Succinct Arguments

Lecture 05:
Holographic PIOP for R1CS

Pratyush Mishra
UPenn
Fall 2025

A PIOP for R1CS

R1CS

An rank-1 constraint system (R1CS) is a generalization of arithmetic circuits

$$(F := (\mathbb{F}, n \in \mathbb{N}, A, B, C), x, w)$$

$$z := \begin{bmatrix} x \\ w \end{bmatrix} \quad \overbrace{\quad}^{n} \; \underbrace{\begin{bmatrix} A \end{bmatrix}}_{n} \begin{bmatrix} z \end{bmatrix} \circ \begin{bmatrix} B \end{bmatrix} \begin{bmatrix} z \end{bmatrix} = \begin{bmatrix} C \end{bmatrix} \begin{bmatrix} z \end{bmatrix}$$

What checks do we need?

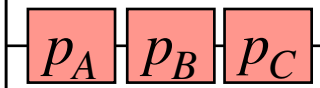
Step 1: Correct Hadamard product
check that for each i, $z_A[i] \cdot z_B[i] = z_C[i]$

Step 2: Correct matrix-vector multiplication
check that $Mz = z_M \quad \forall M \in \{A, B, C\}$

PIOP for Hadamard Product

$\mathbb{P}\text{rover}(F, x, w)$

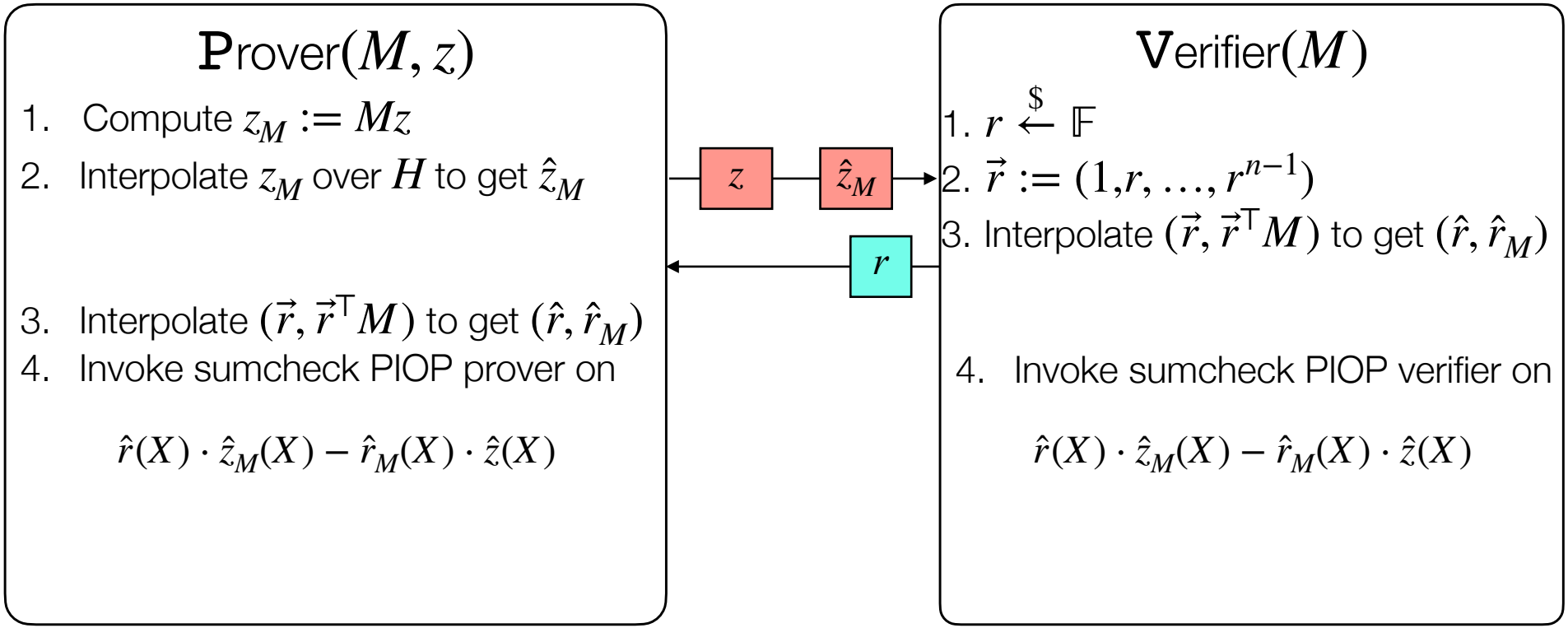
1. Let $H \subseteq \mathbb{F}$ be a set of size n.
2. Interpolate z_A, z_B, z_C to get p_A, p_B, p_C.
3. Run PIOP for zerocheck for polynomial $p_A \cdot p_B - p_C$.



