LatticeFold+

Memory-Efficient Proving at Scale

Binyi Chen

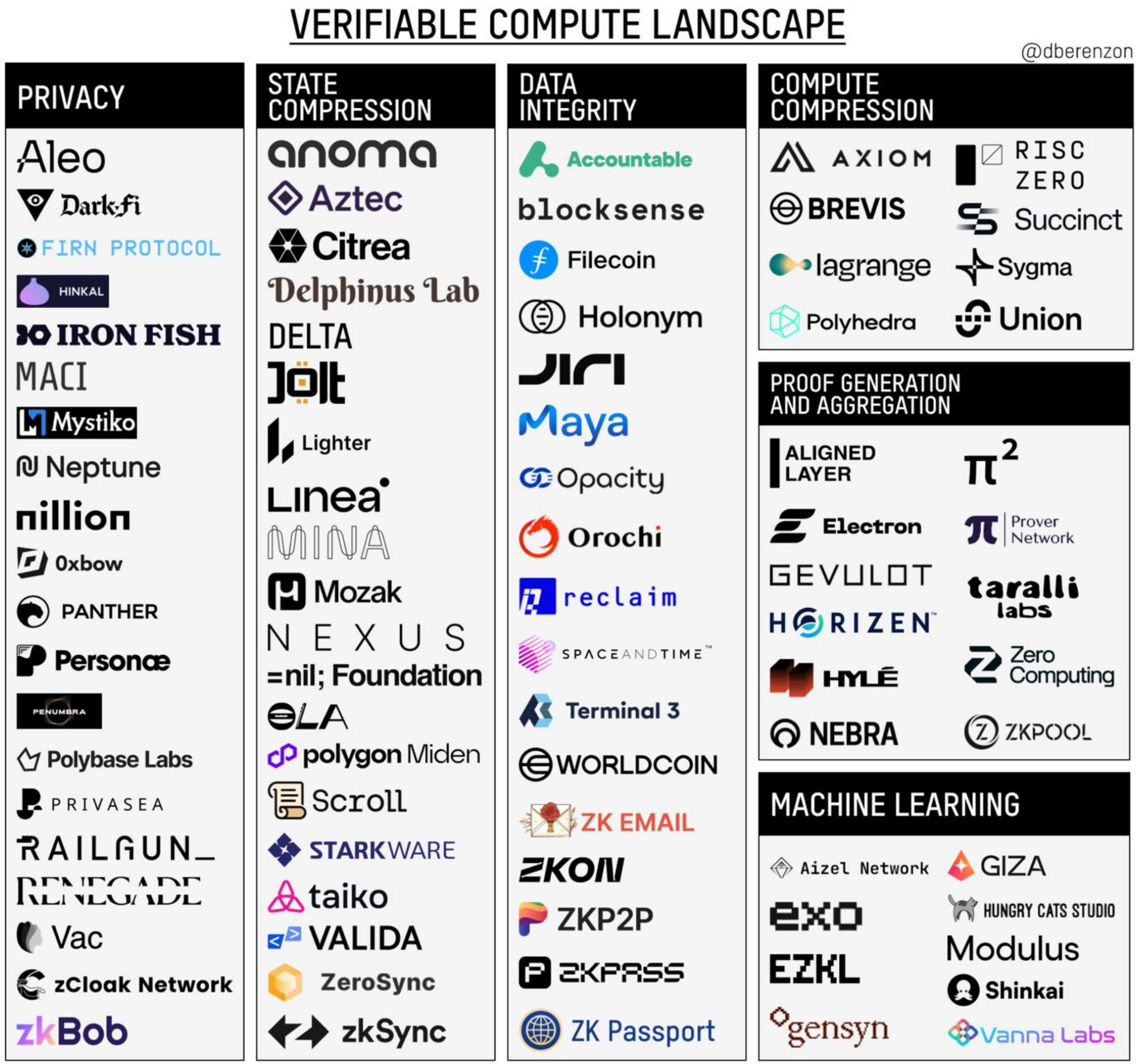
Based on joint work with Dan Boneh

Stanford University

ZK-SNARKs: Advanced Applications

Applications

- zkVM/zkML
- Image/Video Provenance
- ZK Wallet/Passport
- Data Availability
- Decentralized Storage
- ……



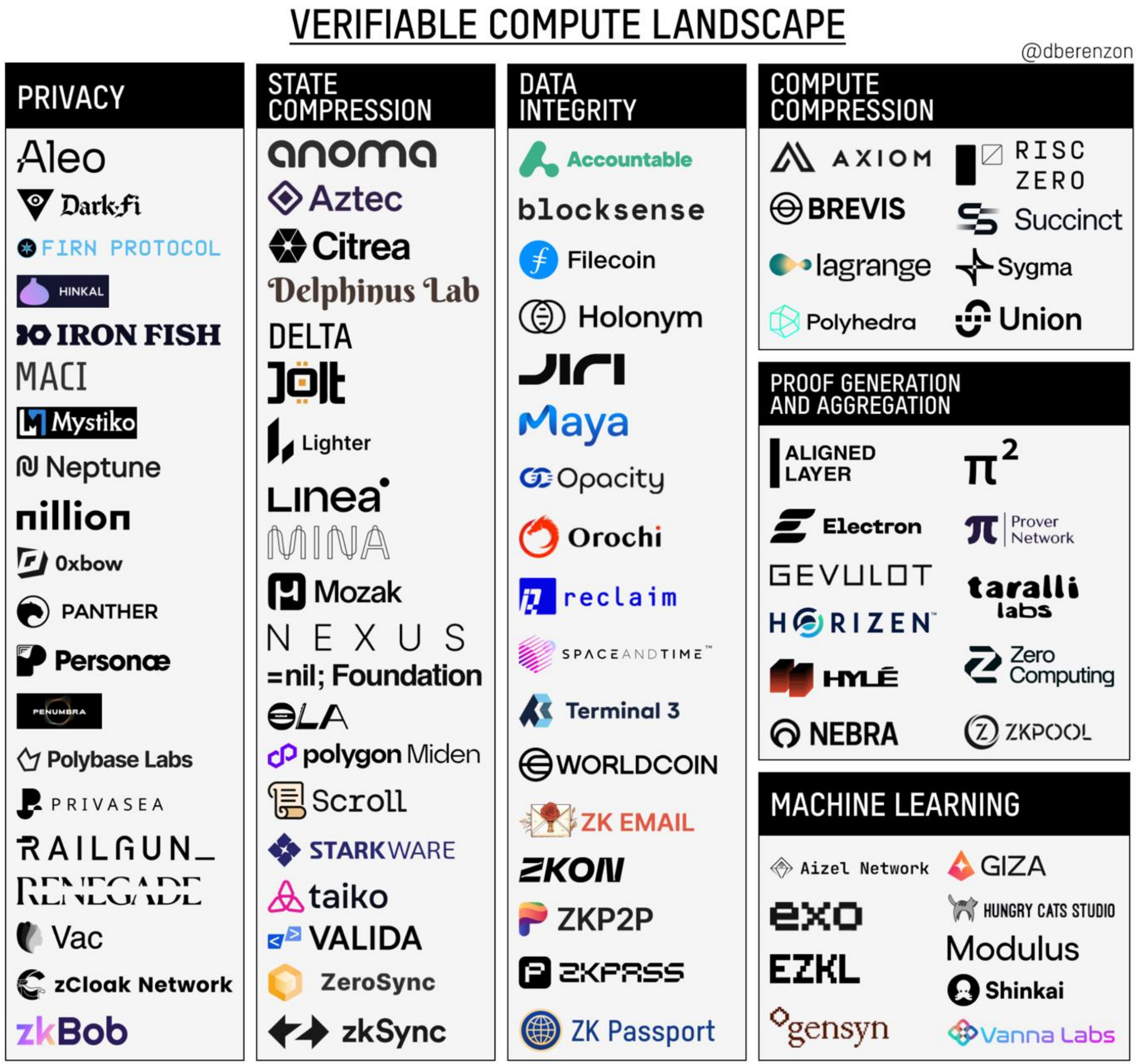
ZK-SNARKs: Advanced Applications

Applications

- zkVM/zkML
- Image/Video Provenance
- ZK Wallet/Passport
- Data Availability
- Decentralized Storage
- ……

