quad$ string π: $\in \mathbb{F}^n$

Let $\mathcal{C} = \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ with $\mathcal{L} = \{a_1, \ldots, a_n\}$ $(d < n)$

- The honest prover represents $f \in \mathbb{F}^{<d}[X]$ by its encoding

$$f \quad \rightarrow \quad \pi = \big(f(a_1), f(a_2), \ldots, f(a_n)\big) = \bar{f} \in \mathcal{C} \subseteq \mathbb{F}^n$$

We will treat π as a function $\pi: \mathcal{L} \rightarrow \mathbb{F}$

New problem: in a Poly-IOP the prover can only send $f \in \mathbb{F}^{<d}[X]$,
but now the prover can send any $\pi: \mathcal{L} \rightarrow \mathbb{F}$, possibly not in $\mathcal{C}$

Representing a polynomial as an IOP oracle

The new problem: prover sends an oracle $\pi : \mathcal{L} \to \mathbb{F}$

- Can Verifier confirm that π is a codeword in $\mathcal{C}$ by only opening a few cells in π ??

  - Can't be done (what if π is wrong in only one cell?)
  - But Verifier can confirm that π is δ-close to some codeword, for δ<(unique decoding distance) $\Rightarrow$ π represents a unique poly.

How to check? Reed-Solomon IOPP (e.g., FRI)

But this is not yet a PCS. First, let's develop some tools …

Quotienting

Let $a \in \mathbb{F}$ s.t. $a \notin \mathcal{L}$ and let $b \in \mathbb{F}$. Let $f \in \mathbb{F}^{<d}[X]$ and $\delta \in [0,1]$.

Define the quotient map: $\boxed{u : \mathcal{L} \to \mathbb{F} \quad \to \quad q(X) := \frac{u(X)-b}{X-a} : \mathcal{L} \to \mathbb{F}}$

<u>Fact 1</u>: if $u = \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ and $b = f(a)$ then $q \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$

<u>Fact 2</u>: Suppose that for all $\bar{g} \in \mathrm{List}[u, d, \boldsymbol{\delta}]$ we have $b \neq g(a)$.
Then q is $\boldsymbol{\delta}$-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$.

Proof: Suppose $\Delta(q, \bar{h}) \leq \delta$ for some $h \in \mathbb{F}^{<d-1}[X]$ (i.e. $\bar{h} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$).
Set $g(X) := h(X) \cdot (X - a) + b$. Then $\bar{g} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ and $\Delta(u, \bar{g}) \leq \delta$.
But then $\bar{g} \in \mathrm{List}[u, d, \boldsymbol{\delta}]$ and $g(a) = b$. Contradiction!

Visualizing Quotienting

The quotient map for $a \in \mathbb{F} \setminus \mathcal{L}$: $u: \mathcal{L} \to \mathbb{F} \quad \to \quad q(X) := \frac{u(X)-b}{X-a} : \mathcal{L} \to \mathbb{F}$

Honest prover

$u = \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$

and $b = f(a)$

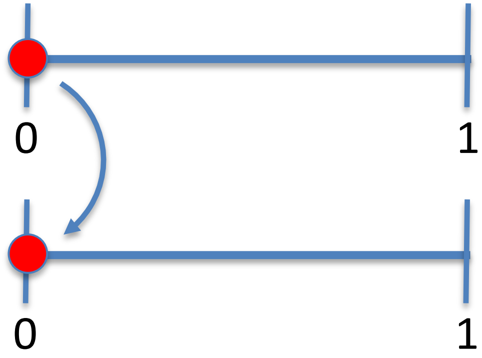
Dishonest prover

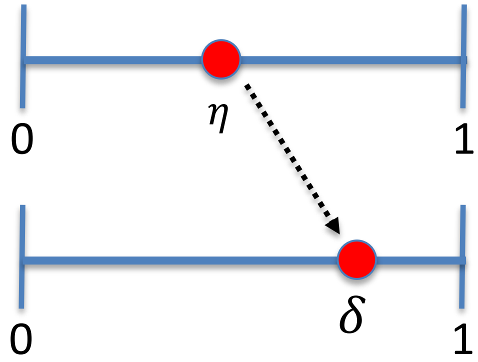
$\Delta(u, \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]) = \eta$ and

$\forall \bar{g} \in \mathrm{List}[u, d, \boldsymbol{\delta}]$: $b \neq g(a)$

distance u to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$:

distance q to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$:





Quotienting by more values

Let $\{a_1, \ldots, a_k\} \subseteq \mathbb{F} \setminus \mathcal{L}$ and $\{b_1, \ldots, b_k\} \subseteq \mathbb{F}$. Let $f: \mathcal{L} \to \mathbb{F}$.

