STIR 🥣

Reed-Solomon Proximity Testing with Fewer Queries

Gal Arnon
Weizmann Institute of Science

Giacomo Fenzi
EPFL

Alessandro Chiesa
EPFL

Eylon Yogev

Motivation

STARKs & Friends

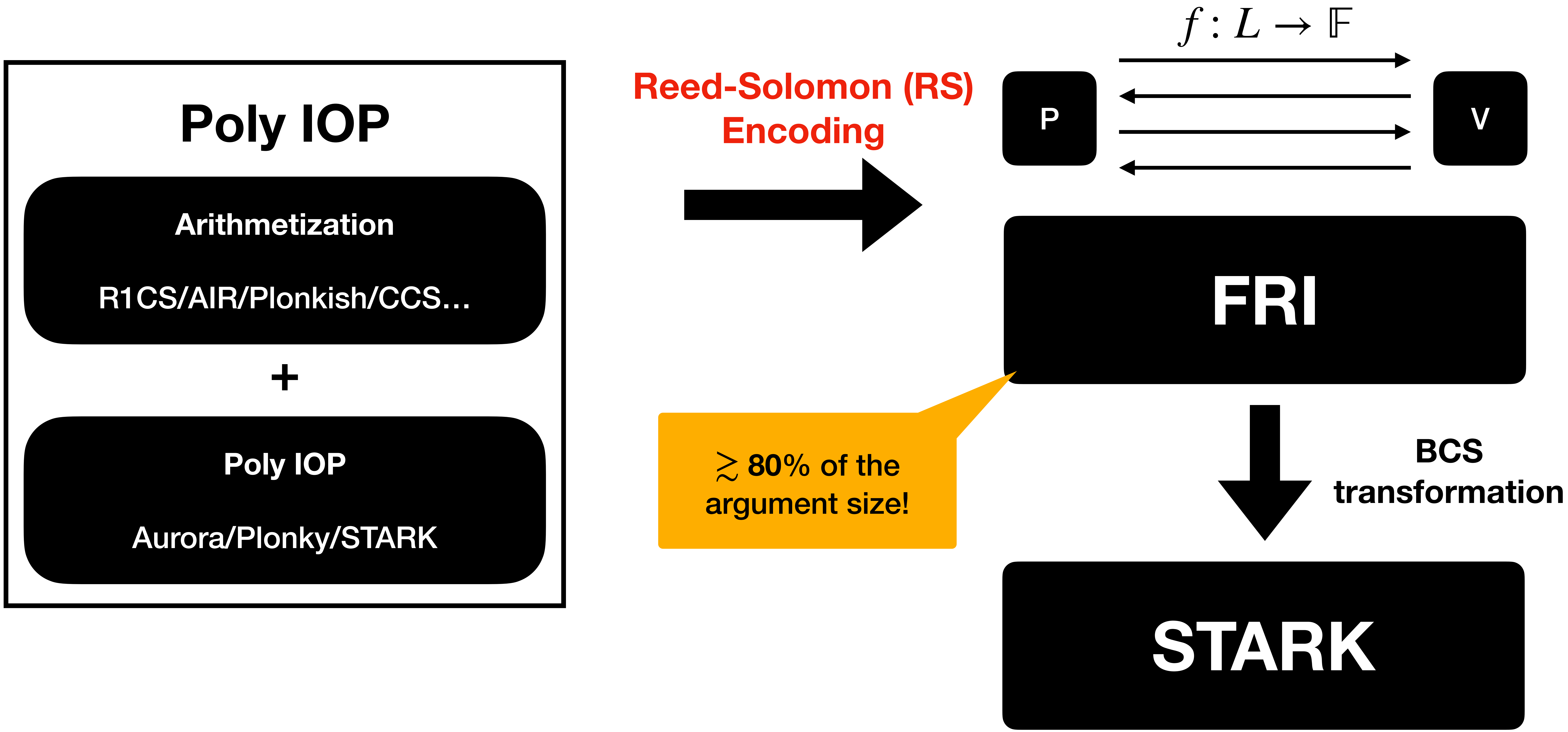
- **IOP**-based SNARKs instantiated using the BCS transformation
- Secure in the ROM: can be instantiated with only hash-functions
- Transparent (public-coin) setup
- Fast proving, not tied to ECC fields, post quantum secure in QROM [CMS].

