DEEP Commitments

and their Applications

Alan Szepieniec
alan@neptune.cash

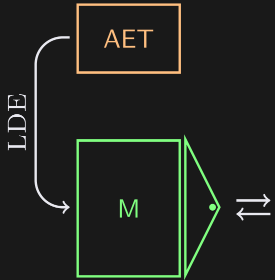
neptune.cash

triton-vm.org

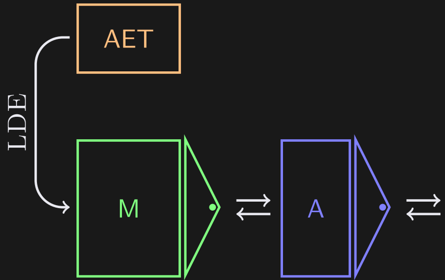
STARK Workflow with DEEP ALI

AET

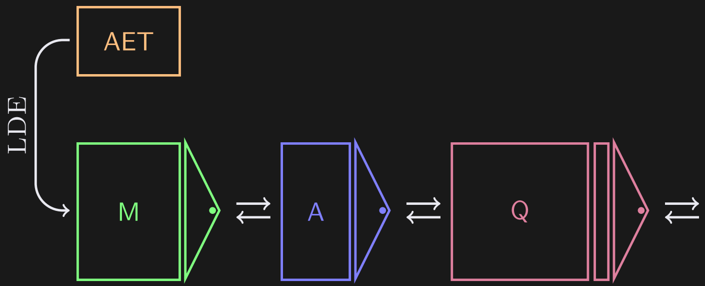
STARK Workflow with DEEP ALI



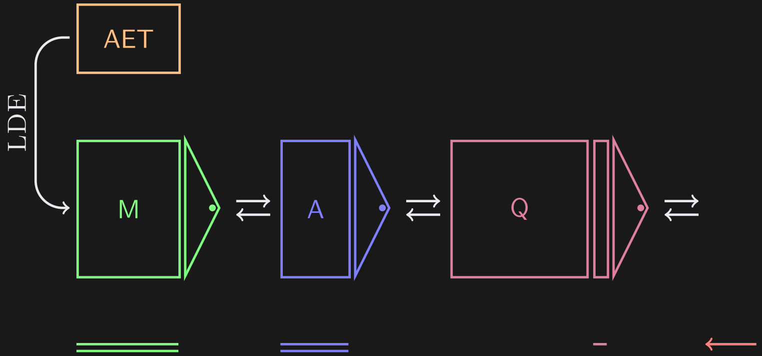
STARK Workflow with DEEP ALI



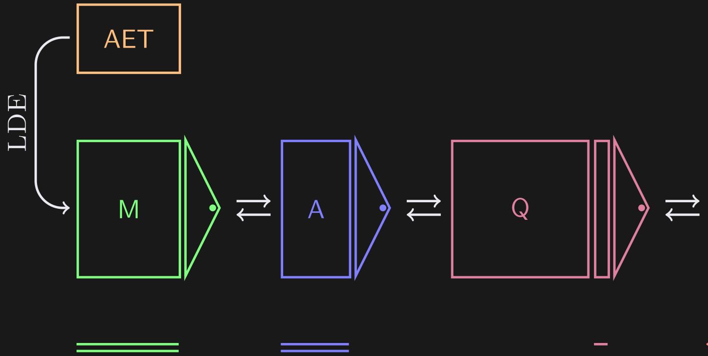
STARK Workflow with DEEP ALI



STARK Workflow with DEEP ALI

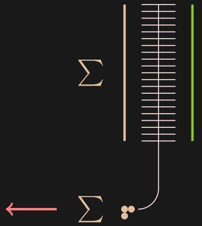


STARK Workflow with DEEP ALI

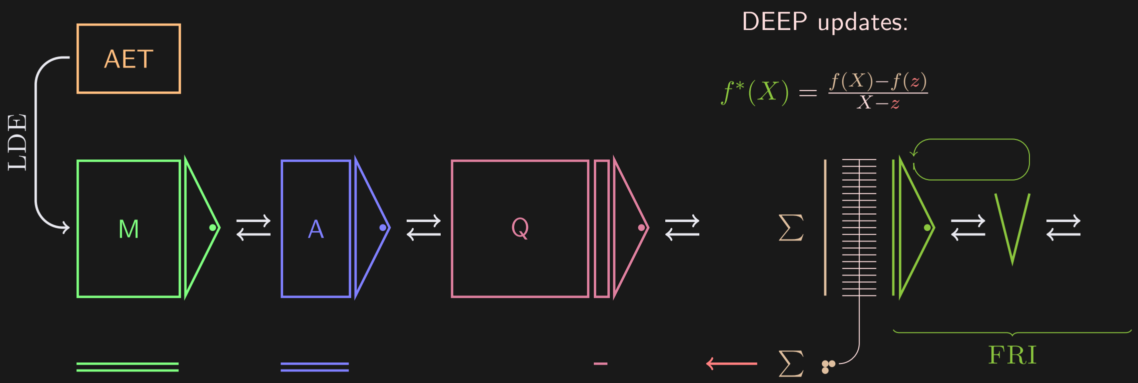


DEEP updates:

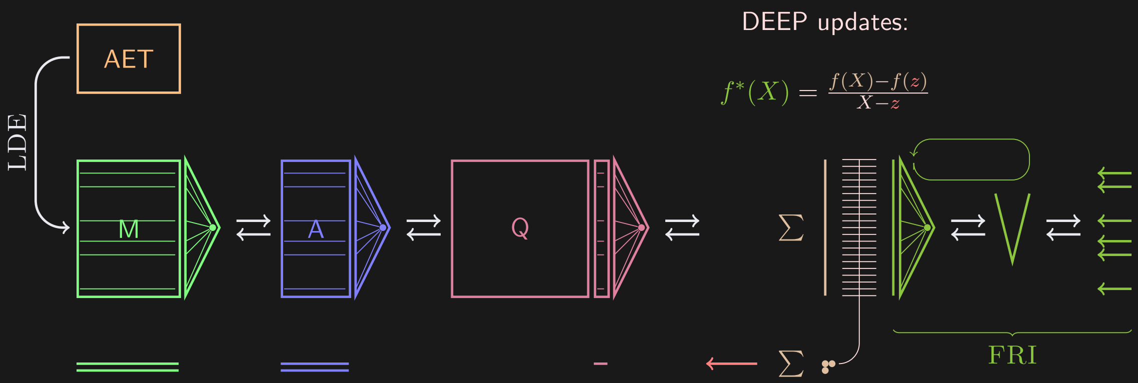
$$f^*(X) = \frac{f(X) - f(z)}{X - z}$$



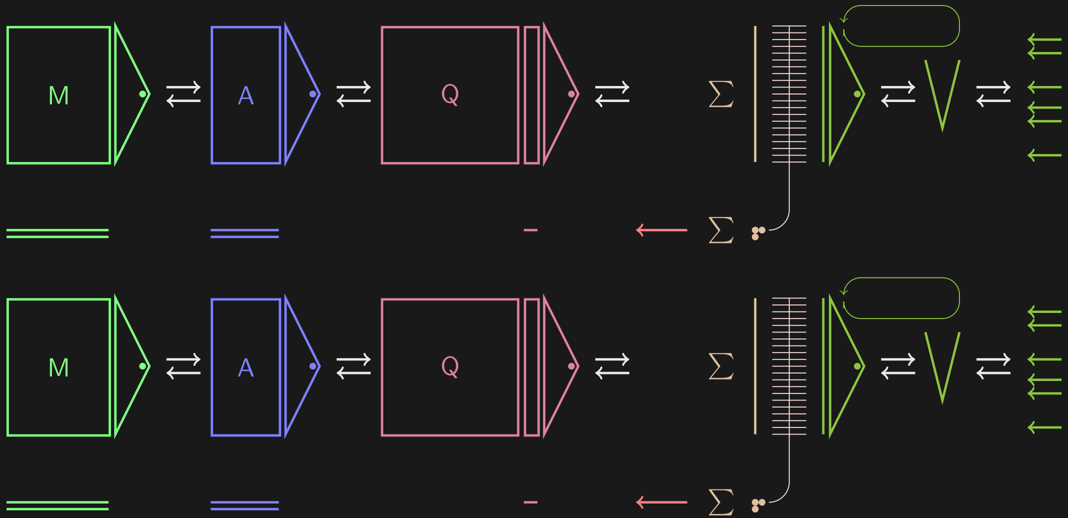
STARK Workflow with DEEP ALI



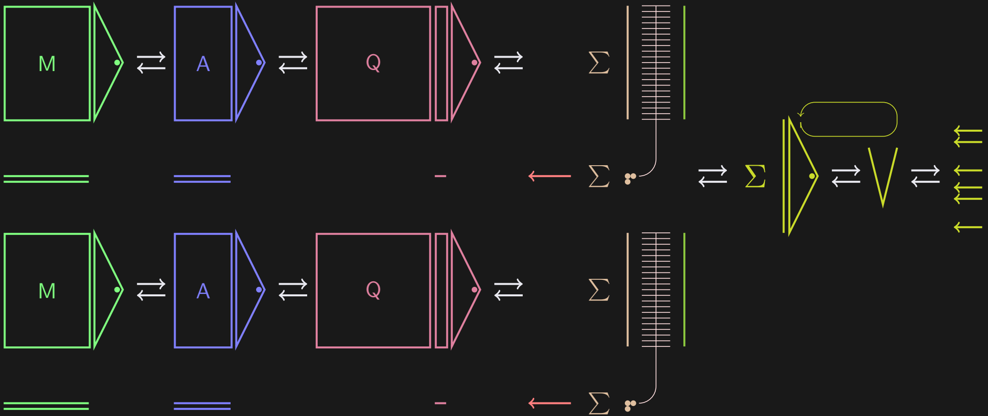