Define polynomials $V(X)$, $I(X) \in \mathbb{F}^{\leq k}[X]$ as

$$V(X) := \prod_{i \in [k]} (X - a_i) \quad \text{and} \quad I(a_i) = b_i \text{ for all } i \in [k] .$$

Define the map: $\boxed{u: \mathcal{L} \to \mathbb{F} \quad \to \quad q(X) := \frac{u(X) - I(X)}{V(X)} : \mathcal{L} \to \mathbb{F}}$

Fact 1: if $u = \bar{f}$ and $b_i = f(a_i)$ for $i \in [k]$ then $q \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d - k]$

Fact 2: (STIR, Lemma 4.4) Suppose that for every $\bar{g} \in \mathrm{List}[u, d, \boldsymbol{\delta}]$ we have that $b_i \neq g(a_i)$ for some $i \in [k]$.

Then $q(X)$ is $\boldsymbol{\delta}$-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d - k]$.

Poly-IOP $\Rightarrow$ IOP: first attempt

P $\xrightarrow{f \in \mathbb{F}^{<d}[X]}$ V

eval f at $a_1 \in \mathbb{F} \setminus \mathcal{L}$ $\longleftarrow$

$\longrightarrow$ $b_1 \leftarrow f(a_1)$

eval f at $a_2 \in \mathbb{F} \setminus \mathcal{L}$ $\longleftarrow$

$\longrightarrow$ $b_2 \leftarrow f(a_2)$

P $\xrightarrow{\pi = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{} : \mathcal{L} \to \mathbb{F}}$ V

honest prover: $\pi := \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$

δ-RS-IOPP for $q_1(X) := \frac{\pi(X) - b_1}{X - a_1} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$ $\longleftrightarrow$

for some $\delta \in [0,1]$ (q_1 is easy to calculate from π)

δ-RS-IOPP for $q_2(X) := \frac{\pi(X) - b_2}{X - a_2} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-1]$ $\longleftrightarrow$

Verifier can conclude: there are $\bar{f_1}, \bar{f_2} \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ s.t. $\begin{cases} f_1(a_1) = b_1 \\ f_2(a_2) = b_2 \end{cases}$

Insufficient! What if $f_1 \neq f_2$? (can happen if $\boldsymbol{\delta}$>unique decoding distance)

A simple observation (DEEP)

$$P \xrightarrow{\quad f \in \mathbb{F}^{<d}[X] \quad} V$$

Claim: (r, s) commits prover to one $f \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$

$$P \xrightarrow{\quad \pi = \boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{}\boxed{} : \mathcal{L} \to \mathbb{F} \quad} V$$

Honest prover: $\pi := \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$

$$P \xleftarrow{\quad \text{sample random } r \in \mathbb{F} \setminus \mathcal{L} \quad} V$$

$$P \xrightarrow{\qquad\qquad} V$$

return $s \in \mathbb{F}$. Honest prover: $s := f(r)$

Fact: Let BAD be the event that $\exists\, \bar{f_1} \neq \bar{f_2} \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ s.t. $f_1(r) = f_2(r) = s$

$$\Pr_r\big[\mathrm{BAD}\big] \leq \underbrace{\binom{|\mathrm{List}[\boldsymbol{\pi}, d, \delta]|}{2}}_{\text{union bound over all pairs}} \cdot \underbrace{\frac{d}{|\mathbb{F}| - |\mathcal{L}|}}_{\Pr[\mathrm{BAD}] \text{ for a fixed } f_1, f_2}$$

A simple observation

Fact: Let BAD be the event that $\exists\, \bar{f_1} \neq \bar{f_2} \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ s.t. $f_1(r) = f_2(r) = s$

$$\Pr_r[\mathrm{BAD}] \leq \binom{|\mathrm{List}[\boldsymbol{\pi},d,\delta]|}{2} \cdot \frac{d}{|\mathbb{F}|-|\mathcal{L}|}$$

When $\delta < 1 - \sqrt{\rho}$ (Johnson bound) then $\left|\mathrm{List}[\boldsymbol{\pi}, d, \delta]\right| <$ const$_\delta$

$\Rightarrow$ If $\mathbb{F}$ is sufficiently large then $\Pr[\mathrm{BAD}] < 2^{-128}$ (negligible)