Common Theme

Large-scale computation



ZK-SNARKs: Advanced Applications

Applications

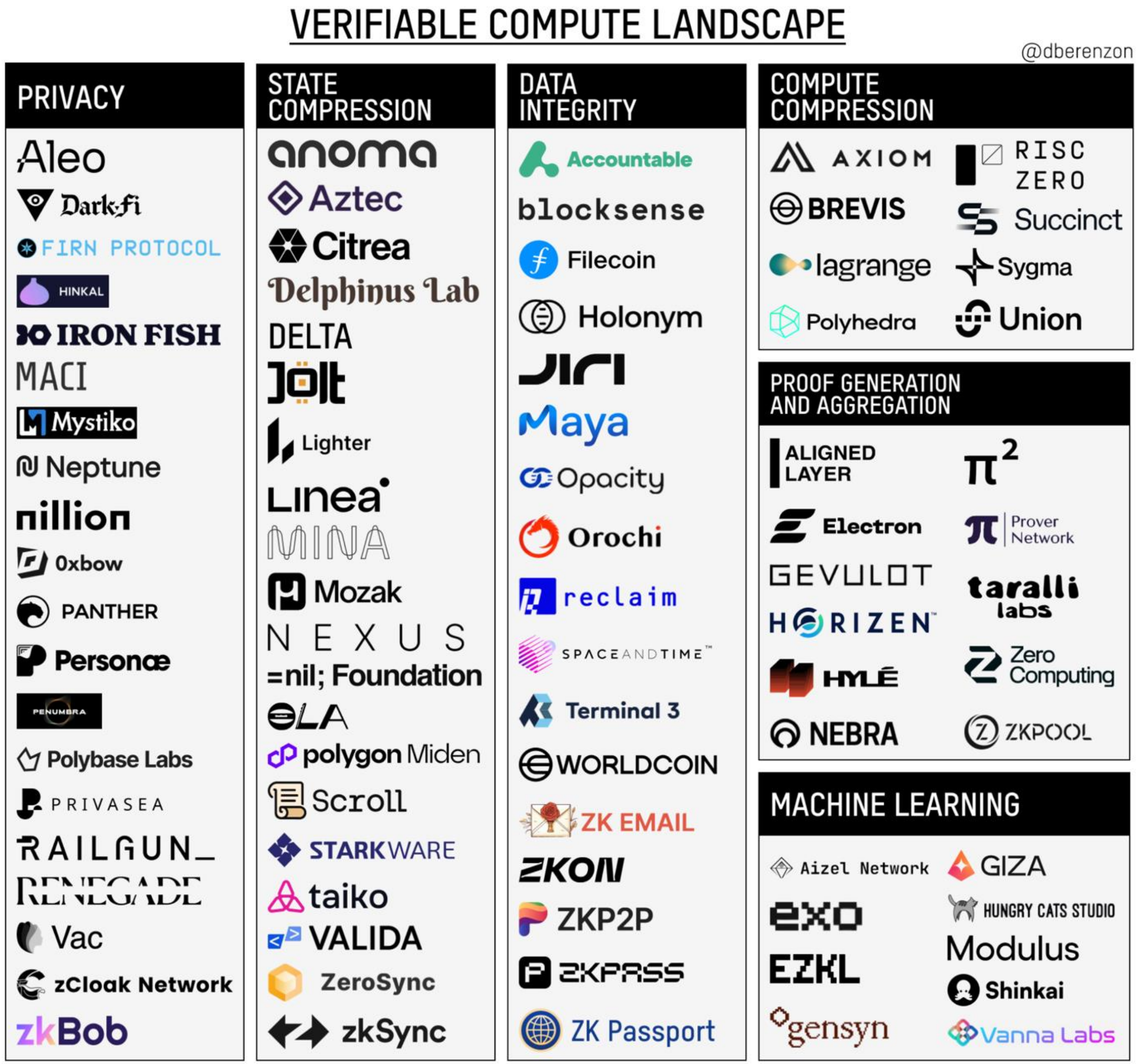
- zkVM/zkML
- Image/Video Provenance
- ZK Wallet/Passport
- Data Availability
- Decentralized Storage
- ……

Common Theme

Large-scale computation

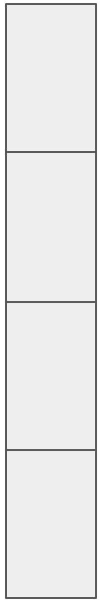
Design Requirements

Scalable prover + post-quantum security

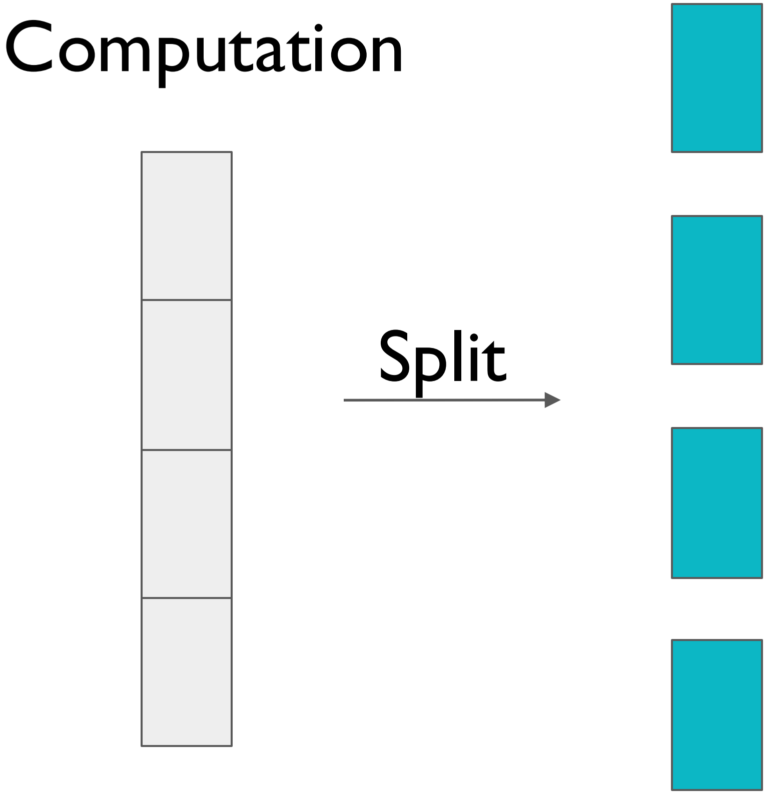


Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]