Rollups: STARKWARE polygon zkSync

zkVMs: Succinct RISC ZERO **And more…**

Anatomy of an IOP-based SNARK

Reducing to low-degree testing



Background

RS Codes and IOPPs

$\mathsf{RS}[\mathbb{F}, L, d]$

Field | Evaluation Domain | Degree

$\hat{p} \in \mathbb{F}^{<d}[X]$

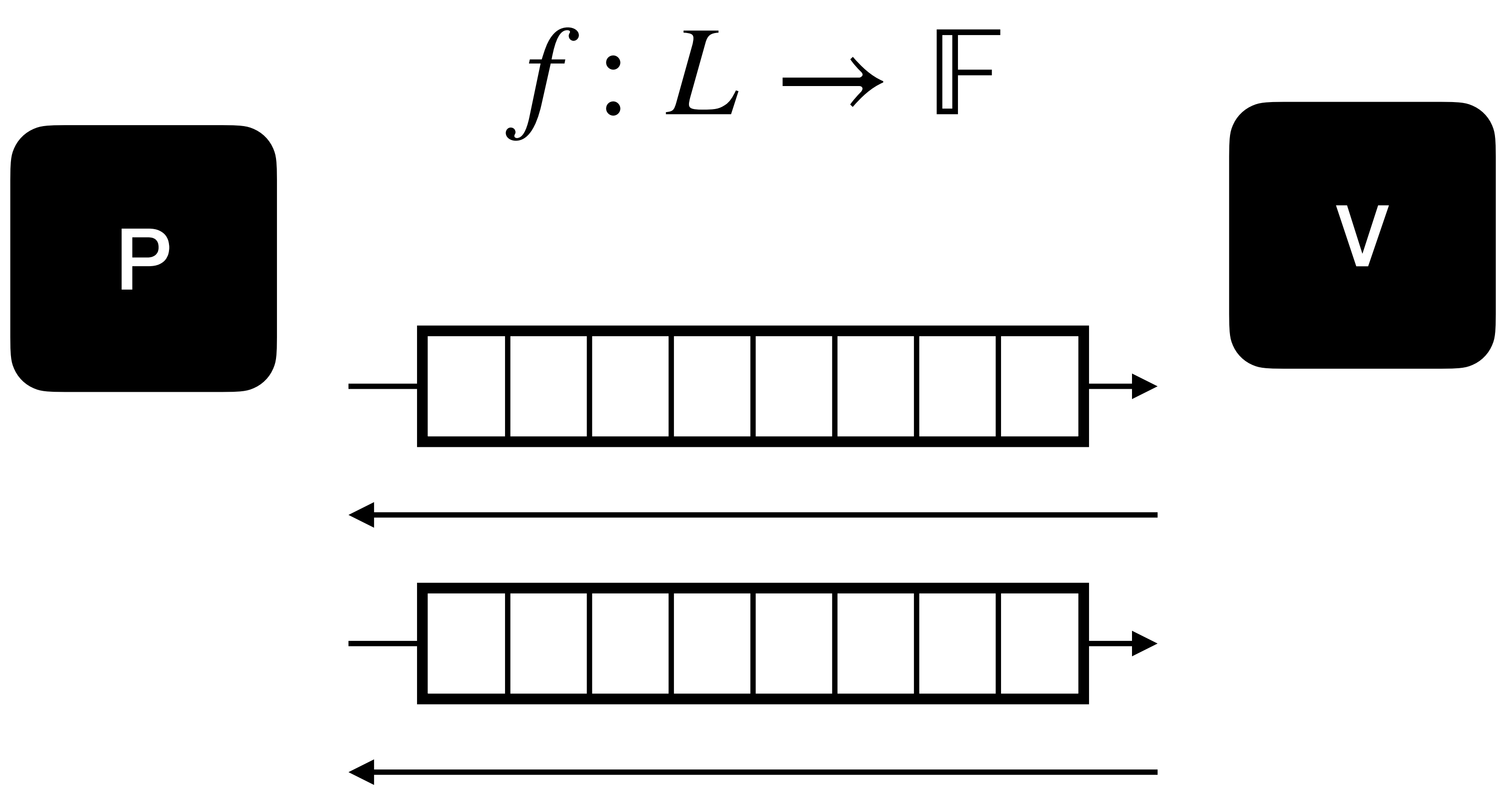
Enc

$f : L \to \mathbb{F}$ with $\hat{p}|_L \equiv f$

Rate: $\rho = d/|L|$, think $\rho = 1/4$

L smooth $\implies$ Enc is an FFT

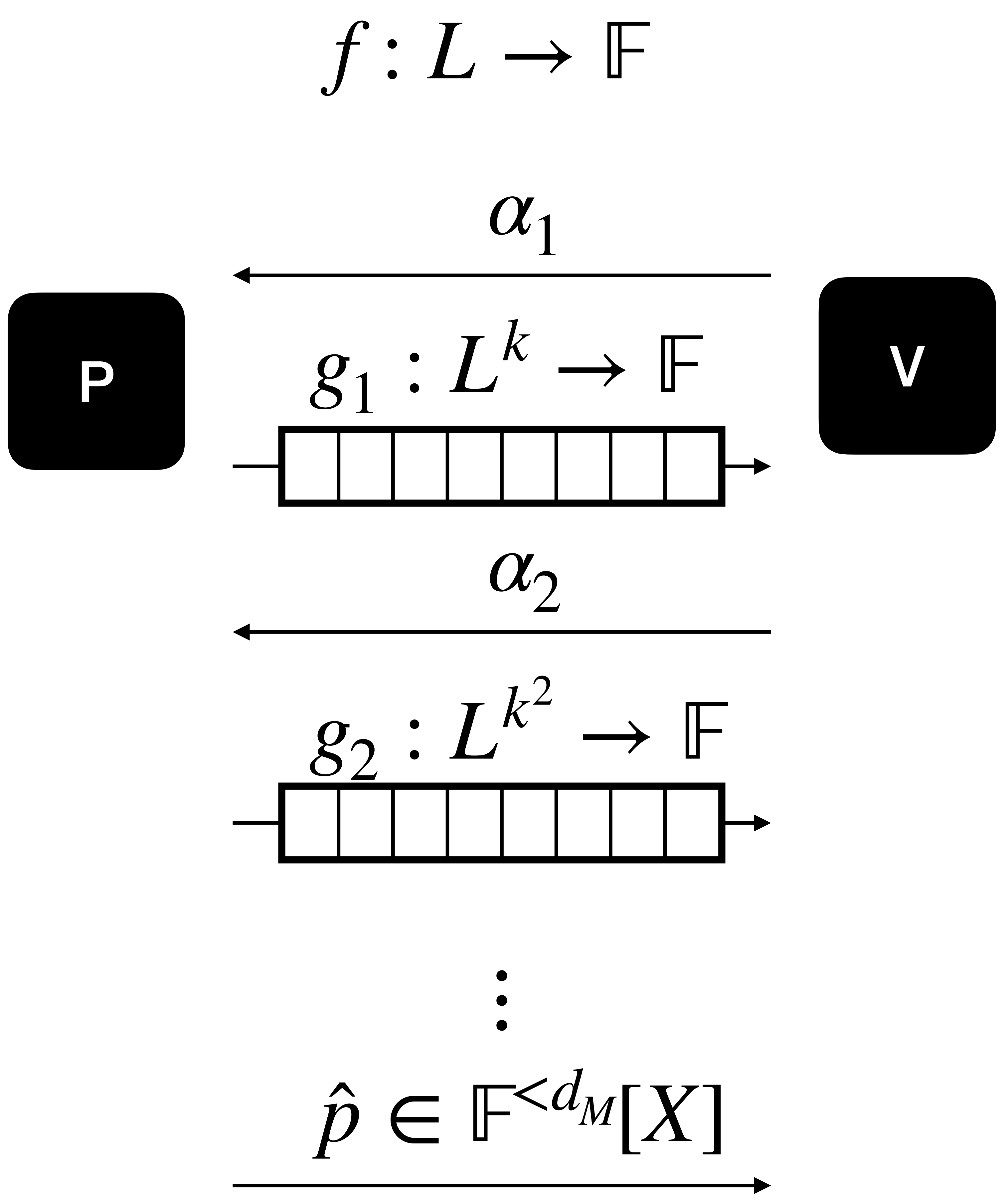
IOPP for RS



- $f \in \mathsf{RS}[\mathbb{F}, L, d] \implies \mathbf{V}$ accepts
- f is δ-far from $\mathsf{RS}[\mathbb{F}, L, d] \implies$ $\mathbf{V}$ rejects w.h.p.

$\mathbf{V}$ makes “few queries” to f and proof oracles

FRI Protocol



In this talk: $L^k = \{x^k : x \in L\}$

g_1 is supposed to be the folding of f around α_1

Recursively test the claim that $g_i \in \mathrm{RS}[\mathbb{F}, L^{k^i}, d/k^i]$

Smooth structure allows to check consistency in a single query-phase at the end

Rounds: $O(\log d)$

Proof length: $O(|L|)$

Queries: $O\left(\lambda \cdot \frac{\log d}{-\log \sqrt{\rho}}\right)$ for $\delta = 1 - \sqrt{\rho}$

(To get λ-bits of security, **without conjecture**)

round-by-round

Our results

STIR

An IOPP for RS with

Rounds: $O(\log d)$

Proof length: $O(|L|)$

Queries:

$$O\left(\lambda \cdot \log\left(\frac{\log d}{-\log\sqrt{\rho}}\right) + \log d\right) \quad \text{for } \delta = 1 - \sqrt{\rho}$$

(To get λ-bits of security, **without conjecture**)

round-by-round

Comparison to FRI

FRI: $O\left(\lambda \cdot \frac{\log d}{-\log \sqrt{\rho}}\right)$

STIR: $O\left(\lambda \cdot \log\left(\frac{\log d}{-\log \sqrt{\rho}}\right) + \log d\right)$