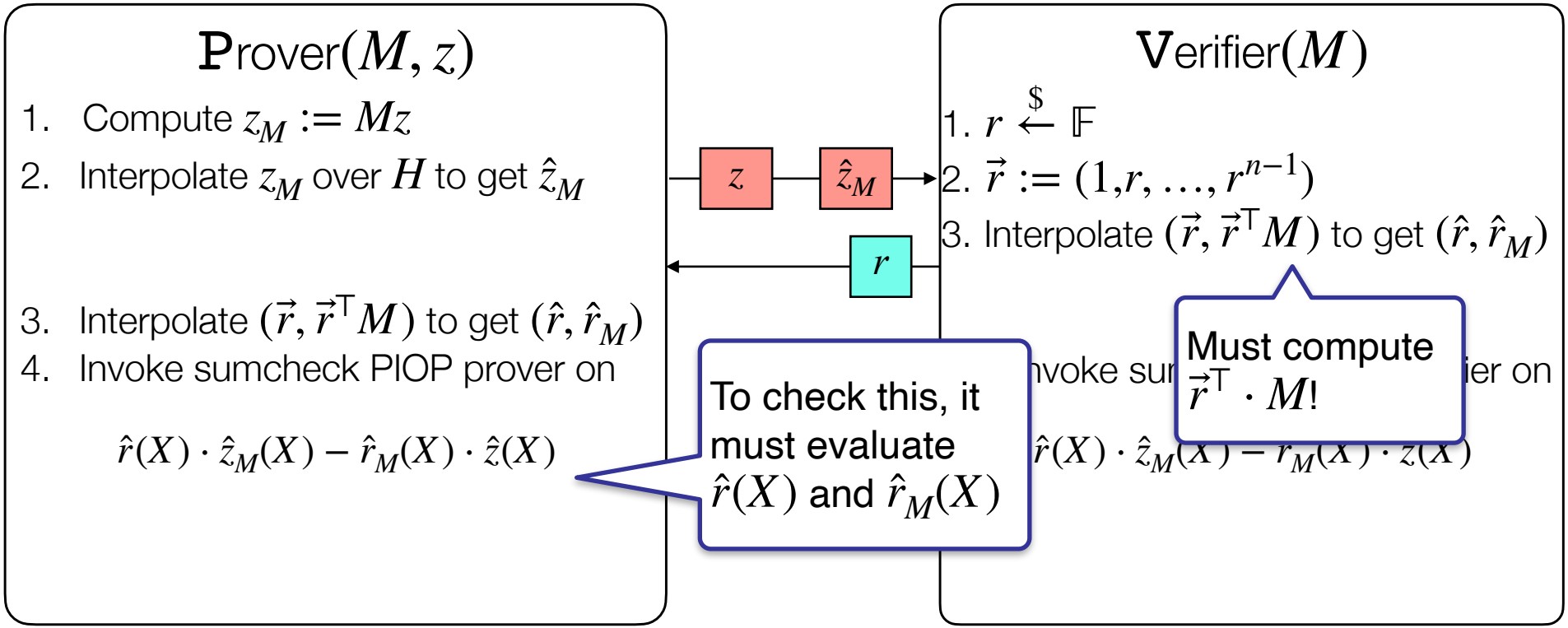
$\mathbb{V}\text{erifier}(F, x)$

Run PIOP verifier for zerocheck for polynomial $p_A \cdot p_B - p_C$.