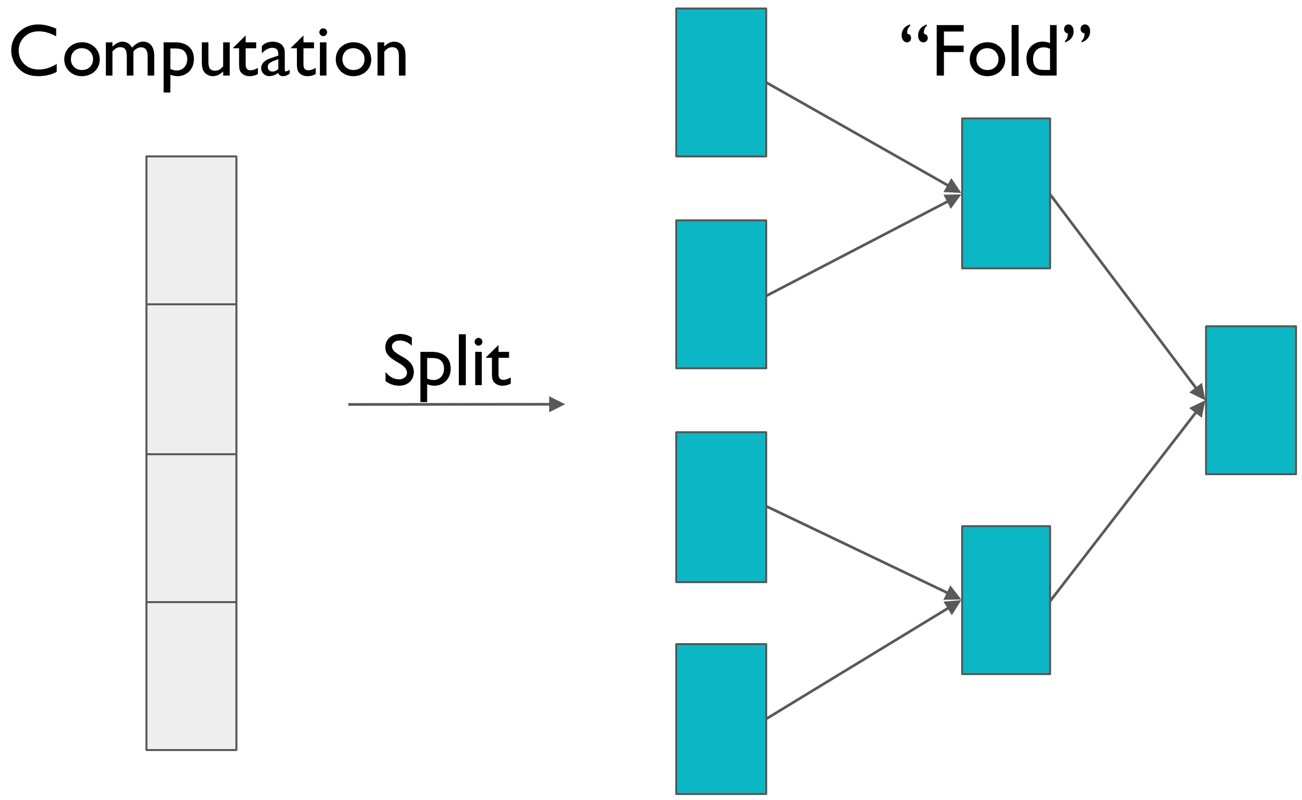
Computation



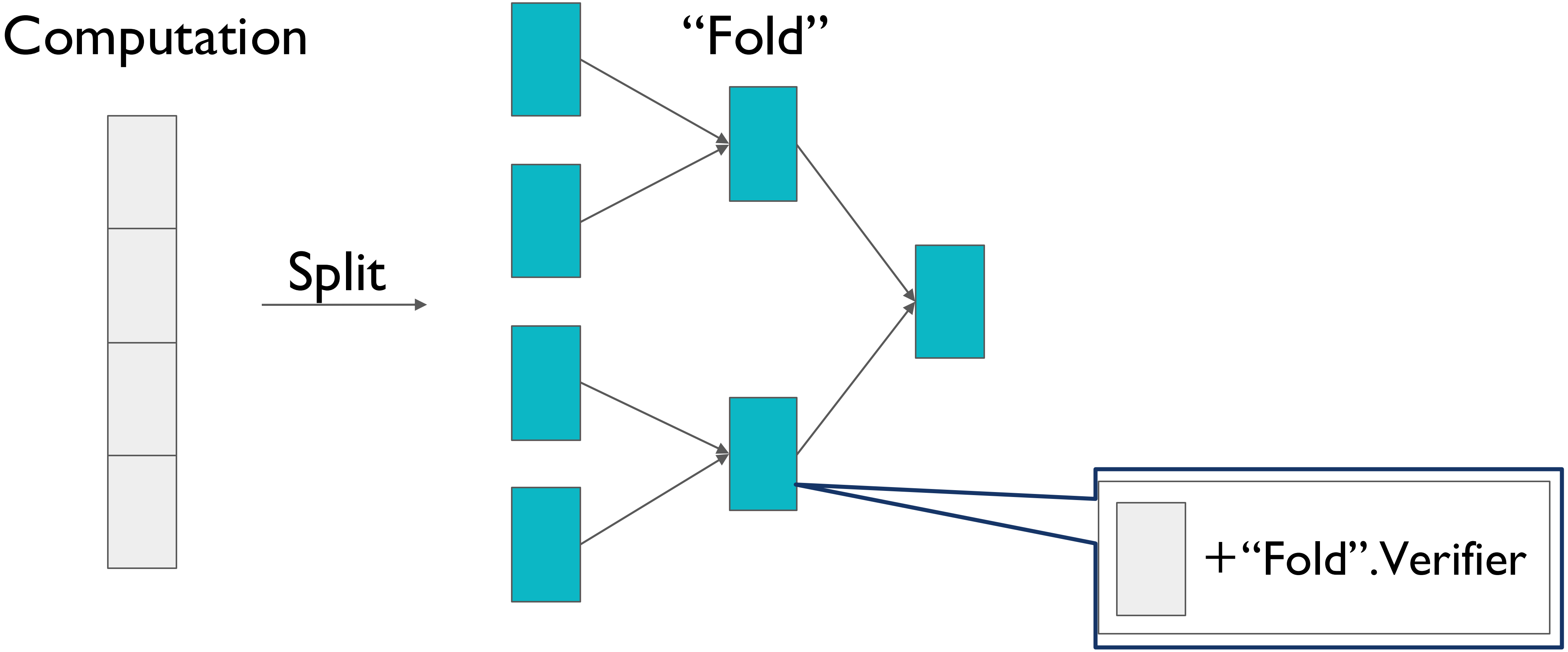
Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



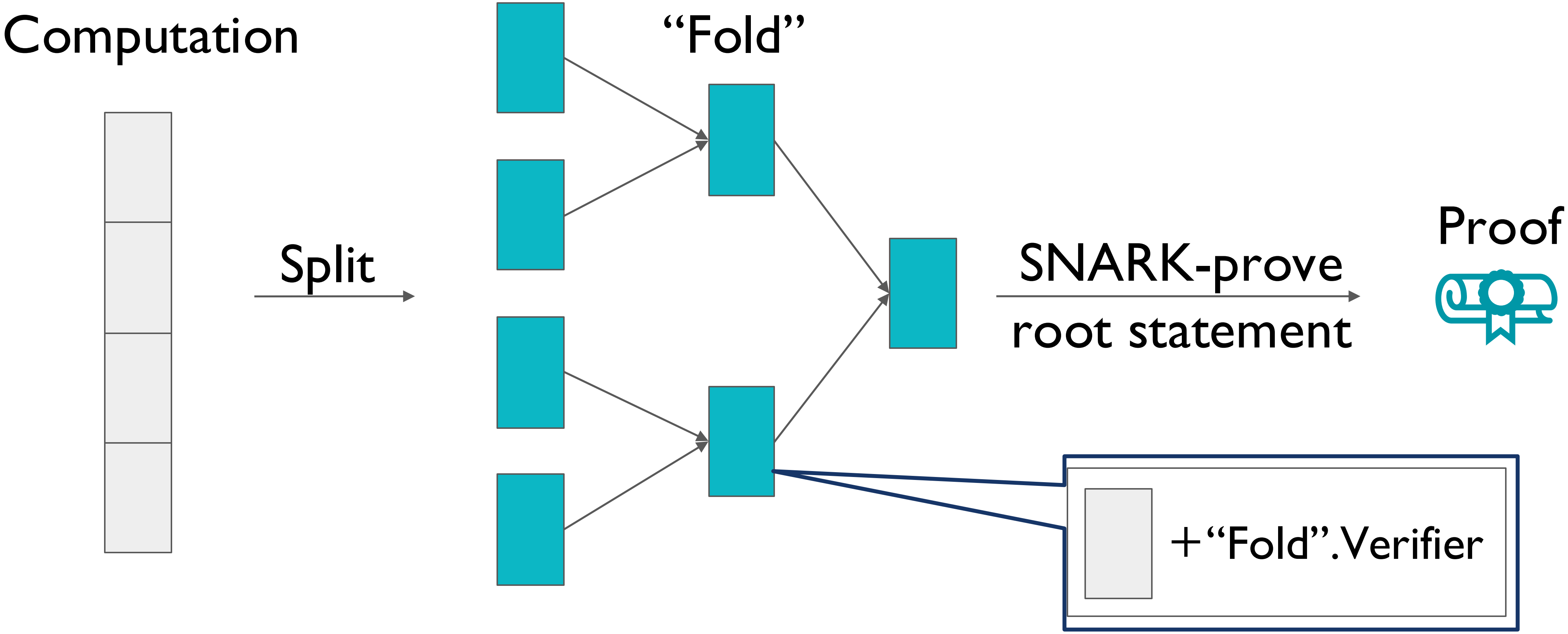
Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