- **Drop-in** replacement of FRI
- **Fewer** queries leads to:
  - Fewer authentication paths $\Longrightarrow$ **smaller** argument size
  - Smaller verifier hash-complexity $\Longrightarrow$ **faster** and more **recursion-friendly**!
- Rough query comparison:
  - Example parameters: $\rho = 1/4, \delta = 1 - \sqrt{\rho}$, targeting 100-bits of security
  - FRI: ~ **400** queries vs STIR: ~ **200** queries

can increase by PoW

Implementation

- Rust 🦀 implementation, available at WizardOfMenlo/stir
- Arkworks as backend, 192-bit field for benchmarks, reasonably optimised
- Implemented both FRI and STIR
- Modular over choice of:
  - Field
  - Fiat-Shamir hash
  - Merkle hash
- Decently well written (for academia! 📚)

```rust
pub trait LowDegreeTest<F, MerkleConfig, FSConfig>
where
    F: FftField,
    MerkleConfig: Config,
    FSConfig: CryptographicSponge,
    FSConfig::Config: Clone,
{
    type Prover: Prover<
        F,
        MerkleConfig,
        FSConfig,
        Commitment = <Self::Verifier as Verifier<F, MerkleConfig, FSConfig>>::Commitment,
        Proof = <Self::Verifier as Verifier<F, MerkleConfig, FSConfig>>::Proof,
    >;
    type Verifier: Verifier<F, MerkleConfig, FSConfig>;

    fn instantiate(
        parameters: Parameters<F, MerkleConfig, FSConfig>,
    ) -> (Self::Prover, Self::Verifier) {
        let prover = Self::Prover::new(parameters.clone());
        let verifier = Self::Verifier::new(parameters);

        (prover, verifier)
    }
}
```

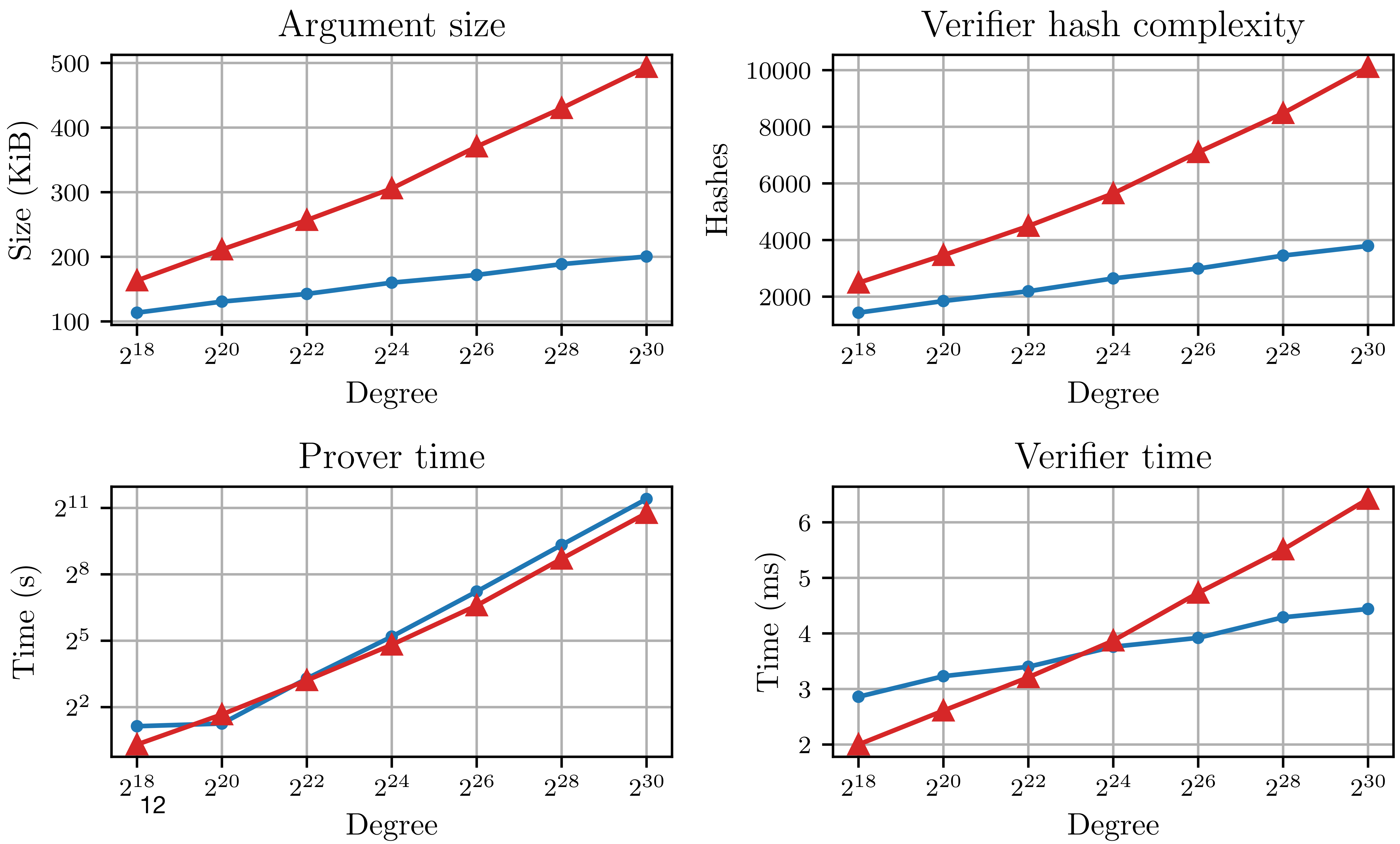
Results

- Compared to FRI: better **argument size** and **verifier hash complexity** across **every** degree-rate pair benchmarked!
- Larger improvements when degree and rate increase

$d = 2^{24}, \rho = 1/4$	**FRI**	**STIR**
Size (KiB)	177	107
Hashes	3.5k	1.8k

$d = 2^{30}, \rho = 1/2$	**FRI**	**STIR**
Size (KiB)	494	200
Hashes	10k	3.8k

$\rho = 1/2$



Techniques

Main Idea: Shift-To-Improve-Rate

A smaller rate makes the code easier to test

- A STIR iteration reduces testing proximity to $C = \mathsf{RS}[\mathbb{F}, L, d]$ to testing proximity to $C' = \mathsf{RS}[\mathbb{F}, L', d/k]$ where $|L'| = |L|/2$.

- New rate is $\rho' = (2/k) \cdot \rho$, if $k > 2$ the rate **improves**! (i.e. the new code is easier to test).

- We also **amplify distance**. If f is δ-far then f' is $(1 - \sqrt{\rho'})$-far to the new code unless with probability $(1-\delta)^t$ where t is number of queries.

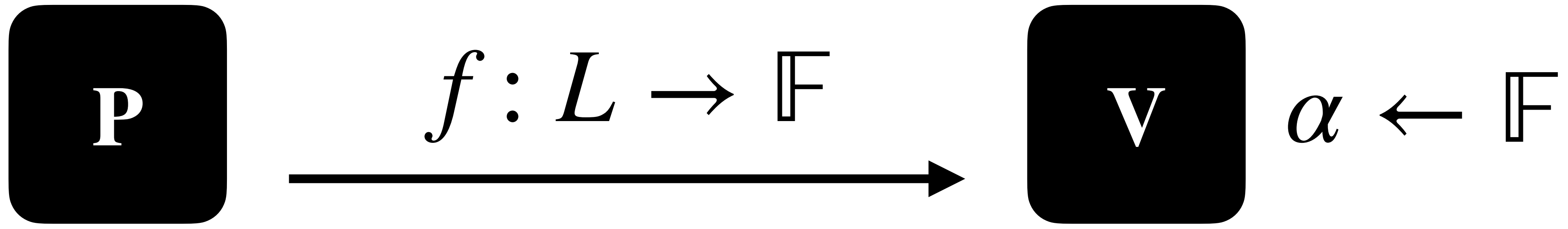
- Round-by-round errors roughly as below so can set:

Less queries each round!

$$(1-\delta)^{t_0}, \rho_1^{\frac{t_1}{2}}, \ \ldots, \rho_M^{\frac{t_M}{2}} \qquad t_0 = \frac{\lambda}{-\log(1-\delta)} \qquad t_i = \frac{\lambda}{-\log(\sqrt{\rho_i})}$$