Next point: *PIOP* for MV checks



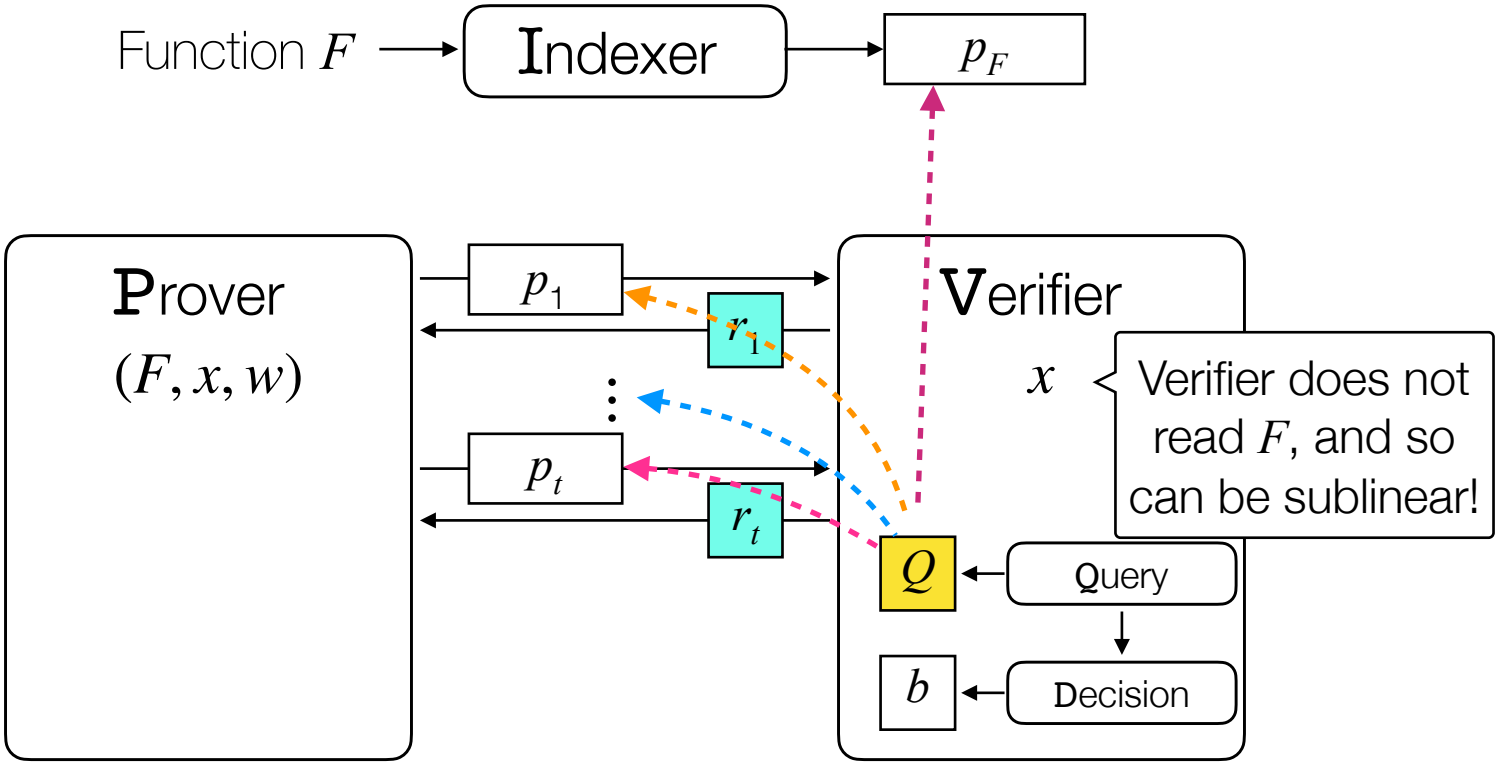
Why is the verifier slow?