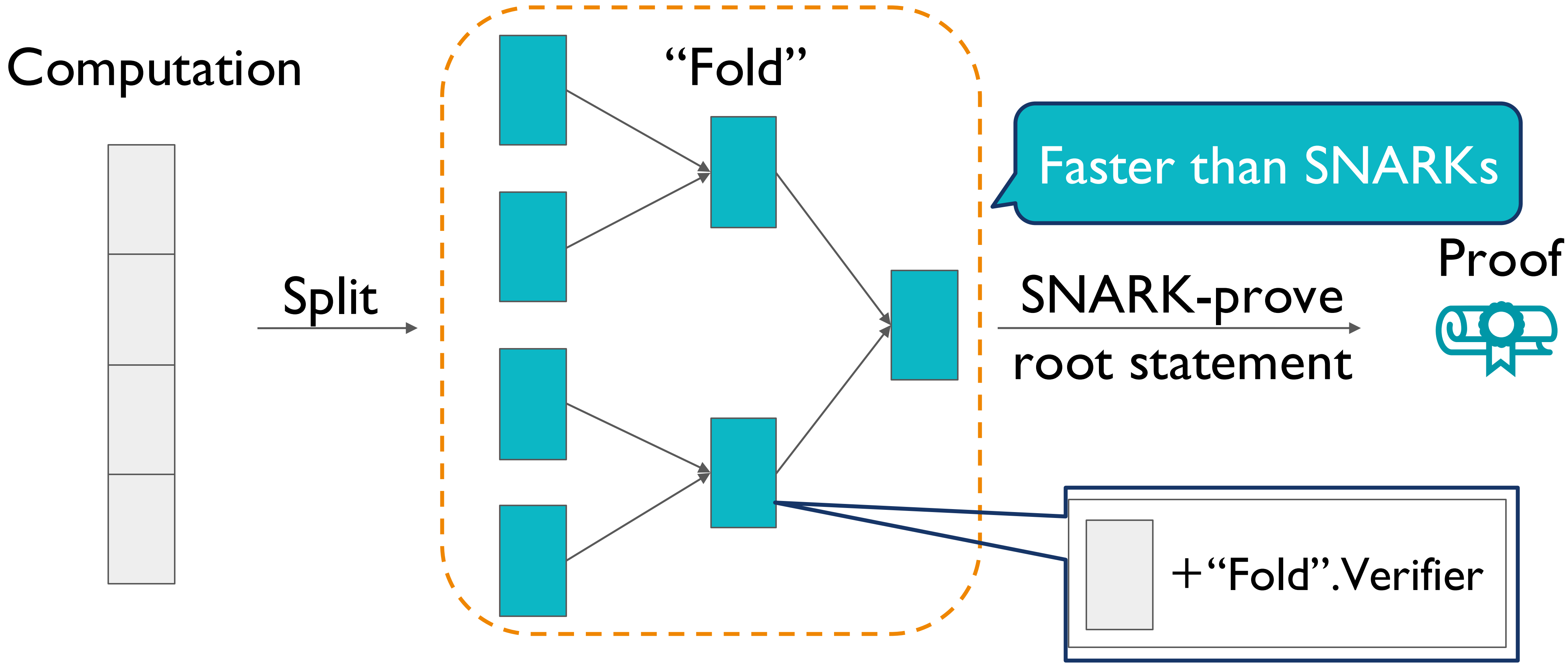
Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



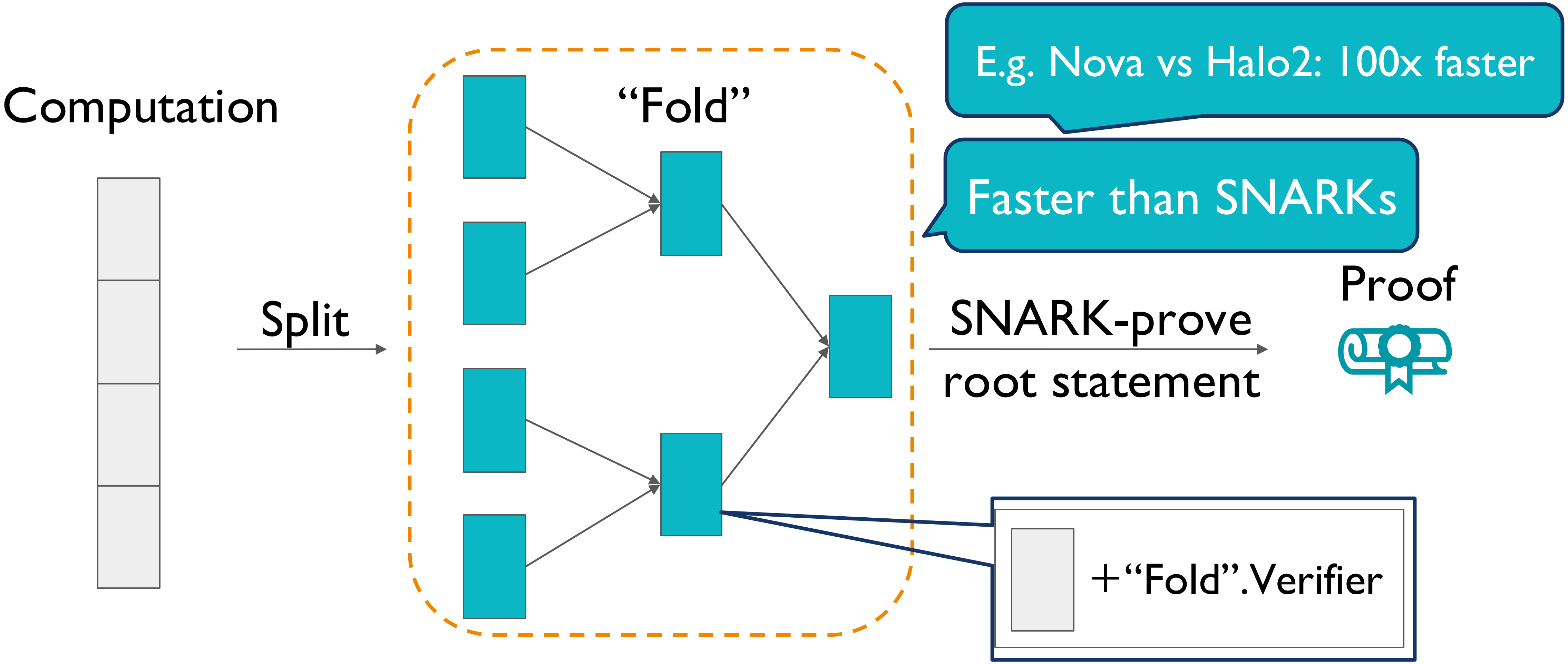
Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