Folding

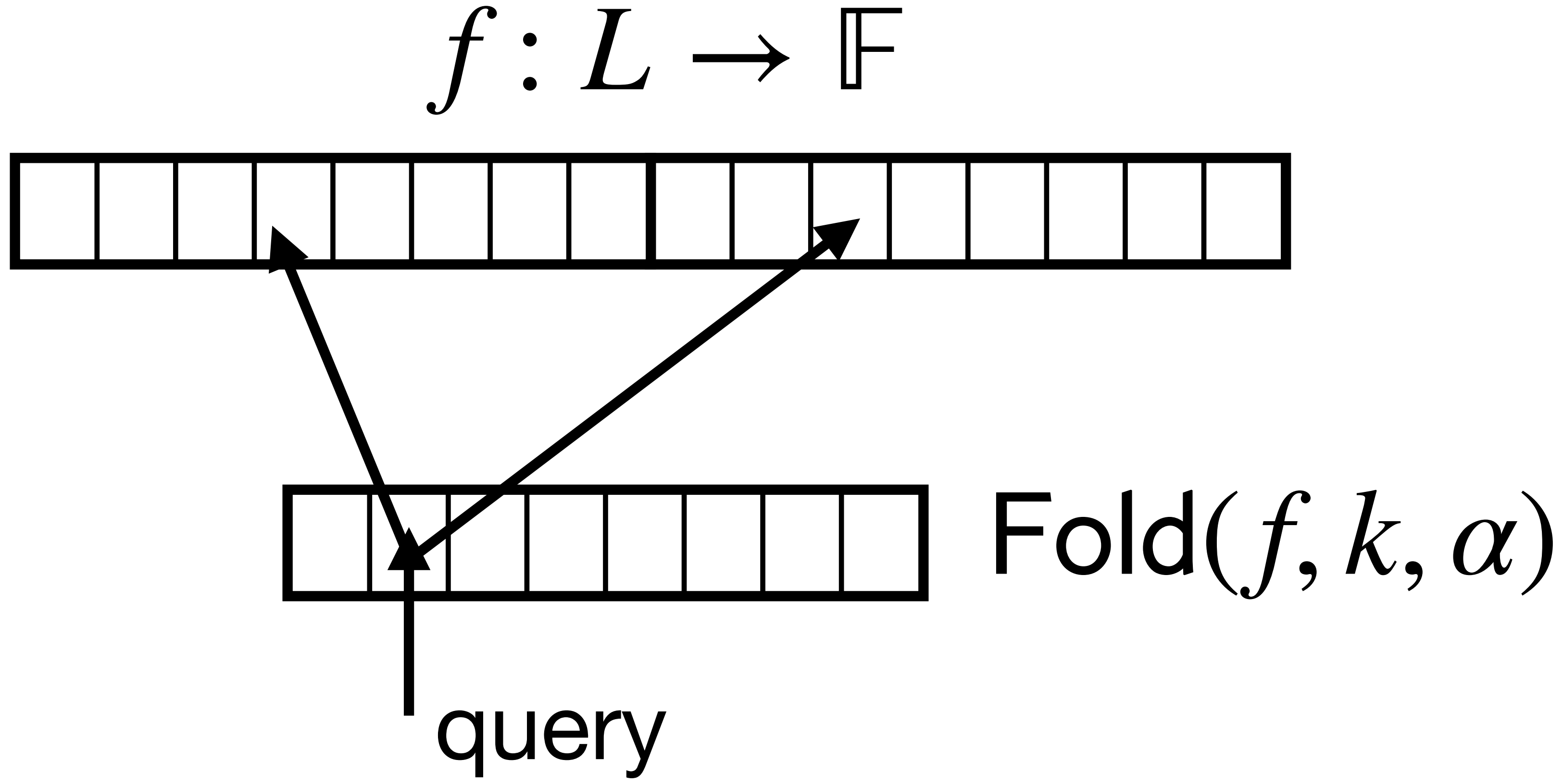
Reduce $\mathsf{RS}[\mathbb{F}, L, d]$ **to** $\mathsf{RS}[\mathbb{F}, L^k, d/k]$



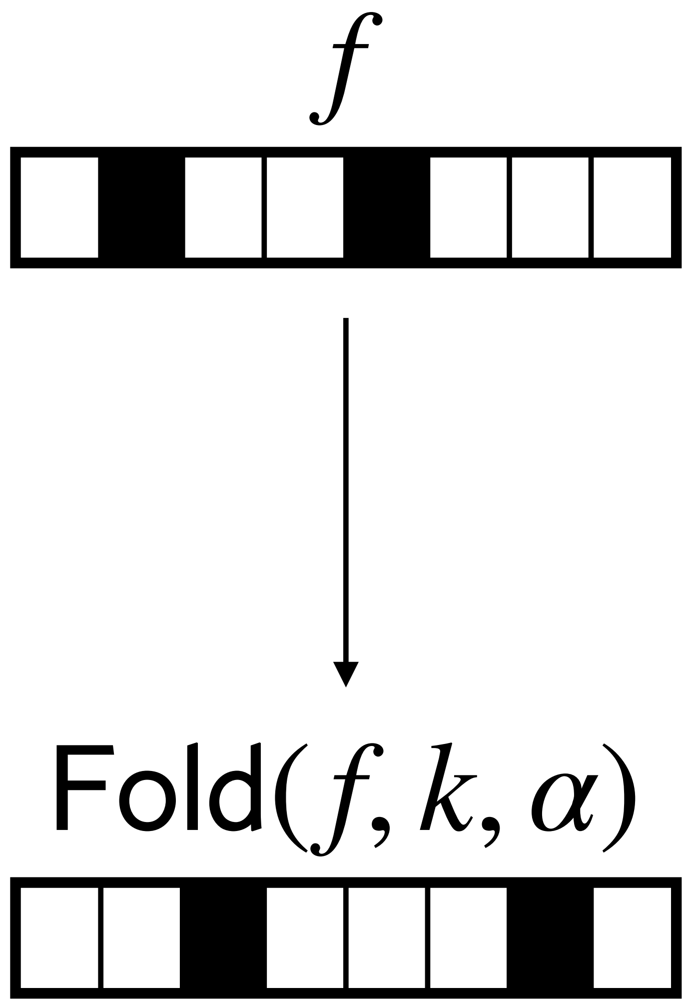
Selecting α defines a function $\mathsf{Fold}(f, k, \alpha)$

Local

By querying f at k locations, $\mathbf{V}$ can compute $\mathsf{Fold}(f, k, \alpha)$ at $z \in L^k$