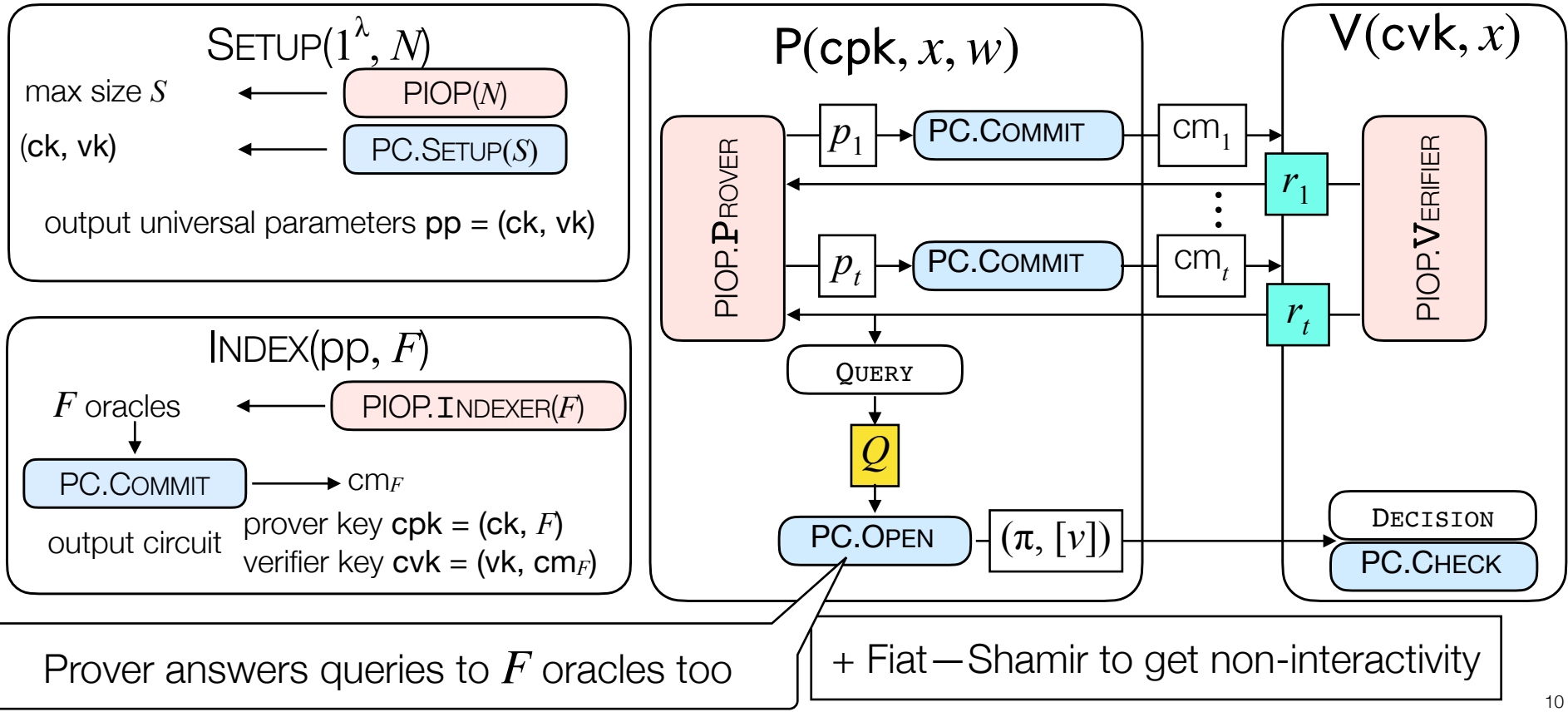
Sublinear verification for PIOP-based SNARKs

Holographic PIOPs [CHMMVW20, COS20]