(otherwise, repeat with multiple random $r_1, \dots, r_t \in \mathbb{F} \setminus \mathcal{L}$)

$\Rightarrow$ Only one $f \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ satisfies $f(r) = s$, with high probability

Poly-IOP $\Rightarrow$ IOP: second attempt

P $\qquad$ V

$$\xrightarrow{f \in \mathbb{F}^{<d}[X]}$$

$V(X) \coloneqq (X - a_1)(X - r)$

$I(a_1) \coloneqq b_1, \quad I(r) = s$

$$\xleftarrow{\text{eval } f \text{ at } a_1 \in \mathbb{F} \setminus \mathcal{L}}$$

$$\xrightarrow[b_1 \leftarrow f(a_1)]{}$$

P $\qquad$ V

$$\xrightarrow[\text{Honest prover: } \pi \coloneqq \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]]{\pi = \square\square\square\square\square\square\square\square\square\square : \mathcal{L} \rightarrow \mathbb{F}}$$

$$\xleftarrow{\text{sample random } r \in \mathbb{F} \setminus \mathcal{L}}$$

$$\xrightarrow[\text{return } s \in \mathbb{F}. \qquad \text{Honest prover: } s \coloneqq f(r)]{}$$

$$\xleftrightarrow[\delta\text{-RS-IOPP for } q_1(X) \coloneqq \frac{\pi(X) - I(X)}{V(X)} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-2]]{}$$

Verifier can conclude: there is $\bar{f}_1 \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ s.t. $\begin{cases} f_1(a_1) = b_1 \\ f_1(r) = s \end{cases}$

Poly-IOP $\Rightarrow$ IOP: second attempt

P $\xrightarrow{f \in \mathbb{F}^{<d}[X]}$ V

$V(X) := (X - a_2)(X - r)$

$I(a_2) := b_2, \quad I(r) = s$

$\xleftarrow{\text{eval } f \text{ at } a_2 \in \mathbb{F} \setminus \mathcal{L}}$

$\xrightarrow[b_2 \leftarrow f(a_2)]{}$

P $\xrightarrow[\text{Honest prover: } \pi := \bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]]{\pi = \square\square\square\square\square\square\square\square\square\square : \mathcal{L} \to \mathbb{F}}$ V

$\xleftarrow{\text{sample random } r \in \mathbb{F} \setminus \mathcal{L}}$

$\xrightarrow[\text{return } s \in \mathbb{F}. \quad \text{Honest prover: } s := f(r)]{}$

$\longleftrightarrow$

δ-RS-IOPP for $q_2(X) := \dfrac{\pi(X) - I(X)}{V(X)} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d-2]$

Verifier can conclude: there is $\bar{f}_2 \in \mathrm{List}[\boldsymbol{\pi}, d, \delta]$ s.t. $f_2(a_2) = b_2, \ f_2(r) = s$

Now: $\delta < 1 - \sqrt{\rho}$ and $f_1(r) = f_2(r) = s \quad \Rightarrow \quad f_1 = f_2$ w.h.p, as required

Poly-IOP $\Rightarrow$ IOP: summary

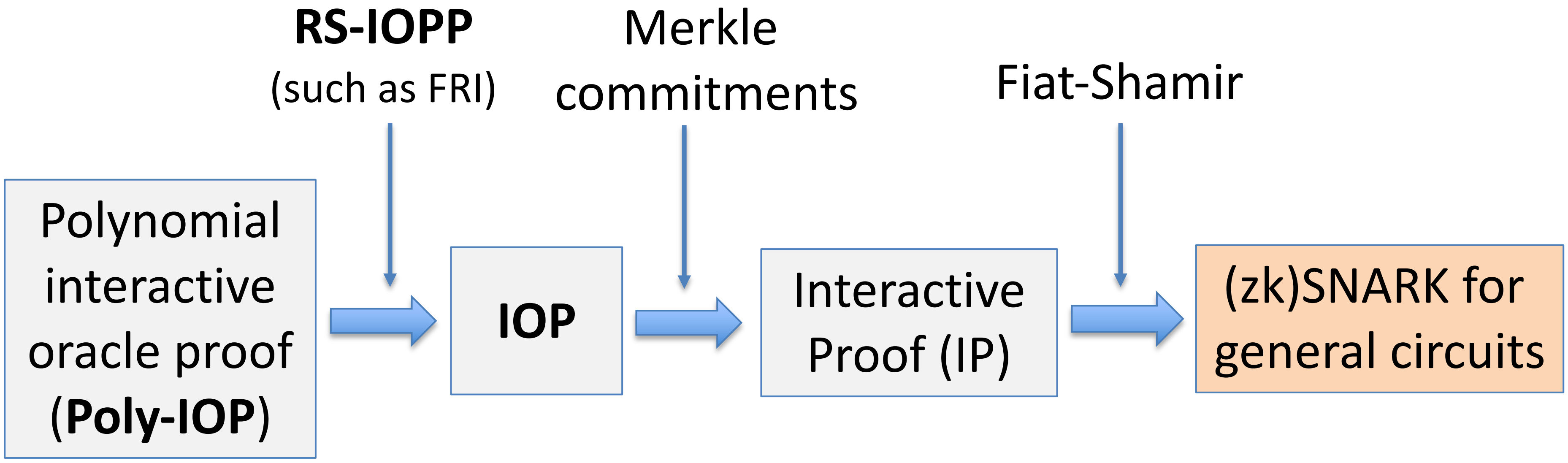
- The IOP prover encodes $f \in \mathbb{F}^{<d}[X]$ using a linear code (RS)
  (other linear codes can be used, possibly with a faster encoding than RS)