Distance Preserving



f is δ-far from RS

w.h.p. over α

Fold is δ-far from RS

Quotienting

Enforce constraints on f or amplify distance

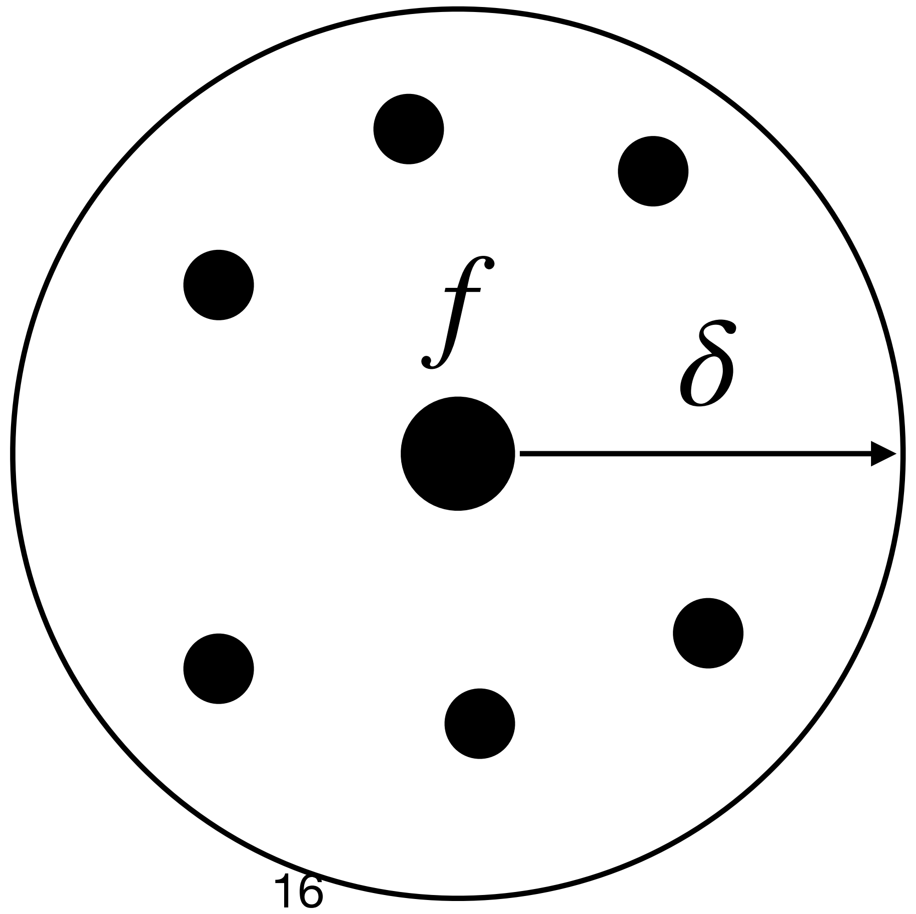
Let $f : L \to \mathbb{F}$ be a function and $\mathsf{Ans} : S \to \mathbb{F}$ be a list of (claimed) evaluations of (the extension of) f on S

$$\mathsf{Quotient}(f, \mathsf{Ans})(x) := \frac{f(x) - \hat{\mathsf{Ans}}(x)}{V_S(x)}$$

Local

$\mathbf{V}$ can compute $\mathsf{Quotient}(f, \mathsf{Ans})$ at $x \in L - S$ by querying f at x