Introduce a new algorithm to preprocess the matrices



Holographic PIOPs + PC Schemes → Preprocessing SNARKs



Verifier Complexity of Holographic PIOP-based SNARKs

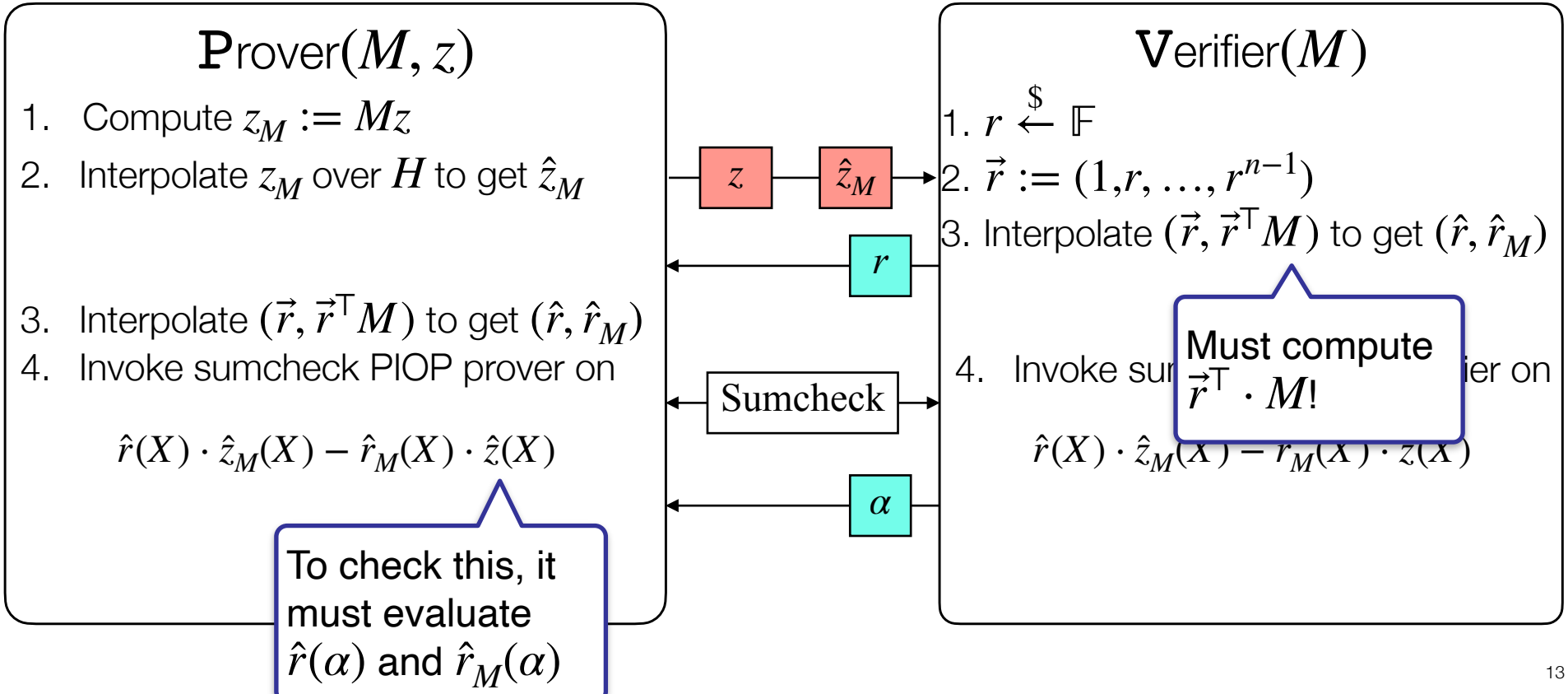
$T(\text{SNARK.V}) = T(\text{CHECK}) + T(\text{HIOP.}\mathbf{V})$

Now sublinear!

Holography enables sublinear verification for arbitrary circuits computations!

Holographic PIOP for R1CS

Why is the verifier slow?



Step 1: Efficient $\hat{r}(X)$

Can write $\hat{r}(X)$ as $\sum_{i \in H} r^i \cdot L_H^i(X)$

Efficiently evaluating this at a random point β
requires efficiently computing each r^i and $L_H^i(\beta)$