- δ-RS-IOPP applied to a quotient $q(X) := \frac{\pi(X)-I(X)}{V(X)}$
  proves evaluations of the encoded polynomial to the Verifier.

- For $\delta < 1-\sqrt{\rho}$: an out of domain query (r, s) ensures that the prover is bound to a unique polynomial, w.h.p

Poly-IOP ⇒ IOP ⇒ SNARK

A direct SNARK construction:



An RS-IOPP is the key ingredient in compilation

Poly-IOP $\Rightarrow$ IOP: remarks

Remark 1: what if Poly-IOP Verifier wants to query f at $a \in \mathcal{L}$??

- The problem: $q(X) \coloneqq \frac{\pi(X)-I(X)}{(X-a)(X-r)} : \mathcal{L} \to \mathbb{F}$

  is undefined at $X = a$ (not a problem when $a \notin \mathcal{L}$)

- Solution: $Q(X) \coloneqq \big(f(X) - I(X)\big)/(X-a)(X-r)$ is a poly. in $\mathbb{F}^{<d-2}[X]$.

  Honest prover defines $q(a) \coloneqq Q(a)$ and runs the RS-IOPP on q.

Remark 2: naively, the IOP uses one RS-IOPP per query to f

- In practice, we can batch many RS-IOPPs into one RS-IOPP
- Let's see how … first we need some tools

One last topic before the break:

Distance Preserving Transformations

Towards an efficient RS-IOPP

Distance Preserving Transformations

Let $\mathcal{L}, \mathcal{L}' \subseteq \mathbb{F}$, d, d' some degree bounds, and $\delta \in [0,1]$.

<u>Def</u>: A **distance preserving transformation** is a randomized map

$$T(u_1, \dots, u_k; r) \to u$$

that maps $u_1, \dots, u_k \colon \mathcal{L} \to \mathbb{F}$ to $u \colon \mathcal{L}' \to \mathbb{F}$ such that:

case 1: (the honest case)

if $u_1, \dots, u_k \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ then $u \in \mathrm{RS}[\mathbb{F}, \mathcal{L}', d']$ for all r.

case 2: (the dishonest case)

if some u_j is δ-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ then
u is δ-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}', d']$, w.h.p over r.

Example 1: batch RS-IOPP

Setting: Prover has $u_0, \dots, u_k : \mathcal{L} \to \mathbb{F}$, Verifier has oracles for $u_0, \dots, u_k$.

Goal: convince Verifier that all $u_0, \dots, u_k$ are δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$.

- **Naively**: run k RS-IOPP protocols $\Rightarrow$ expensive
- **Better**: batch all k into a single function $u : \mathcal{L} \to \mathbb{F}$

step 1: Verifier samples random r in $\mathbb{F}$; sends to prover

step 2: Prover sets $u := u_0 + r \cdot u_1 + r^2 u_2 + \cdots + r^k u_k : \mathcal{L} \to \mathbb{F}$

step 3: Both run RS-IOPP on $u : \mathcal{L} \to \mathbb{F}$

when Verifier wants u(a) for some $a \in \mathcal{L}$, prover opens all $u_0(a), \dots, u_k(a)$

Why is this distance preserving?