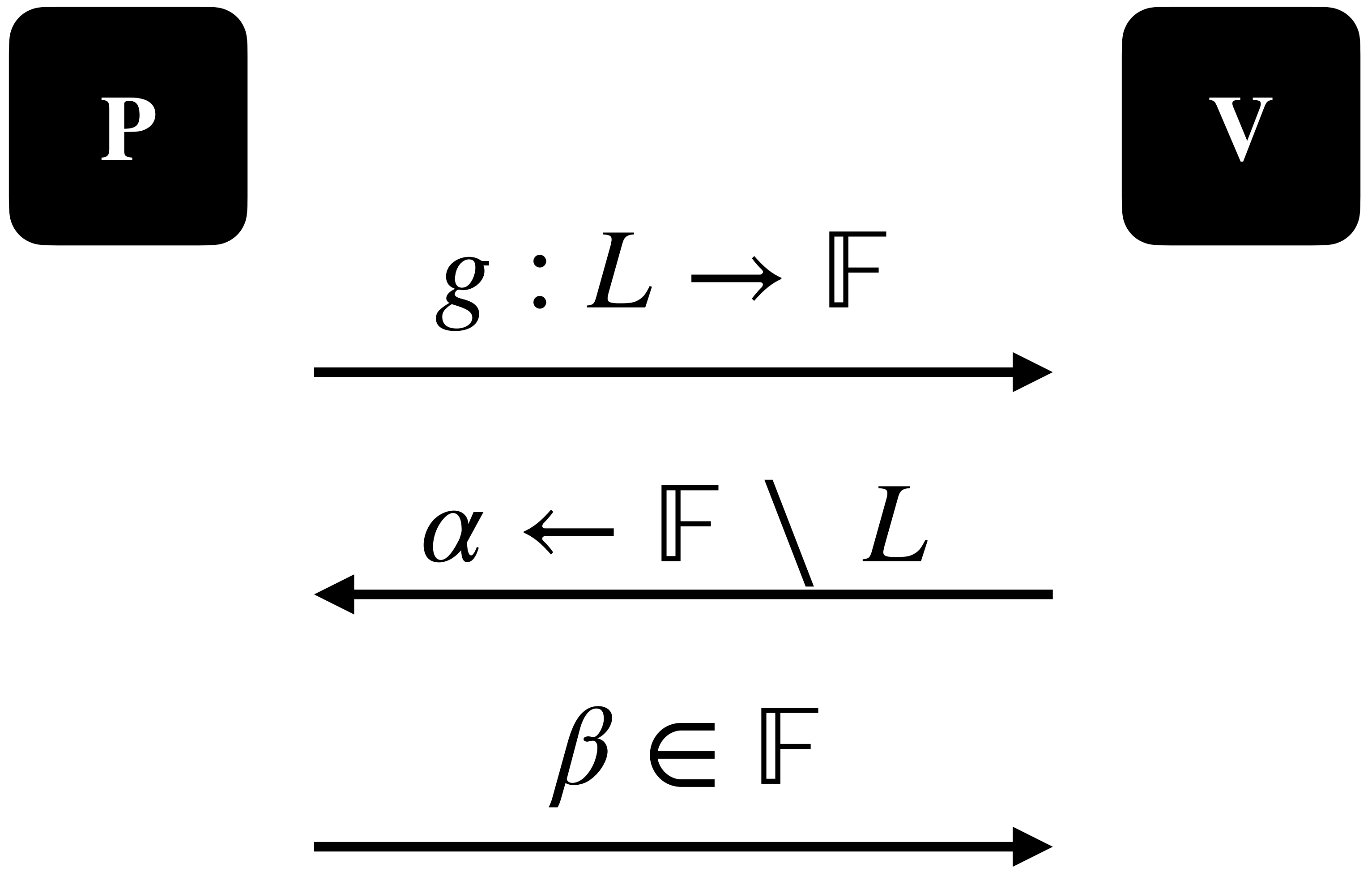
Consistency



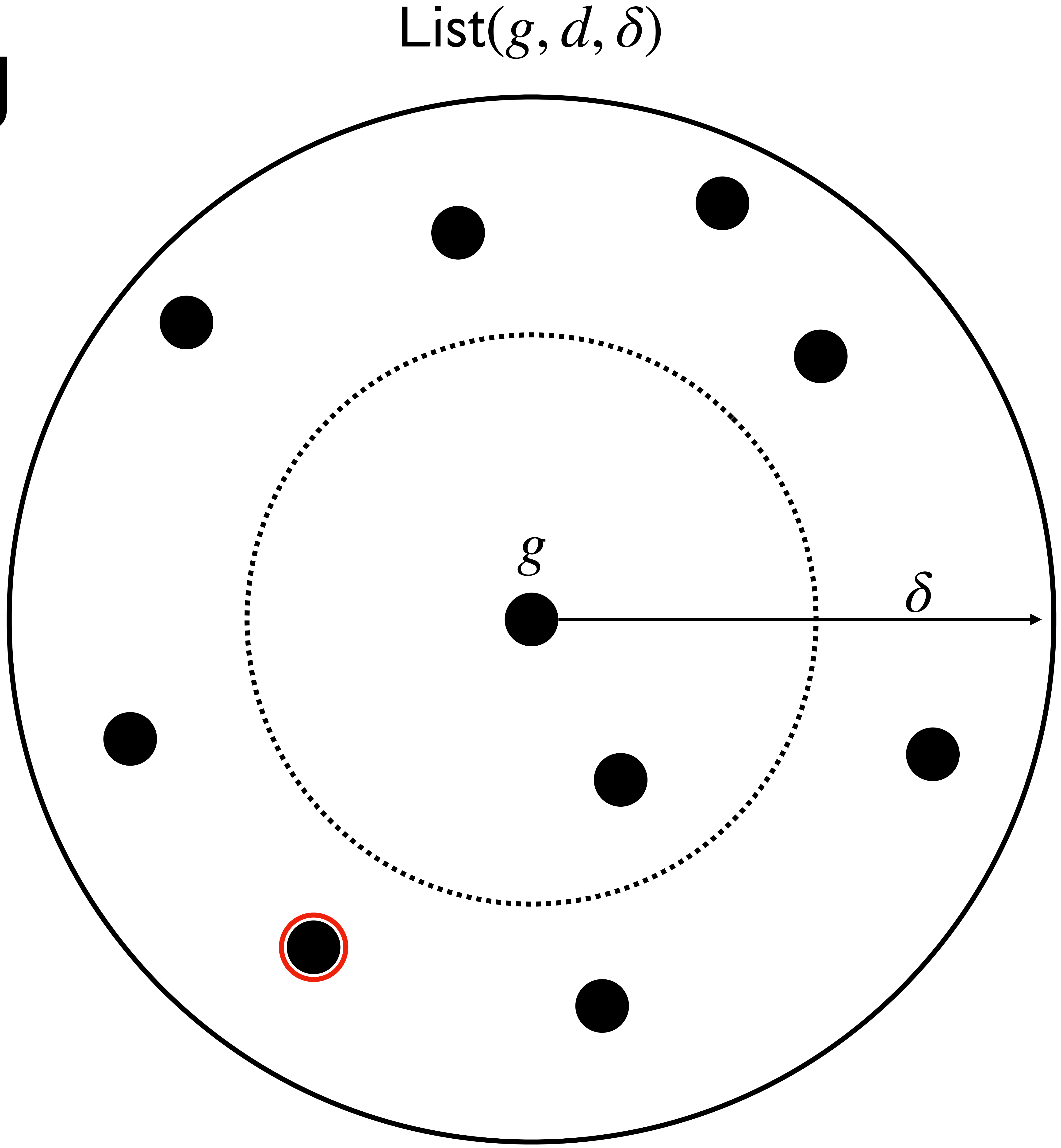
If every $\hat{v} \in \mathsf{List}(f, d, \delta)$ has $\hat{v}\,|_S \not\equiv \mathsf{Ans}$ then $\mathsf{Quotient}(f, \mathsf{Ans})$ is δ-far from RS

Out Of Domain sampling

Move to unique decoding range



By fundamental theorem of algebra there is at most **one** $\hat{u} \in \mathsf{List}(g, d, \delta)$ such that $\hat{u}(\alpha) = \beta$, w.h.p.



Use $\mathsf{Quotient}(g, \alpha \mapsto \beta)$ to enforce the constraint

STIR iteration

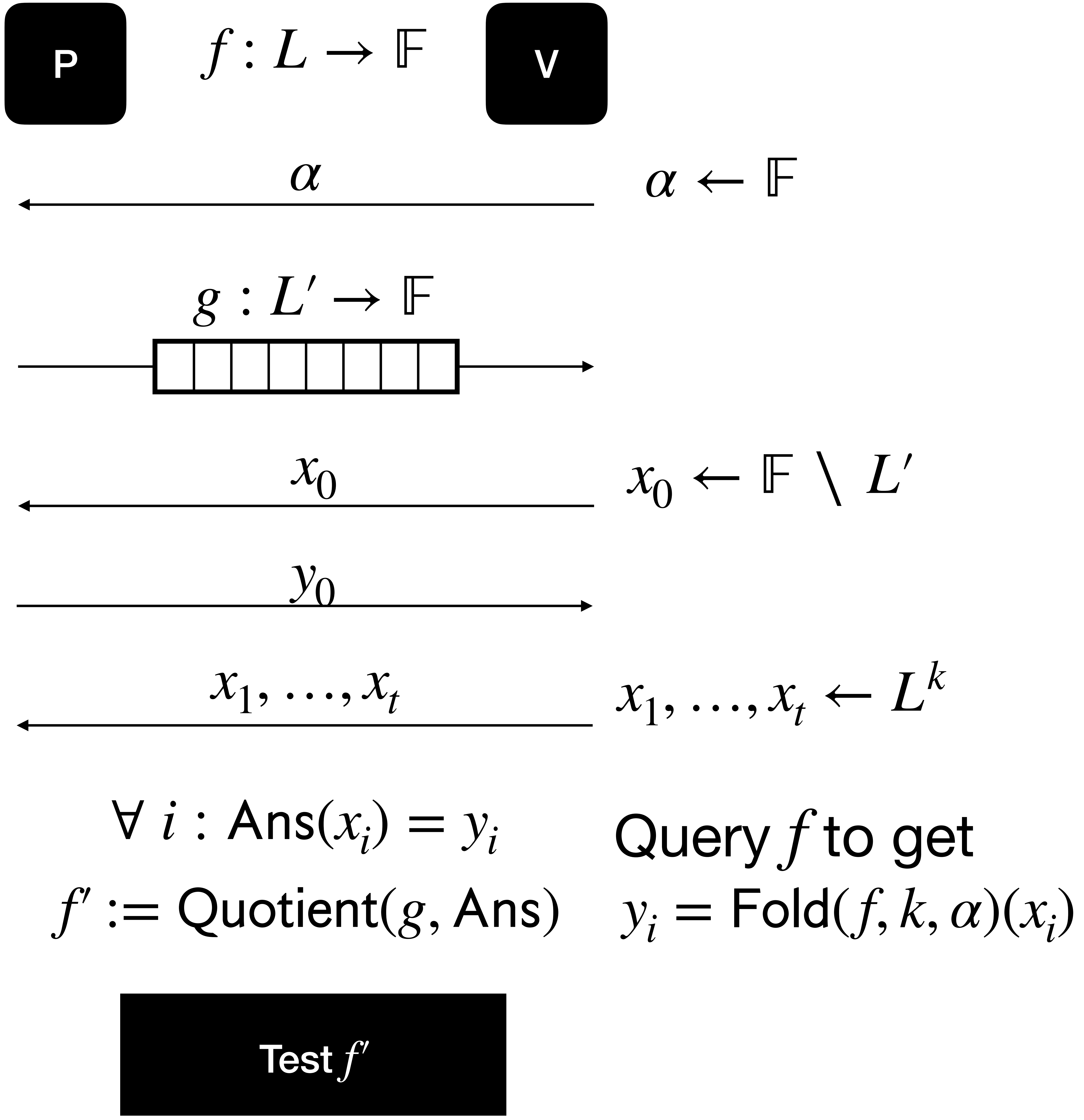
g is *claimed* to be equal to (the extension of) Fold(f, k, α) on L'

Problem: We can only query Fold(f, k, α) on $L^k \neq L'$.

Enforce consistency via Quotient!