Let's interpret this as $\sum_{i \in H} Y^i \cdot L_H^i(X)$

Monomial basis

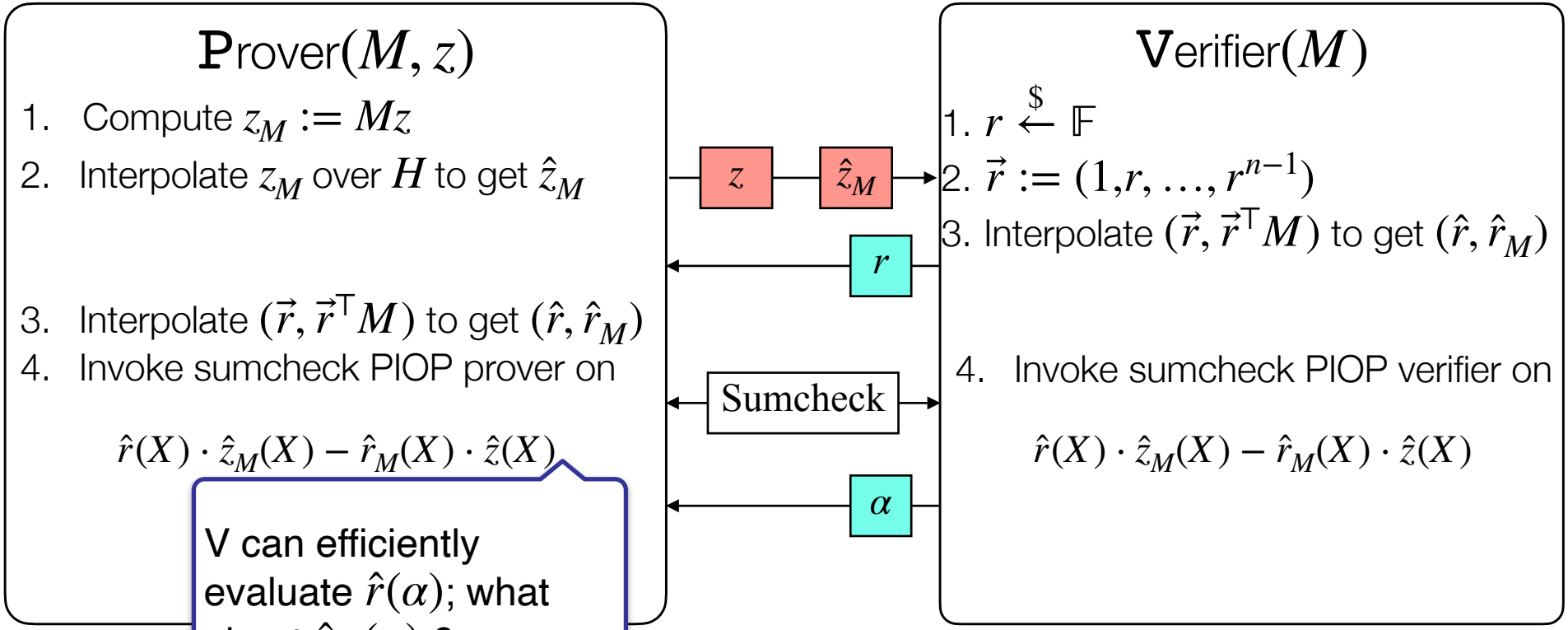
Lagrange basis

Step 1: Efficient $\hat{r}(X)$

1. Replace Monomial with Lagrange basis $\sum_{i \in H} L_H^i(Y) \cdot L_H^i(X)$
2. Can rewrite this as $\frac{v_H(Y)X - v_H(X)Y}{|H|(X - Y)}$

This can be evaluated in time $O(\log |H|)$!

Why is the verifier slow?



For all $j \in H$, $\hat{r}_M(j) = \sum_{i \in H} \hat{r}(i) \cdot \hat{M}(i, j)$.

So, therefore the interpolation looks like $\hat{r}_M(X) = \sum_{i \in H} \hat{r}(i) \cdot \hat{M}(i, X)$, and so

$$\hat{r}_M(\alpha) = \sum_{i \in H} \hat{r}(i) \cdot \hat{M}(i, \alpha)$$

This is yet another sumcheck, so we engage in another sumcheck PIOP, which will eventually result in requiring an evaluation $\hat{M}(\beta, \alpha)$, where β is random

How to evaluate $\hat{M}(\beta, \alpha)$?

How to encode matrix?

Polynomial Interpolation of Lists:

Given a list $A = (a_0, \ldots, a_d)$, and a set $H \subseteq \mathbb{F}$, the interpolation of A over H is

$$\hat{a}(X) := \sum_{i \in H} a_i \cdot L_H^i(X)$$

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot L_H^i(X) \cdot L_H^j(Y)$$

Problem: computing this requires $O(|H|^2)$ work

Insight: The matrices are sparse!

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot L_H^i(X) \cdot L_H^j(Y)$$

Most M_{ij} are zero!