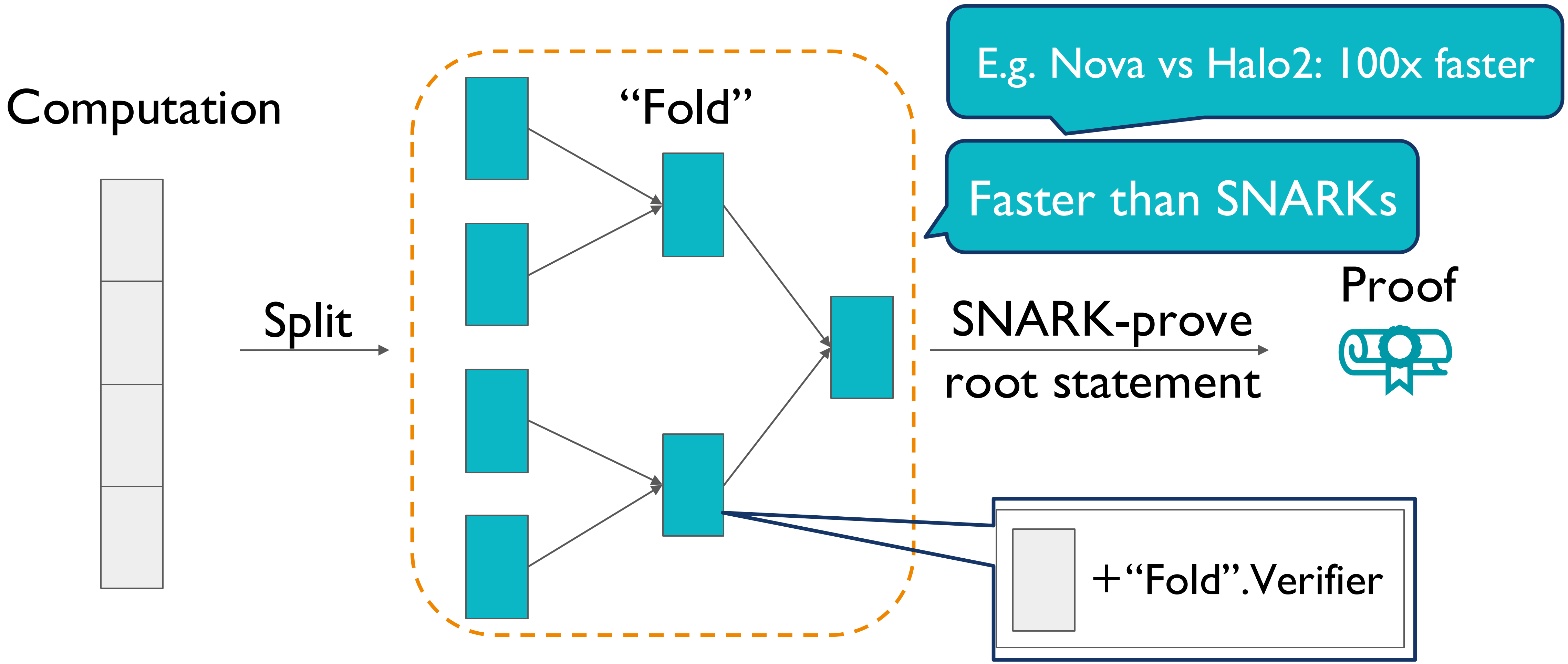
Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]

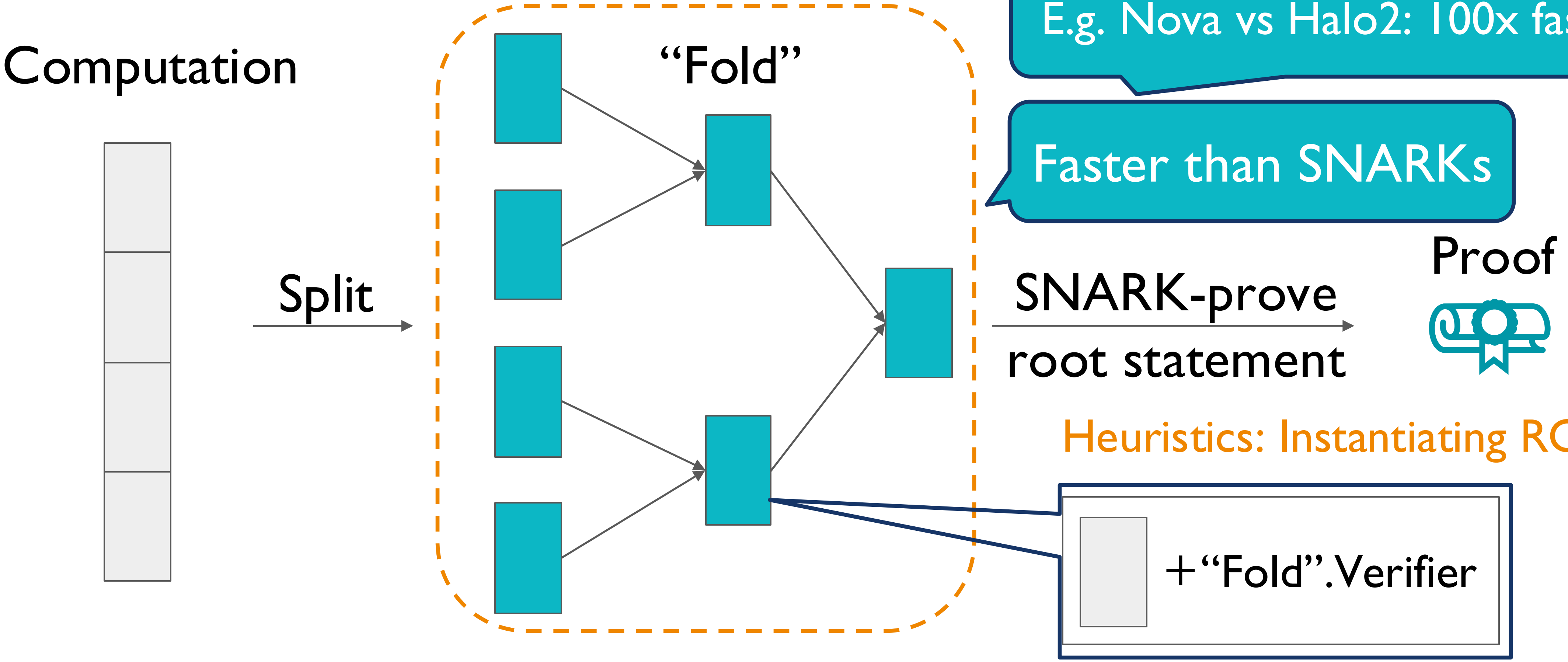


Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



Our focus: Public-coin interactive folding schemes

Piecemeal SNARKs [Val'08, BCCT'12, BGH'19, COS'20, NDCTB'24…]



E.g. Nova vs Halo2: 100x faster

Faster than SNARKs

Heuristics: Instantiating RO

Our focus: Public-coin interactive folding schemes

Folding Schemes [BGH'19, BCLMS'20, KST'21]

$(x_1, w_1) \in_? R_{NP}$

$(x_2, w_2) \in_? R_{acc}$

$$\Pi_{\mathrm{fd}}[P, V]$$

Folding Schemes [BGH'19, BCLMS'20, KST'21]

$(x_1, w_1) \in_? R_{NP}$

$(x_2, w_2) \in_? R_{acc}$

$$\longrightarrow \boxed{\Pi_{\mathrm{fd}}[P, V]} \longrightarrow$$

$(x, w) \in_? R_{acc}$

Folding Schemes [BGH'19, BCLMS'20, KST'21]

$(x_1, w_1) \in_? R_{NP}$

$(x_2, w_2) \in_? R_{acc}$

$$\Pi_{\mathrm{fd}}[P, V]$$

$(x, w) \in_? R_{acc}$

Properties:

- Completeness + Knowledge soundness + Succinctness
- Advantage: faster provers & verifiers than SNARKs

Folding Schemes [BGH'19, BCLMS'20, KST'21]

$|R_{NP}| \approx |R_{acc}|$

$(x_1, w_1) \in_? R_{NP}$

$(x_2, w_2) \in_? R_{acc}$

$$\longrightarrow \boxed{\Pi_{\mathrm{fd}}[P, V]} \longrightarrow$$

$(x, w) \in_? R_{acc}$

Properties:

- Completeness + Knowledge soundness + Succinctness
- Advantage: faster provers & verifiers than SNARKs

Lattice-Based Folding

Nice features

- Plausible post-quantum security
- Fast prover + succinct verifier

Lattice-Based Folding

Nice features

- Plausible post-quantum security
- Fast prover + succinct verifier

Much simpler than Lattice SNARKs

[BS'23], [NS'25], [CMNW'24], [KLNO'24], [HSS'24], [FLV'23, [CLM'23], [KLNO'25]

Lattice-Based Folding

Nice features

- Plausible post-quantum security
- Fast prover + succinct verifier

Prior/Concurrent Work

- LatticeFold [BC'24] / Neo [NS'25] :
  - MSIS [LS'15] + Sumcheck-based norm proof
- Lova [FKNP'24] :
  - SIS [Ajtai'96] + Euclidean norm proof
  - Less concretely efficient