Case 1: (an honest prover)

if $u_0, \dots, u_k \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ then $u \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ for <u>all</u> $r \in \mathbb{F}$

Case 2: (a dishonest prover)

if some u_j is δ-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$, we need to argue that u is δ-far from $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$, with high probability over $r \in \mathbb{F}$

When $\delta \in [0, 1 - \sqrt{\rho})$, Case 2 follows from the celebrated <u>BCIKS</u> proximity gap theorem.

The proximity gap theorem

Thm (BCIKS'20, Thm. 6.2): $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$ an RS-code with const. rate $\rho := d/n$ (say, $\rho = 0.5$)

$n := |\mathcal{L}|$

Let $u_0, \dots, u_k : \mathcal{L} \to \mathbb{F}$ and $0 < \delta < 1 - 1.01\sqrt{\rho}$.

For $r \in \mathbb{F}$ define $u^{(r)} := u_0 + r \cdot u_1 + r^2 u_2 + \cdots + r^k u_k$.

Suppose that $\Pr_r[\ u^{(r)}$ is δ-close to $\mathrm{RS}[\mathbb{F},\mathcal{L},d]\ ] > err$

then all u_j are δ-close to $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$,

where $$\begin{cases} err = O\left(\frac{kn}{|\mathbb{F}|}\right) & \text{for } 0 < \delta < \frac{1-\rho}{2} \\ err = O\left(\frac{kn^2}{|\mathbb{F}|}\right) & \text{for } \frac{1-\rho}{2} < \delta < 1 - 1.01\sqrt{\rho} \end{cases}$$

We will assume that err is negligible, i.e. err $< 1/2^{128}$

(if not, use multiple r)

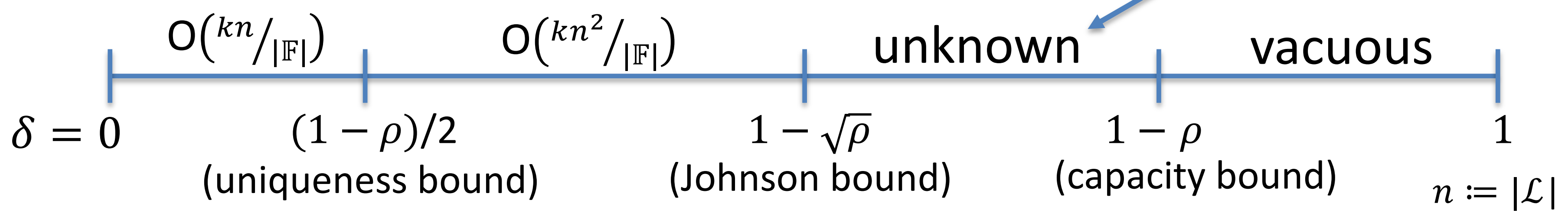
The proximity gap theorem

Suppose that $\Pr_r[\ u^{(r)}$ is δ-close to $\mathrm{RS}[\mathbb{F},\mathcal{L},d]\ ] > err$

then all u_j are δ-close to $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$

Contra-positive: if some u_j is δ-far from $\mathrm{RS}[\mathbb{F},\mathcal{L},d]$
then $u^{(r)}$ is δ-far with high probability, over r.

Proximity gap error (err) as a function of $\delta \in [0,1]$:



A stronger form: correlated proximity

Thm (BCIKS'20, Thm. 6.2):

Let $u_0, \ldots, u_k \colon \mathcal{L} \to \mathbb{F}$ and $0 < \delta < 1 - 1.01\sqrt{\rho}$.

Suppose that $\Pr_r\big[\, u^{(r)}$ is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]\,\big] > err$

then there is an $S \subseteq \mathcal{L}$ such that $|S| \geq (1-\delta) \cdot |\mathcal{L}|$ and

for all j: $\exists f_j \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ s.t. $\forall x \in S$: $u_j(x) = f_j(x)$

$\Rightarrow$ $u_0, \ldots, u_k$ are δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ on the same positions S .

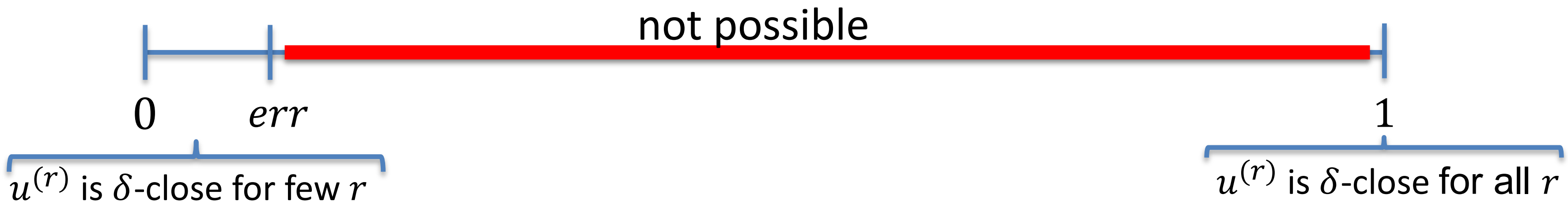
(recall $u^{(r)} := u_0 + r \cdot u_1 + r^2 u_2 + \cdots + r^k u_k$)

Why is this called a proximity gap??