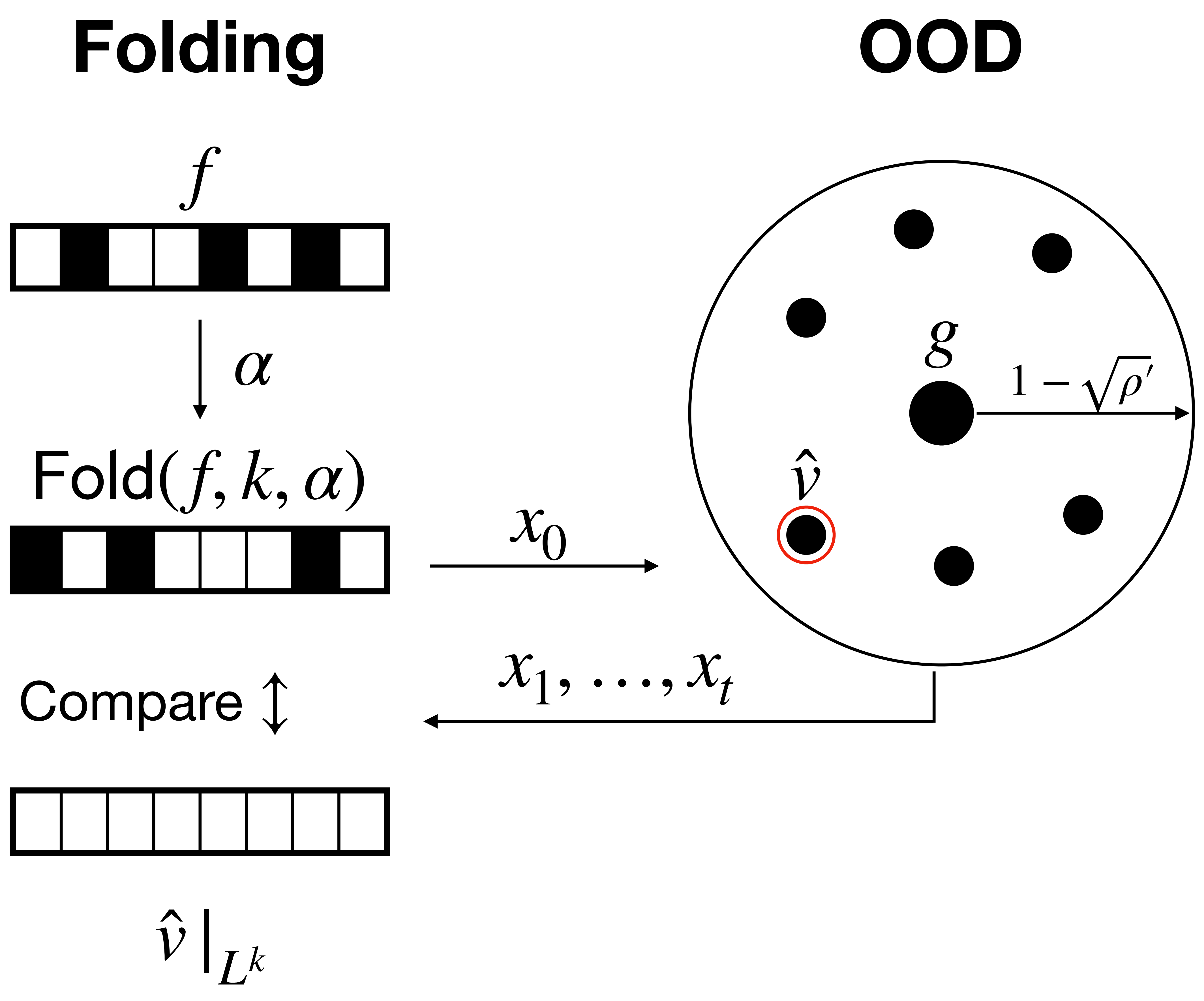
Query Fold(f, k, α) at $x_1, \ldots, x_t \in L^k$ to get $y_1, \ldots, y_t$

New function is quotient of g w.r.t. to these points + OOD sample



Soundness Analysis

Claim: if f is δ-far from C, unless with probability $\approx (1-\delta)^t$, f' is $(1-\sqrt{\rho'})$ far from C'



$\hat{v}$ is **unique** close codeword to g with $\hat{v}(x_0) = y_0$

If at any point $\hat{v}(x_i) \neq y_i$ then, by **quotients,** f' is $(1-\sqrt{\rho'})$-far from C'

Since Fold is δ-far from the code, $\Delta(\hat{v}|_{L^k}, \text{Fold}(f,k,\alpha)) > \delta$

$$\Pr\left[f' \text{ is } 1-\sqrt{\rho'} \text{ close}\right]$$

$$\leq \Pr\left[\forall i,\ \hat{v}(x_i) = y_i\right]$$

$$= \Pr\left[\forall i,\ \hat{v}(x_i) = \text{Fold}(f,k,\alpha)(x_i)\right]$$

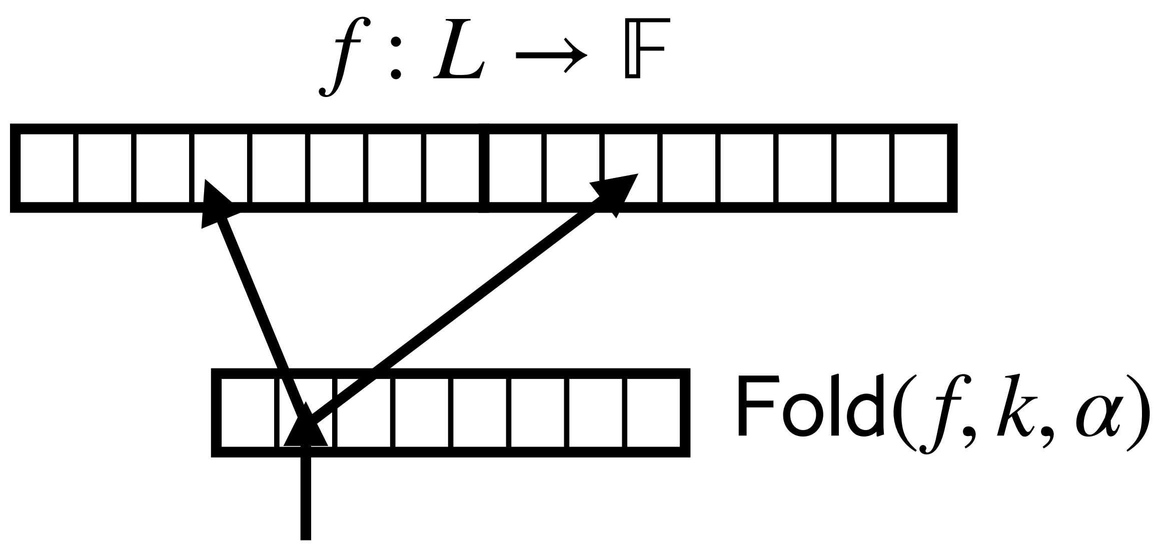
$$\leq (1-\delta)^t$$

Conclusion

What we saw

What we did have time to talk about

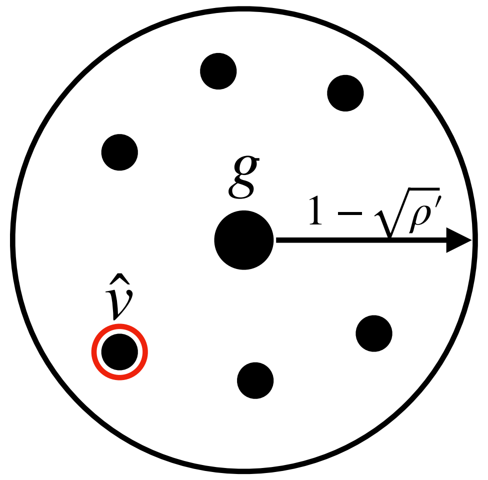
- Techniques
  - Folding and its properties
  - Quotienting and its properties
  - Out-Of-Domain sampling
- STIR 🥣
- Soundness analysis of STIR