Lattice-Based Folding

Nice features

- Plausible post-quantum security
- Fast prover + succinct verifier

Prior/Concurrent Work

- LatticeFold [BC'24] / Neo [NS'25] :
  - MSIS [LS'15] + Sumcheck-based norm proof
- Lova [FKNP'24] :
  - SIS [Ajtai'96] + Euclidean norm proof
  - Less concretely efficient

Prover: Require computing *many* helper commitments

Our Contribution

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ (e.g. q = 64-bit prime, $d = 64$)
$\kappa \cdot d$ = Lattice dimension
n = # of ring elems in input witness

LatticeFold+:

Our Contribution

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ (e.g. q = 64-bit prime, $d = 64$)
$\kappa \cdot d$ = Lattice dimension
n = # of ring elems in input witness

LatticeFold+:

- Security assumption: Module-SIS [LS'15]

Our Contribution

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ (e.g. q = 64-bit prime, $d = 64$)
$\kappa \cdot d$ = Lattice dimension
n = # of ring elems in input witness

Highly parallelizable

LatticeFold+:

- Prover cost per $\mathbb{Z}_q$: $O(\kappa \log d)$ $\mathbb{Z}_q$-muls + $O(\kappa d)$ $\mathbb{Z}_q$-adds
- Security assumption: Module-SIS [LS'15]

Our Contribution

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ (e.g. q = 64-bit prime, $d = 64$)
$\kappa \cdot d$ = Lattice dimension
n = # of ring elems in input witness

Highly parallelizable

LatticeFold+:

- Prover cost per $\mathbb{Z}_q$: $O(\kappa \log d)$ $\mathbb{Z}_q$-muls + $O(\kappa d)$ $\mathbb{Z}_q$-adds
- Verifier: $O(\kappa d + \log n)$ $\mathbb{F}_q$-hashes + $O(d)$ R_q-ops
- Security assumption: Module-SIS [LS'15]

Our Contribution

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ (e.g. q = 64-bit prime, $d = 64$)
$\kappa \cdot d$ = Lattice dimension
n = # of ring elems in input witness

Key technique

- An algebraic norm proof without high-arity decomposition
- $\log(d)$-factor fewer helper commitments than prior work

LatticeFold+:

- Prover cost per $\mathbb{Z}_q$: $O(\kappa \log d)$ $\mathbb{Z}_q$-muls + $O(\kappa d)$ $\mathbb{Z}_q$-adds
- Verifier: $O(\kappa d + \log n)$ $\mathbb{F}_q$-hashes + $O(d)$ R_q-ops
- Security assumption: Module-SIS [LS'15]

Technical Overview

A Simpler Goal

$$R_{\mathrm{cm},B} = \left\{ \begin{array}{c} \mathbb{x} = (\mathrm{cm}), \mathbb{w} = (f \in R^n) \ \text{s.t.} \\ \mathrm{cm} = \mathbf{A} f \bmod q \wedge \|f\| < B \end{array} \right\}$$

$\mathbf{A} \leftarrow_{\$} R_q^{\kappa \times n}$

$\|\cdot\|$: infinity norm

A Simpler Goal

$(x_1, w_1) \in_? R_{cm,B}$

$(x_2, w_2) \in_? R_{acc,B}$

$$\Pi_{\mathrm{fd}}[P, V]$$

$(x, w) \in_? R_{acc,B}$

$$R_{\mathrm{cm},B} = \left\{ \begin{matrix} \mathbb{x} = (\mathrm{cm}), \mathbb{w} = (f \in R^n) \ \text{ s.t.} \\ \mathrm{cm} = \mathbf{A}f \bmod q \wedge \|f\| < B \end{matrix} \right\}$$

A Simpler Goal

$(x_1, w_1) \in_? R_{cm,B}$

$(x_2, w_2) \in_? R_{acc,B}$

$\longrightarrow \boxed{\Pi_{\mathrm{fd}}[P, V]} \longrightarrow$ $(x, w) \in_? R_{acc,B}$

$$R_{\mathrm{cm},B} = \left\{ \begin{matrix} \mathbb{x} = (\mathrm{cm}), \mathbb{w} = (f \in R^n) \ \text{ s.t.} \\ \mathrm{cm} = \mathbf{A}f \bmod q \wedge \|f\| < B \end{matrix} \right\}$$

$R_{acc,B} \approx R_{cm,B}$

A Simpler Goal

$(\mathbb{x}_1, \mathbb{w}_1) \in_? R_{\mathrm{cm},B}$

$(\mathbb{x}_2, \mathbb{w}_2) \in_? R_{\mathrm{acc},B}$

$\Pi_{\mathrm{fd}}[P, V]$

Idea: random linear combination

$(\mathbb{x}, \mathbb{w}) \in_? R_{\mathrm{acc},B}$

$$R_{\mathrm{cm},B} = \left\{ \begin{matrix} \mathbb{x} = (\mathrm{cm}), \mathbb{w} = (f \in R^n) \ \text{ s.t.} \\ \mathrm{cm} = \mathbf{A}f \bmod q \wedge \|f\| < B \end{matrix} \right\}$$

$R_{\mathrm{acc},B} \approx R_{\mathrm{cm},B}$

A Simpler Goal

Challenge: Norm control

Idea: random linear combination

$(x_1, w_1) \in_? R_{cm,B}$

$(x_2, w_2) \in_? R_{acc,B}$

$\Pi_{\mathrm{fd}}[P, V]$

$(x, w) \in_? R_{acc,B}$

$$R_{\mathrm{cm},B} = \left\{ \begin{array}{c} \mathbb{x} = (\mathrm{cm}), \mathbb{w} = (f \in R^n) \ \text{ s.t.} \\ \mathrm{cm} = \mathbf{A}f \bmod q \wedge \|f\| < B \end{array} \right\}$$

$R_{acc,B} \approx R_{cm,B}$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S := \{g \in R, \|g\| \leq 1\}$

$[\mathrm{cm}_1, f_1]$

$[\mathrm{cm}_2, f_2]$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S \coloneqq \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{array}{l}[\mathrm{cm}_1, f_1] \\ \\ \\ [\mathrm{cm}_2, f_2]\end{array}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{array}{c}\mathrm{cm}_1 + \alpha \mathrm{cm}_2 \\ f^* \coloneqq f_1 + \alpha f_2\end{array}$$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S := \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{array}{l}[\mathrm{cm}_1, f_1] \\ \\ [\mathrm{cm}_2, f_2]\end{array}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{array}{c}\mathrm{cm}_1 + \alpha\mathrm{cm}_2 \\ f^* := f_1 + \alpha f_2\end{array}$$

- $|S| = \Omega(2^\lambda)$
- $\|f^*\| \leq O(\lambda) \cdot \|f_2\|$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S := \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{matrix}[\mathrm{cm}_1, f_1] \\ \\ [\mathrm{cm}_2, f_2]\end{matrix}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{matrix}\mathrm{cm}_1 + \alpha \mathrm{cm}_2 \\ f^* := f_1 + \alpha f_2\end{matrix}$$

- $|S| = \Omega(2^\lambda)$
- $\|f^*\| \leq O(\lambda) \cdot \|f_2\|$

Decomposition: [BC'24]

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S := \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{array}{l}[\mathrm{cm}_1, f_1] \\ \\ \\ [\mathrm{cm}_2, f_2]\end{array}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{array}{c}\mathrm{cm}_1 + \alpha \mathrm{cm}_2 \\ f^* := f_1 + \alpha f_2\end{array}$$

- $|S| = \Omega(2^\lambda)$
- $\|f^*\| \leq O(\lambda) \cdot \|f_2\|$

Decomposition: [BC'24]

$f \in R^n$
s.t. $\|f\| < b^k$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S \coloneqq \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{matrix}[\mathrm{cm}_1, f_1] \\ \\ [\mathrm{cm}_2, f_2]\end{matrix}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{matrix}\mathrm{cm}_1 + \alpha \mathrm{cm}_2 \\ f^* \coloneqq f_1 + \alpha f_2\end{matrix}$$