Suppose that $\Pr_r\big[\, u^{(r)}$ is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d] \,\big] > err$ then all u_j are δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ on the same positions $S \subseteq \mathcal{L}$

But if all $u_0, \dots, u_k : \mathcal{L} \to \mathbb{F}$ are δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ on positions $S \subseteq \mathcal{L}$, then $u^{(r)}$ is δ-close <u>for all</u> $r \in \mathbb{F}$.

So $\Pr_r\big[\, u^{(r)}$ is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d] \,\big]$ exhibits a gap:


not possible
0
err
1
$u^{(r)}$ is δ-close for few r
$u^{(r)}$ is δ-close for all r

Proximity gaps for other linear codes?

A similar proximity gap holds for every linear code.

min. distance

Thm: (Zeilberger'24) Let $\mathcal{C} \subseteq \mathbb{F}^n$ be an $[n, dim, \underset{\uparrow}{l}]_p$ linear code.
Then $\mathcal{C}$ has a correlated proximity gap for $0 < \delta < 1 - \sqrt[4]{\tau}$
and err = O$\left({}^{kn}/_{|\mathbb{F}|}\right)$, where $\tau := 1 - (l/n)$.

(For RS-code $\tau \approx \rho$, so this gap is much weaker than BCIKS'20)

This can be used in a $\mathcal{C}$-proximity IOPP (e.g., Basefold, Blaze)

2nd Distance preserving example: 2-way folding

From now on set $\mathcal{L} = \{1, \omega, \omega^2, \ldots, \omega^{n-1}\} \subseteq \mathbb{F}$, where

- n is a power of two, and
- ω is an n-th primitive root of unity $(\omega^n = 1)$

  (requires that n divides $|\mathbb{F}| - 1$)

Then:

- $\omega^{n/2} = -1$ so that if $x = \omega^i \in \mathcal{L}$ then $-x = \omega^{i+(n/2)} \in \mathcal{L}$
- $|\mathcal{L}^2| = |\{a^2 : a \in \mathcal{L}\}| = |\mathcal{L}|/2 = n/2$ $\quad (-a,\ a \to a^2)$

2-way folding a polynomial

A folding transformation: let's start with an example.

Let $f(X) = 1 + 2X + 3X^2 + 4X^3 + 5X^4 + 6X^5 \in \mathbb{F}^{<6}[X]$

Define $f_{\text{even}}(X) \coloneqq 1 + 3X + 5X^2$ and $f_{\text{odd}}(X) \coloneqq 2 + 4X + 6X^2$

Then: $f(X) = f_{\text{even}}(X^2) + X \cdot f_{\text{odd}}(X^2)$

Define: for $r \in \mathbb{F}$ define $f_{\text{fold},r} \coloneqq f_{\text{even}} + r \cdot f_{\text{odd}} \in \mathbb{F}^{<3}[X]$

2-way folding a polynomial: more generally

For $f \in \mathbb{F}^{<d}[X]$ (with d even) define:

- $f_{\text{even}}(X^2) := \frac{f(X)+f(-X)}{2}$ and $f_{\text{odd}}(X^2) := \frac{f(X)-f(-X)}{2X}$
- $f_{\text{fold},r}(X) := f_{\text{even}}(X) + r \cdot f_{\text{odd}}(X) \in \mathbb{F}^{<d/2}[X]$

Then: $f(X) = f_{\text{even}}(X^2) + \mathrm{X} \cdot f_{\text{odd}}(\mathrm{X}^2)$

- for every $a \in \mathbb{F}$: $f_{\text{fold},r}(a^2)$ can be eval given $f(a), f(-a)$
- $\bar{f} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d] \Rightarrow \overline{f_{\text{fold},r}} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2]$ $\xleftarrow[\text{rate = } d/|\mathcal{L}|]{\text{unchanged}}$