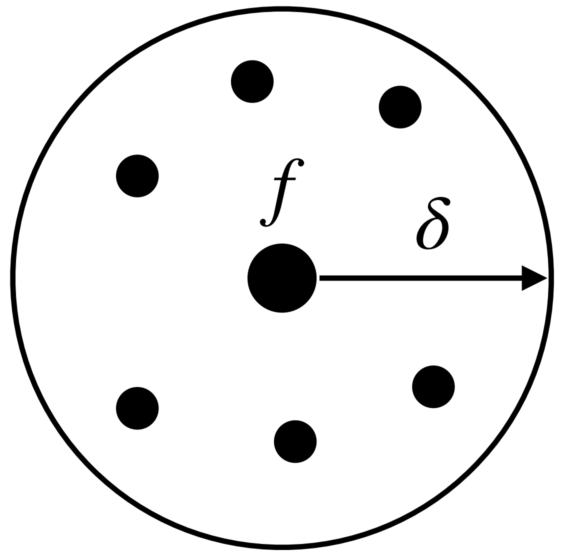


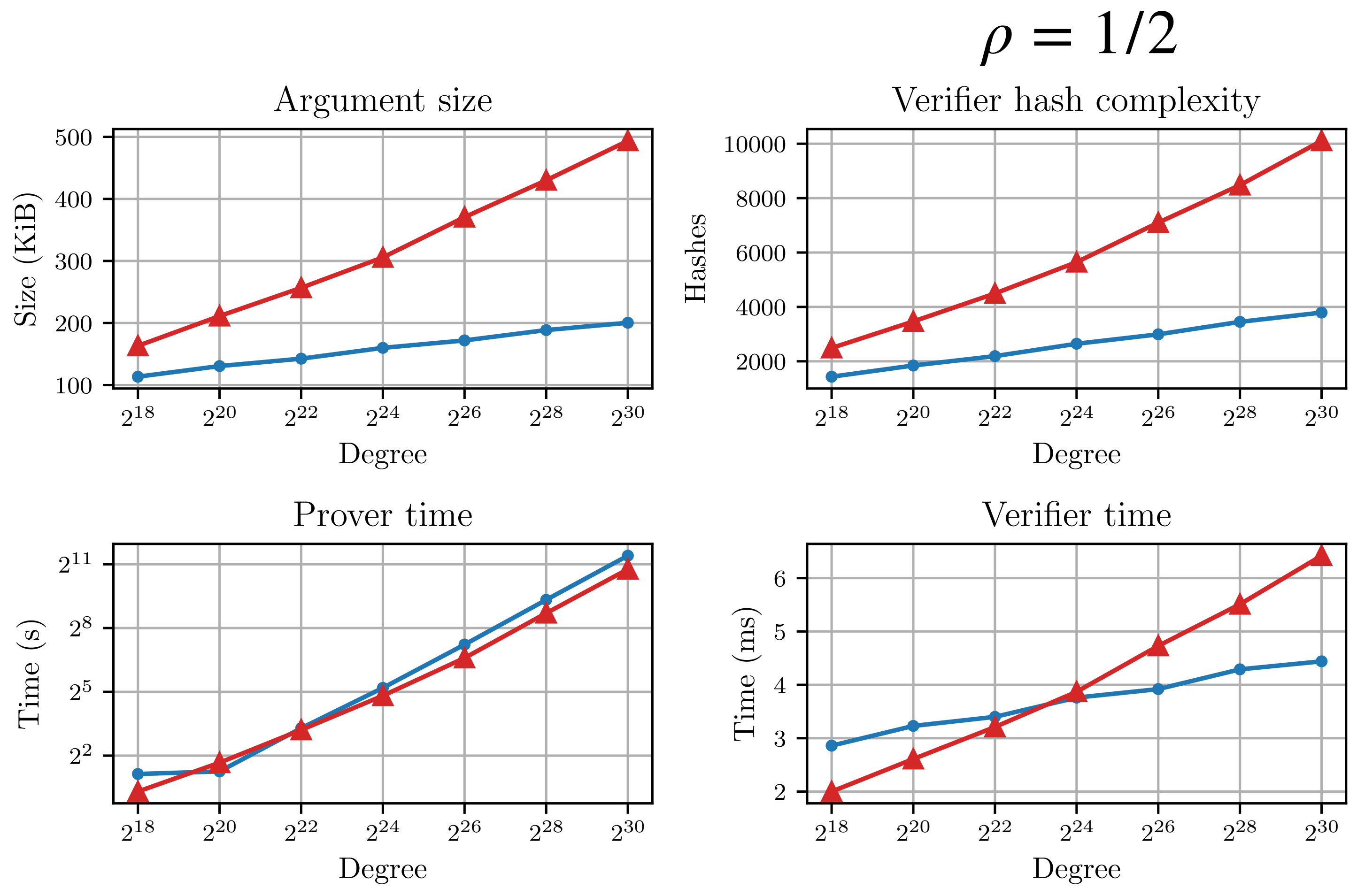
```
pub trait LowDegreeTest<F, MerkleConfig, FSConfig>
where
    F: FftField,
    MerkleConfig: Config,
    FSConfig: CryptographicSponge,
    FSConfig::Config: Clone,
{
    type Prover: Prover<
        F,
        MerkleConfig,
        FSConfig,
        Commitment = <Self::Verifier as Verifier<F, MerkleConfig, FSConfig>>::Commitment,
        Proof = <Self::Verifier as Verifier<F, MerkleConfig, FSConfig>>::Proof,
    >;
    type Verifier: Verifier<F, MerkleConfig, FSConfig>;

    fn instantiate(
        parameters: Parameters<F, MerkleConfig, FSConfig>,
    ) -> (Self::Prover, Self::Verifier) {
        let prover = Self::Prover::new(parameters.clone());
        let verifier = Self::Verifier::new(parameters);

        (prover, verifier)
    }
}
```







There is more!

What we did not have time to talk about

Degree corrections

- $\mathrm{Quotient}(f, \mathrm{Ans})$ has degree $d - |S|$, how to bump up to d?

High-soundness compiler for Poly-IOPs

- Builds on compiler in [ACY23] to achieve concrete efficiency

Round-by-round soundness of STIR $\implies$ secure in non-interactive setting

Wh… xt?

W… time to talk about next talk!

USE STIR🥣!!!

- O… …vid, Shahar 👀)
- Basefold S…

- **Breaking the $O(\log d)$-qu…**
  - Work in [ACY23] achieves $O(\log \log$…
  - Not concretely efficient, lacks efficient soundn…
  - Exciting!!!

Thank you!

See paper:
ia.cr/2024/390

And blog post:
gfenzi.io/papers/stir

Extra slides

What about the conjecture?

FRI and STIR benefit in roughly the same way

- Conjecture on list-decoding up to distance $1-\rho$ (instead of $1-\sqrt{\rho}$)
- FRI queries:

$$O\left(\lambda \cdot \frac{\log d}{-\log \rho}\right)$$

In both, for $\delta = 1-\rho$,

reduces queries by ~2x

- STIR queries:

$$O\left(\lambda \cdot \log\left(\frac{\log d}{-\log \rho}\right) + \log d\right)$$