- $|S| = \Omega(2^\lambda)$
- $\|f^*\| \leq O(\lambda) \cdot \|f_2\|$

Decomposition: [BC'24]

$$f \in R^n \xrightarrow{\mathsf{Decomp}_{\mathrm{b,k}}(f)} f_1, f_2, \ldots, f_k \in R^n$$

s.t. $\|f\| < b^k$

Norm Control Techniques

$\|\cdot\|$: infinity norm

Low norm challenges: $S \coloneqq \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{matrix}[\mathrm{cm}_1, f_1] \\ \\ [\mathrm{cm}_2, f_2]\end{matrix}\right\} \xrightarrow{\alpha \leftarrow_R S} \begin{matrix}\mathrm{cm}_1 + \alpha \mathrm{cm}_2 \\ f^* \coloneqq f_1 + \alpha f_2\end{matrix}$$

- $|S| = \Omega(2^\lambda)$
- $\|f^*\| \leq O(\lambda) \cdot \|f_2\|$

Decomposition: [BC'24]

$$f \in R^n \xrightarrow{\mathrm{Decomp}_{b,k}(f)} f_1, f_2, \ldots, f_k \in R^n \begin{cases} f = \sum_i b^{i-1} \cdot f_i \\ \|f_1\|, \ldots, \|f_k\| < b \end{cases}$$

s.t. $\|f\| < b^k$

Decompose-and-Fold [BC'24]

$|| \cdot ||$: infinity norm

$S := \{g \in R, ||g|| \leq 1\}$

Norm upper-bound

$[\mathrm{cm}_1, f_1, b^k]$

$[\mathrm{cm}_2, f_2, b^k]$

Decompose-and-Fold [BC'24]

$\|\cdot\|$: infinity norm

$$S \coloneqq \{g \in R, \|g\| \le 1\}$$

$$\left.\begin{array}{l}[\mathrm{cm}_1, f_1, b^k] \\ \\ \\ [\mathrm{cm}_2, f_2, b^k]\end{array}\right\} \xrightarrow[\text{Decompose}]{\text{Prover}} \left\{\begin{array}{l}[\overline{\mathrm{cm}}_1, \bar{f}_1, b] \\ \quad \dots\dots \\ \quad \dots\dots \\ [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}, b]\end{array}\right.$$

Decompose-and-Fold [BC'24]

$\|\cdot\|$: infinity norm

$S := \{g \in R, \|g\| \leq 1\}$

$$[\mathrm{cm}_1, f_1, b^k], \quad [\mathrm{cm}_2, f_2, b^k]$$

$$\xrightarrow[\text{Decompose}]{\text{Prover}}$$

$$[\overline{\mathrm{cm}}_1, \bar{f}_1, b], \quad \dots \dots, \quad \dots \dots, \quad [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}, b]$$

$$\xrightarrow[\vec{\alpha} \leftarrow_R S^{2k}]{\text{Verifier}}$$

$$[\overline{\mathrm{cm}}^*, \bar{f}^*] = \sum \vec{\alpha}_i \cdot [\overline{\mathrm{cm}}_i, \bar{f}_i]$$

$$\approx 2k\lambda b$$

Decompose-and-Fold [BC'24]

$\|\cdot\|$: infinity norm

$$S := \{g \in R, \|g\| \leq 1\}$$

$$\left.\begin{matrix} [\mathrm{cm}_1, f_1, b^k] \\ \\ [\mathrm{cm}_2, f_2, b^k] \end{matrix}\right\} \xrightarrow[\text{Decompose}]{\text{Prover}} \left\{\begin{matrix} [\overline{\mathrm{cm}}_1, \bar{f}_1, b] \\ \cdots\cdots \\ \cdots\cdots \\ [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}, b] \end{matrix}\right\} \xrightarrow[\vec{\alpha} \leftarrow_R S^{2k}]{\text{Verifier}} [\overline{\mathrm{cm}}^*, \bar{f}^*] = \sum \vec{\alpha}_i \cdot [\overline{\mathrm{cm}}_i, \bar{f}_i]$$

$$\approx 2k\lambda b$$

$$\leq B = b^k$$

Decompose-and-Fold [BC'24]

$\|\cdot\|$: infinity norm

$S \coloneqq \{g \in R, \|g\| \leq 1\}$

$$\left.\begin{matrix}[\mathrm{cm}_1, f_1, b^k] \\ \\ [\mathrm{cm}_2, f_2, b^k]\end{matrix}\right\} \xrightarrow[\text{Decompose}]{\text{Prover}} \left\{\begin{matrix}[\overline{\mathrm{cm}}_1, \bar{f}_1, b] \\ \cdots\cdots \\ \cdots\cdots \\ [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}, b]\end{matrix}\right\} \xrightarrow[\vec{\alpha} \leftarrow_R S^{2k}]{\text{Verifier}} [\overline{\mathrm{cm}}^*, \bar{f}^*] = \sum \vec{\alpha}_i \cdot [\overline{\mathrm{cm}}_i, \bar{f}_i]$$

$\approx 2k\lambda b$

Warning: Satisfies completeness but not knowledge soundness

Decompose-and-Fold [BC'24]

$\|\cdot\|$: infinity norm

$S := \{g \in R, \|g\| \le 1\}$

Need a batch-PoK of norms

$[\mathrm{cm}_1, f_1, b^k]$

$[\mathrm{cm}_2, f_2, b^k]$

$\xrightarrow[\text{Decompose}]{\text{Prover}}$

$[\overline{\mathrm{cm}}_1, \bar{f}_1, b]$

… …

… …

$[\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}, b]$

$\xrightarrow[\vec{\alpha} \leftarrow_R S^{2k}]{\text{Verifier}}$

$[\overline{\mathrm{cm}}^*, \bar{f}^*] = \sum \vec{\alpha}_i \cdot [\overline{\mathrm{cm}}_i, \bar{f}_i]$

$\approx 2k\lambda b$

Warning: Satisfies completeness but not knowledge soundness

LatticeFold Norm Proofs [BC'24]

$$[\overline{\mathrm{cm}}_1, \bar{f}_1] \in_? R_{\mathrm{cm},\, b}$$

… …

… …

$$[\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}] \in_? R_{\mathrm{cm},\, b}$$

LatticeFold Norm Proofs [BC'24]

$$\left[\overline{\mathrm{cm}}_1, \bar{f}_1\right] \in_? R_{\mathrm{cm},\, b}$$

$$\left.\begin{array}{c} \left[\overline{\mathrm{cm}}_1, \bar{f}_1\right] \in_? R_{\mathrm{cm},\, b} \\ \dots\ \dots \\ \dots\ \dots \\ \left[\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}\right] \in_? R_{\mathrm{cm},\, b} \end{array}\right\}$$

A batched Sumcheck of degree $2b$

LatticeFold Norm Proofs [BC'24]

$$\left.\begin{array}{r} [\overline{\mathrm{cm}}_1, \bar{f}_1] \in_? R_{\mathrm{cm},\, b} \\ \dots\,\dots \\ \dots\,\dots \\ [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}] \in_? R_{\mathrm{cm},\, b} \end{array}\right\}$$

A batched Sumcheck of degree $2b$ $\xrightarrow[\text{[LFKN'90]}]{\text{Sumcheck Protocol}}$ $\left(x = [\overline{\mathrm{cm}}^*, r, v], \bar{f}^*\right) \in_? R_{\mathrm{acc},\, B}$

LatticeFold Norm Proofs [BC'24]

$[\overline{\mathrm{cm}}_1, \bar{f}_1] \in_? R_{\mathrm{cm},\, b}$

… …

… …

$[\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}] \in_? R_{\mathrm{cm},\, b}$

A batched Sumcheck of degree $2b$

Sumcheck Protocol [LFKN'90]

$\left(x = [\overline{\mathrm{cm}}^*, \ldots], \bar{f}^*\right) \in_? R_{\mathrm{acc},\, B}$

$R_{\mathrm{acc},\, B} \approx (R_{\mathrm{cm},\, B} +$ "an eval stmnt"$)$

LatticeFold Norm Proofs [BC'24]