Folding an arbitrary word $u\colon \mathcal{L} \to \mathbb{F}$

For $u\colon \mathcal{L} \to \mathbb{F}$ and $r \in \mathbb{F}$ define $u_e, u_o, u_{\text{fold},r}\colon \mathcal{L}^2 \to \mathbb{F}$ as

- for $a \in \mathcal{L}$: $u_e(a^2) := \frac{u(a)+u(-a)}{2}$ and $u_o(a^2) := \frac{u(a)-u(-a)}{2a}$
- for $b \in \mathcal{L}^2$: $u_{\text{fold},r}(b) := u_e(b) + r \cdot u_o(b)$

(recall $|\mathcal{L}^2| = |\mathcal{L}|/2$)

Lemma (distance preservation): for $0 < \delta < 1 - \sqrt{\rho}$

- $u \in \text{RS}[\mathbb{F}, \mathcal{L}, d] \Rightarrow u_{\text{fold},r} \in \text{RS}[\mathbb{F}, \mathcal{L}^2, d/2]$ for all $r \in \mathbb{F}$
- u is δ-far from $\text{RS}[\mathbb{F}, \mathcal{L}, d] \Rightarrow$

$$\Pr_r[\, u_{\text{fold},r} \text{ is } \delta\text{-far from } \text{RS}[\mathbb{F}, \mathcal{L}^2, d/2]\,] \geq 1 - err$$

Folding an arbitrary word $u: \mathcal{L} \to \mathbb{F}$

For $u: \mathcal{L} \to \mathbb{F}$ and $r \in \mathbb{F}$ define $u_e, u_o, u_{\text{fold},r}: \mathcal{L}^2 \to \mathbb{F}$ as

- for $a \in \mathcal{L}$: $u_e(a^2) := \frac{u(a)+u(-a)}{2}$ and $u_o(a^2) := \frac{u(a)-u(-a)}{2a}$
- for $b \in \mathcal{L}^2$: $u_{\text{fold},r}(b) := u_e(b) + r \cdot u_o\,(b)$

(recall $|\mathcal{L}^2| = |\mathcal{L}|/2$)

Lemma (distance preservation): for $0 < \delta < 1 - \sqrt{\rho}$

- $u \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d] \Rightarrow u_{\text{fold},r} \in \mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2]$ for all $r \in \mathbb{F}$
- $\Pr_r[\, u_{\text{fold},r}$ is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2]\,] > err \Rightarrow$

  u is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$

(contra-positive)

Why is this true?

The first part of the lemma is easy. Let's prove the second part.

- Suppose that $\Pr_r\big[\, u_{\mathrm{fold},r}$ is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2] \,\big] > err$
- Then by the BCIKS'20 theorem, there are $g_e, g_o \in \mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2]$ that match u_e, u_o on a set $S \subseteq \mathcal{L}^2$ of size $|S| \geq (1-\delta)(n/2)$
- Define $g \colon \mathcal{L} \to \mathbb{F}$ as $g(a) \coloneqq g_e(a^2) + a \cdot g_o(a^2) \in \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$
- Then: $g(a) = u(a)$ for all $a \in \mathcal{L}$ for which $a^2 \in S$ (2$|S|$ values in $\mathcal{L}$)
- But then $\Delta(u, g) \leq 1 - \frac{2|S|}{n} = 1 - \frac{|S|}{n/2} \leq \delta.$

  $\Rightarrow$ u is δ-close to $\mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$

An important corollary

Let $\mathcal{C} = \mathrm{RS}[\mathbb{F}, \mathcal{L}, d]$ and $\mathcal{C}' = \mathrm{RS}[\mathbb{F}, \mathcal{L}^2, d/2]$

Corollary: For $u: \mathcal{L} \to \mathbb{F}$ (folding does not decrease distance, w.h.p)

- if $\Delta(u, \mathcal{C}) < 1 - \sqrt{\rho}$ then $\Pr_r\left[\, \Delta\left(u_{\mathrm{fold},r}, \mathcal{C}'\right) \geq \Delta(u, \mathcal{C}) \,\right] \geq 1 - err$
- if $\Delta(u, \mathcal{C}) \geq 1 - \sqrt{\rho}$ then $\Pr_r\left[\, \Delta\left(u_{\mathrm{fold},r}, \mathcal{C}'\right) \geq 1 - \sqrt{\rho} \,\right] \geq 1 - err$

Recall: $\Delta(u, \mathcal{C}) \leq \delta \iff u$ is δ-close to $\mathcal{C}$

4-way folding $u\colon \mathcal{L} \to \mathbb{F}$ (using $i^2 = -1$)