Can rewrite as $\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot \frac{v_H(X)}{X - i} \cdot \frac{v_H(Y)}{Y - j}$,

Additionally, sum only over non-zero entries!

Final Matrix Encoding

Let m be the number of non-zero entries, and $K \subset \mathbb{F}$ be a subset of size m. Then, a *sparse* bivariate interpolation of A over K is

$$\hat{M}(X,Y) := \sum_{k \in K} \mathsf{v}(k) \cdot \frac{v_H(X)}{X - \mathsf{r}(k)} \cdot \frac{v_H(Y)}{Y - \mathsf{c}(k)}$$

Value of k-th non-zero entry

Row-index of k-th non-zero entry

Column-index of k-th non-zero entry

Actually, we need only m terms, so we will replace $\mathsf{r}, \mathsf{c}, \mathsf{v}$ with their interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X,Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$

Final Matrix Encoding

Let m be the number of non-zero … set of size m.
Then, a *sparse* bivariate interpolat…

$$\hat{M}(X,Y) := \sum_{k \in K} \ldots$$

Q: How to do this sumcheck?

This is a rational function!

We only know how to do sumcheck for polynomials!

Actually, we need polynomials, so … eir interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X,Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$

Final Matrix Encoding

Let m be the number of non-zero [illegible] set of size m.
Then, a *sparse* bivariate interpolat[illegible]

$$\hat{M}(X,Y) := \sum_{k \in K} [illegible]$$

Q: How to do this sumcheck?

This is a rational function!

We only know how to do sumcheck for polynomials!

Actually, we need polynomials, so [illegible]eir interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X,Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$

Final Matrix Encoding

Let m be the number of non-zero [illegible] set of size m.
Then, a *sparse* bivariate interpolat[illegible]

$$\hat{M}(X, Y) := \sum_{k \in K} [illegible]$$

A: interpolate into a polynomial!

Actually, we need polynomials, so [illegible] eir interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X, Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$

Rational → Polynomial

Key point: just like for other functions, here we just care about behavior of $\frac{v_H(X)}{X - \hat{\mathsf{r}}(k)}$ over H

So we will replace with an interpolation p, and

1. Perform sumcheck with p, and
2. Check that $p - \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} = 0$ over H.

What to do for multilinear case?

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot \mathrm{eq}(i, X) \cdot \mathrm{eq}(j, Y)$$

Most M_{ij} are zero!

Cannot rewrite as $\hat{M}(X, Y) := \sum_{i \in H} \sum_{j \in H} M_{ij} \cdot \frac{v_H(X)}{X - i} \cdot \frac{v_H(Y)}{Y - j}$!

We can still try to exploit sparsity!

Polynomial Interpolation of Matrices?:

Given a list $M \in \mathbb{F}^{n \times n}$, and a set $H \subseteq \mathbb{F}$, the bivariate interpolation of A over H is

$$\hat{M}(X, Y) := \sum_{i \in K} M_k \cdot \mathrm{eq}(i, \mathsf{r}(k)) \cdot \mathrm{eq}(j, \mathsf{c}(k))$$

Final Matrix Encoding

Let m be the number of non-zero entries, and $K \subset \mathbb{F}$ be a subset of size m. Then, a *sparse* bivariate interpolation of A over K is

$$\hat{M}(X, Y) := \sum_{k \in K} \mathsf{v}(k) \cdot \frac{v_H(X)}{X - \mathsf{r}(k)} \cdot \frac{v_H(Y)}{Y - \mathsf{c}(k)}$$

Value of k-th non-zero entry

Row-index of k-th non-zero entry

Column-index of k-th non-zero entry

Actually, we need only m of them, so we will replace $\mathsf{r}, \mathsf{c}, \mathsf{v}$ with their interpolations over K, i.e. $\hat{\mathsf{r}}, \hat{\mathsf{c}}, \hat{\mathsf{v}}$

$$\hat{M}(X, Y) := \sum_{k \in K} \hat{\mathsf{v}}(k) \cdot \frac{v_H(X)}{X - \hat{\mathsf{r}}(k)} \cdot \frac{v_H(Y)}{Y - \hat{\mathsf{c}}(k)}$$