$[\overline{\mathrm{cm}}_1, \bar{f}_1] \in_? R_{\mathrm{cm},\, b}$

… …

… …

$[\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}] \in_? R_{\mathrm{cm},\, b}$

A batched Sumcheck of degree $2b$

$\xrightarrow[\text{[LFKN'90]}]{\text{Sumcheck Protocol}}$

$\left(x = [\overline{\mathrm{cm}}^*, r, v], \bar{f}^*\right) \in_? R_{\mathrm{acc},\, B}$

Pros: Polylogarithmic verifier time

LatticeFold Norm Proofs [BC'24]

$\left.\begin{array}{l} [\overline{\mathrm{cm}}_1, \bar{f}_1] \in_? R_{\mathrm{cm},\, b} \\ \cdots\ \cdots \\ \cdots\ \cdots \\ [\overline{\mathrm{cm}}_{2k}, \bar{f}_{2k}] \in_? R_{\mathrm{cm},\, b} \end{array}\right\}$ A batched Sumcheck of degree $2b$ $\xrightarrow[\text{[LFKN'90]}]{\text{Sumcheck Protocol}}$ $\left(x = [\overline{\mathrm{cm}}^*, r, v], \bar{f}^*\right) \in_? R_{\mathrm{acc},\, B}$

Pros: Polylogarithmic verifier time

Prover overhead: Fix $B = b^k$

- Small b & Large k: Expensive computation to $\overline{\mathrm{cm}}_1, \ldots, \overline{\mathrm{cm}}_{2k}$
- Small k & Large b: Expensive high-degree Sumcheck

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

$$a \in [0, d) \cap \mathbb{Z} \iff \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

Degree = d

$$a \in [0, d) \cap \mathbb{Z} \quad \Longleftrightarrow \quad \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

Degree = d

$$a \in [0, d) \cap \mathbb{Z} \quad \Longleftrightarrow \quad \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

Over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$:

$$a \in [0, d) \cap \mathbb{Z} \quad \Longleftrightarrow$$

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

Degree = d

$$a \in [0, d) \cap \mathbb{Z} \iff \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

Over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$:

$$T(X) = \sum_{i \in [0,d)} -i \cdot X^{d-i}$$

$$a \in [0, d) \cap \mathbb{Z} \iff \exists \Delta = X^a : \ a = \mathrm{ct}(T(X) \cdot \Delta)$$

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

Degree = d

$$a \in [0, d) \cap \mathbb{Z} \quad \Longleftrightarrow \quad \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

Over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$:

$T(X) = \sum_{i \in [0,d)} -i \cdot X^{d-i}$

$$a \in [0, d) \cap \mathbb{Z} \quad \Longleftrightarrow \quad \exists \Delta = X^a : \; a = \mathrm{ct}(T(X) \cdot \Delta)$$

$-a \cdot X^{d-a} \cdot X^a = a$

An Algebraic Norm Proof over Rq

Over $\mathbb{Z}_q$:

Degree = d

$$a \in [0, d) \cap \mathbb{Z} \iff \prod_{i=[0,d)} (a - i) = 0 \bmod q$$

Over $R_q = \mathbb{Z}_q[X]/(X^d + 1)$:

$$a \in [0, d) \cap \mathbb{Z} \iff \exists \Delta = X^a : \ a = \mathrm{ct}(T(X) \cdot \Delta)$$

Degree = 1

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d \times n}$

$$\mathbb{x} = \mathrm{CM}(\ \boxed{\vec{f} \in R^n}\ )$$

P

V

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d \times n}$

E.g., want
$\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$$\mathbb{x} = \mathrm{CM}(\ \boxed{\vec{f} \in R^n}\ )$$

P

V

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d \times n}$

E.g., want $\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$\mathbb{x} = \mathrm{CM}(\ \boxed{\vec{f} \in R^n}\ )$

P

V

$\mathrm{CM}(\ \boxed{\mathrm{D} = \left(X^{F_{i,j}}\right)_{i \in [d], j \in [n]}}\ )$

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d \times n}$

E.g., want
$\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$\mathbb{x} = \mathrm{CM}(\vec{f} \in R^n)$

P

$\mathrm{CM}(\mathrm{D} = \left(X^{F_{i,j}}\right)_{i \in [d], j \in [n]})$

1 double commitment

V

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d \times n}$

E.g., want $\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$\mathbb{x} = \mathrm{CM}(\boxed{\vec{f} \in R^n})$

P

V

$\mathrm{CM}(\boxed{\mathrm{D} = \left(X^{F_{i,j}}\right)_{i \in [d], j \in [n]}})$

Constant check

$\forall i \in [d], j \in [n] : \mathrm{F}_{i,j} = \mathrm{ct}(T(X) \cdot \mathrm{D}_{i,j})$

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d\times n}$

E.g., want $\mathrm{F}_{i,j} \in [0, d)\forall i, j$

$\mathbb{x} = \mathrm{CM}(\ \vec{f} \in R^n\ )$

P

V

$\mathrm{CM}(\ \mathrm{D} = \left(X^{\mathrm{F}_{i,j}}\right)_{i\in[d], j\in[n]}\ )$

Constant check

$\forall i \in [d], j \in [n] :\ \mathrm{F}_{i,j} = \mathrm{ct}(T(X) \cdot \mathrm{D}_{i,j})$

Monomial check

$\forall i \in [d], j \in [n] : \mathrm{D}_{i,j} \in \{0,1, X, \dots, X^{d-1}\}$

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d\times n}$

E.g., want

$\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$$\mathbb{x} = \mathrm{CM}(\ \boxed{\vec{f} \in R^n}\ )$$

P

V

$$\mathrm{CM}(\ \boxed{\mathrm{D} = \left(X^{\mathrm{F}_{i,j}}\right)_{i\in[d], j\in[n]}}\ )$$

Constant check

$$\forall i \in [d], j \in [n]: \ \mathrm{F}_{i,j} = \mathrm{ct}(T(X) \cdot \mathrm{D}_{i,j})$$

Monomial check

$$\forall i \in [d], j \in [n]: \mathrm{D}_{i,j} \in \{0,1, X, \dots, X^{d-1}\}$$

A (batched) deg-3 Sumcheck over $\mathbb{Z}_q$

Batch Range Proof

$\mathrm{Coef}(\vec{f}) = \mathrm{F} \in \mathbb{Z}^{d\times n}$

E.g., want
$\mathrm{F}_{i,j} \in [0, d) \forall i, j$

$\mathbb{x} = \mathrm{CM}(\boxed{\vec{f} \in R^n})$

P

V

$\mathrm{CM}(\boxed{\mathrm{D} = \left(X^{\mathrm{F}_{i,j}}\right)_{i\in[d], j\in[n]}})$

Constant check

$\forall i \in [d], j \in [n] : \ \mathrm{F}_{i,j} = \mathrm{ct}(T(X) \cdot \mathrm{D}_{i,j})$

Monomial check

$\forall i \in [d], j \in [n] : \mathrm{D}_{i,j} \in \{0,1, X, \dots, X^{d-1}\}$

A (batched) deg-3 Sumcheck over $\mathbb{Z}_q$

$\downarrow$

$(\mathbb{x}' = [\overline{\mathrm{cm}}^*, r, v], \bar{f}^*) \in_? R_{\mathrm{acc}}$

Extensions & Summary

The actual scheme:

- Support larger norm ranges than $[0, d)$
- Support smaller modulus q via tensor-of-rings [NS'25, BC'25]
- Folding committed R1CS relations

Extensions & Summary

The actual scheme:

- Support larger norm ranges than $[0, d)$
- Support smaller modulus q via tensor-of-rings [NS'25, BC'25]
- Folding committed R1CS relations

Summary

- Lattice-based folding is easier to build than lattice SNARKs
- Faster & parallelizable provers + succinct verifiers
- Core technique: A new lattice-based algebraic range proof

Future Work

Non-power-of-2 cyclotomic rings?

Non-prime modulus q?

Efficient ℓ_2-norm range proofs?

QROM analysis?

THANK YOU

Eprint:2025/247