For $u\colon \mathcal{L} \to \mathbb{F}$ define $u_0, u_1, u_2, u_3 : \mathcal{L}^4 \to \mathbb{F}$ for $a \in \mathcal{L}$ as

$$\begin{pmatrix} 4 \cdot u_0(a^4) \\ 4a \cdot u_1(a^4) \\ 4a^2 \cdot u_2(a^4) \\ 4a^3 \cdot u_3(a^4) \end{pmatrix} := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & -1 & 1 & -1 \\ 1 & i & i^2 & i^3 \end{pmatrix} \cdot \begin{pmatrix} u(a) \\ u(ia) \\ u(i^2 a) \\ u(i^3 a) \end{pmatrix}$$

(a degree-4 FFT)

The 4-way fold of *u*: *for* $r \in \mathbb{F}$ define $u_{4\text{fold},r}\colon \mathcal{L}^4 \to \mathbb{F}$ as

$$u_{4\text{fold},r}(b) := u_0(b) + r \cdot u_1(b) + r^2 \cdot u_2(b) + r^3 \cdot u_3(b) \quad \text{for } b \in \mathcal{L}^4$$

Evaluating $u_{4\text{fold},r}(X)$ at $b \in \mathcal{L}^4$ requires <u>four</u> evals. of $u(X)$.

4-way folding $u\colon \mathcal{L} \to \mathbb{F}$ (using $i^2 = -1$)

For $u\colon \mathcal{L} \to \mathbb{F}$ define $u_0, u_1, u_2, u_3 : \mathcal{L}^4 \to \mathbb{F}$ for $a \in \mathcal{L}$ as

$$\begin{pmatrix} 4 \cdot u_0(a^4) \\ 4a \cdot u_1(a^4) \\ 4a^2 \cdot u_2(a^4) \\ 4a^3 \cdot u_3(a^4) \end{pmatrix} := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & (-i)^2 & (-i)^3 \\ 1 & -1 & 1 & -1 \\ 1 & i & i^2 & i^3 \end{pmatrix} \cdot \begin{pmatrix} u(a) \\ u(ia) \\ u(i^2 a) \\ u(i^3 a) \end{pmatrix}$$

(a degree-4 FFT)

The 4-way fold of *u*: *for* $r \in \mathbb{F}$ define $u_{4\mathrm{fold},r}\colon \mathcal{L}^4 \to \mathbb{F}$ as

$$u_{4\mathrm{fold},r}(b) := u_0(b) + r \cdot u_1(b) + r^2 \cdot u_2(b) + r^3 \cdot u_3(b) \quad \text{for } b \in \mathcal{L}^4$$

<u>**Fact**</u>: the same distance preservation corollary holds for $u_{4\mathrm{fold},r}$

8-way folding $u: \mathcal{L} \to \mathbb{F}$ (using an 8th root of unity)

Can similarly define 8-way folding, or even 2^w folding for $w \geq 3$.

maps $u: \mathcal{L} \to \mathbb{F}$ to $u_{2^w\text{fold},r}: \mathcal{L}^{2^w} \to \mathbb{F}$ ($|\mathcal{L}^{2^w}| = |\mathcal{L}|/2^w$)

(1) evaluating $u_{2^w\text{fold},r}(b)$ requires 2^w evals. of $u(X)$

$\Rightarrow$ uses a degree-2^w FFT (degree-8 FFT for 8-way folding)

(2) the same distance preservation corollary holds for $u_{2^w\text{fold},r}$

End of Segment 1: Brief Summary

For a linear code $\mathcal{C}$: $\text{List}[u, \mathcal{C}, \delta]$ is small up to $\delta < 1 - \sqrt{1-\mu}$

Poly-IOP → IOP compiler:

- Honest $\boldsymbol{P}$ Commits to $f \in \mathbb{F}^{<d}[X]$ by sending its encoding $\bar{f}$ to $\boldsymbol{V}$
- Prove evaluation of f using RS-IOPP on quotient of sent word u
- Out-of-domain eval. commits P to unique word in $\text{List}[u, \mathcal{C}, \delta]$

Folding:

- $(u: \mathcal{L} \rightarrow \mathbb{F}) \quad \rightarrow \quad (u_{\text{fold},r}: \mathcal{L}^2 \rightarrow \mathbb{F})$ is a distance preserving map
- Proof using the BCIKS'20 proximity gap theorem

Let’s put all this machinery to use

See you in part 2 …

END OF SEGMENT