STARK Workflow with DEEP ALI



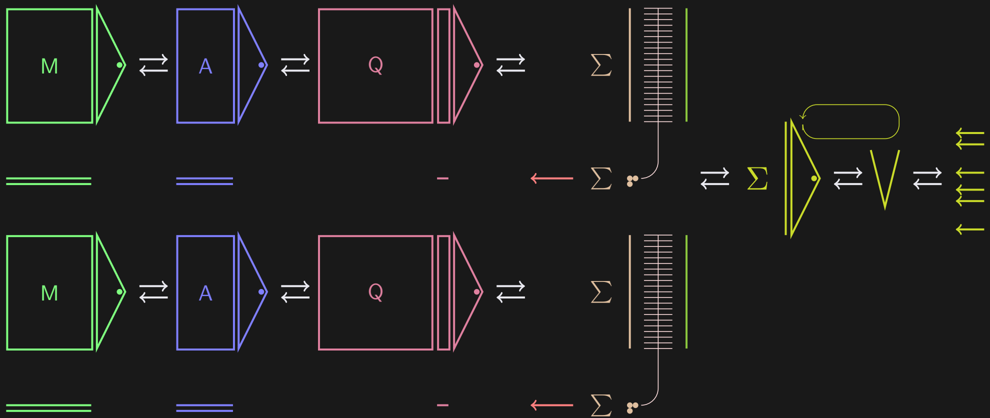
Challenge



Hope: Amortize FRI



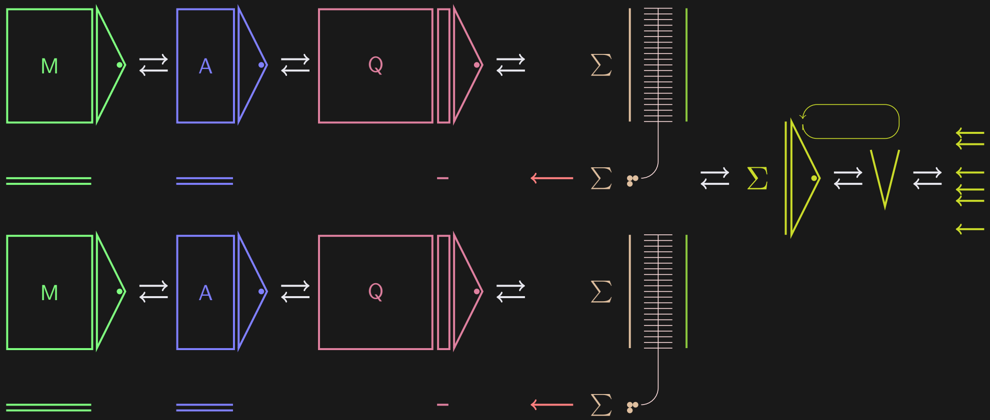
Hope: Amortize FRI



Obstacle 1:

Obstacle 2:

Hope: Amortize FRI



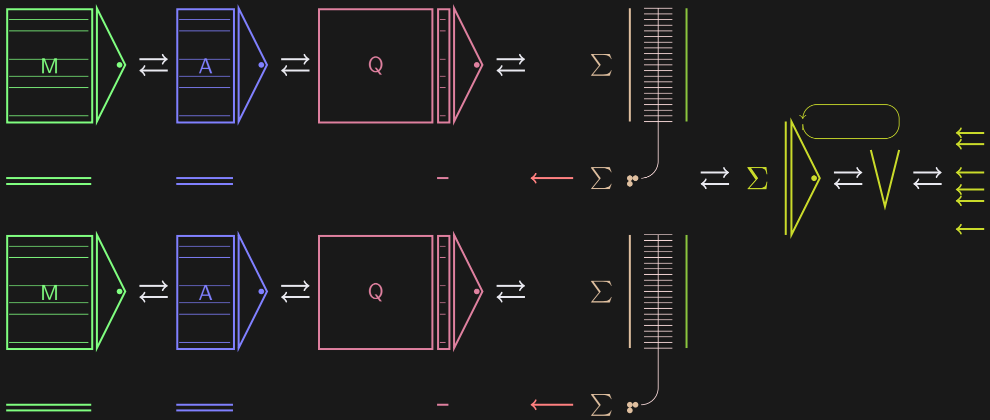
Obstacle 1:

one index set dependent on *both* transcripts

$\Longrightarrow$ prove simultaneously

Obstacle 2:

Hope: Amortize FRI



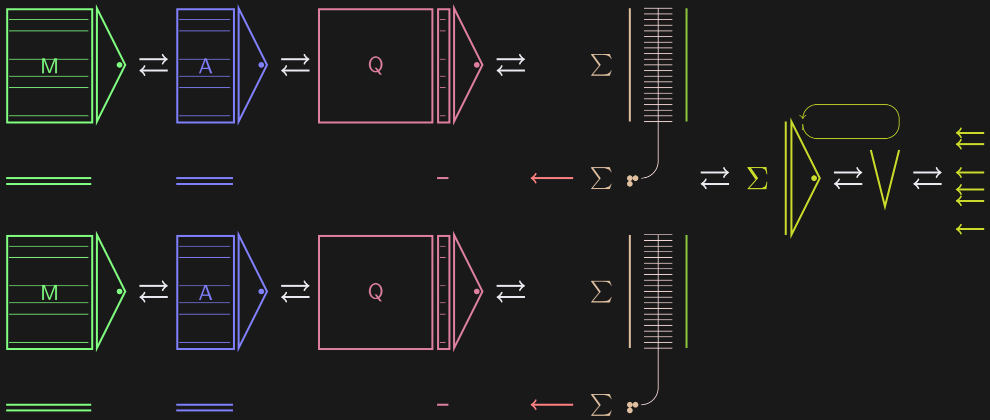
OBSTACLE 1:

one index set dependent on *both* transcripts

$\Longrightarrow$ prove simultaneously

OBSTACLE 2:

Hope: Amortize FRI



Obstacle 1:

one index set dependent on *both* transcripts

$\Longrightarrow$ prove simultaneously

Obstacle 2:

store *two* AETs

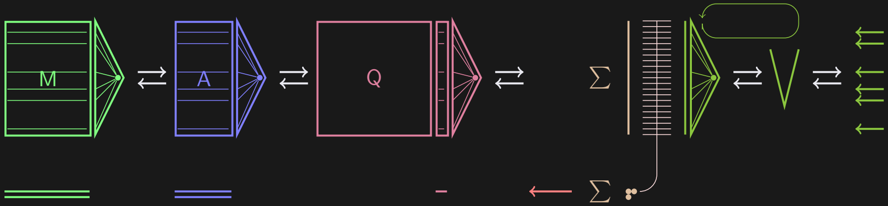
$\longrightarrow$ prohibitive RAM cost

Obstacle 1

$\rightarrow$ split protocol into two parts:

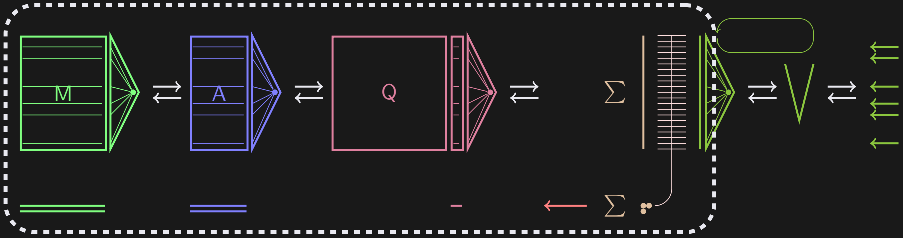
Obstacle 1

→ split protocol into two parts:



Obstacle 1

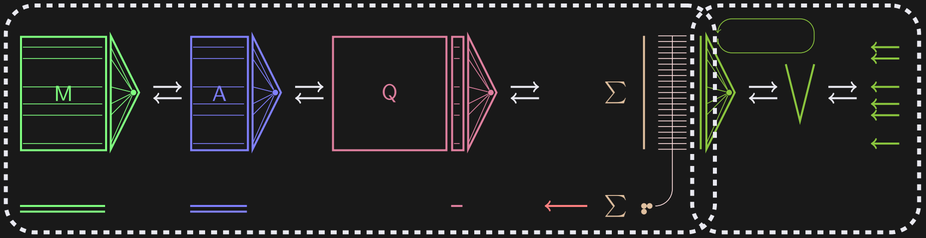
$\longrightarrow$ split protocol into two parts:



DEEP-ALI: reduce AET $\mapsto$ Merkle root

Obstacle 1

→ split protocol into two parts:

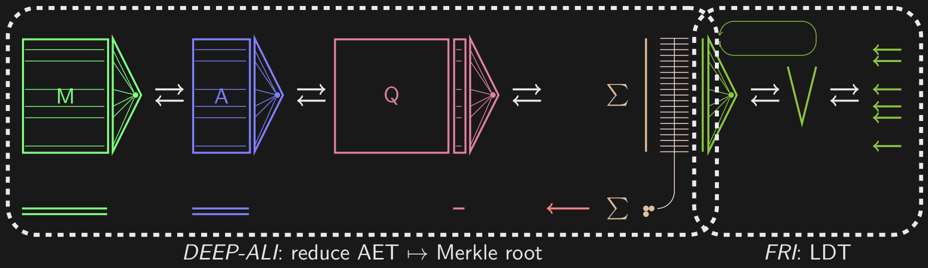


DEEP-ALI: reduce AET ↦ Merkle root

FRI: LDT

Obstacle 1

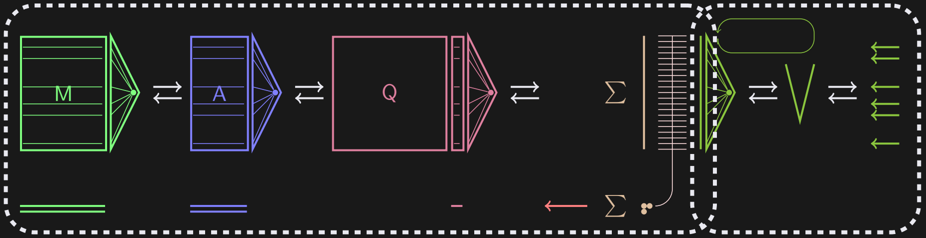
→ split protocol into two parts:



- reduce many AETs to Merkle roots
- concatenate and hash to sample *weights* and *index set*
- run one instance of FRI

Obstacle 1

→ split protocol into two parts:



DEEP-ALI: reduce AET ↦ Merkle root

FRI: LDT

– reduce many AETs to Merkle roots ← *any order* ✓

– concatenate and hash to sample *weights* and *index set*

– run one instance of FRI ← "simultaneous"

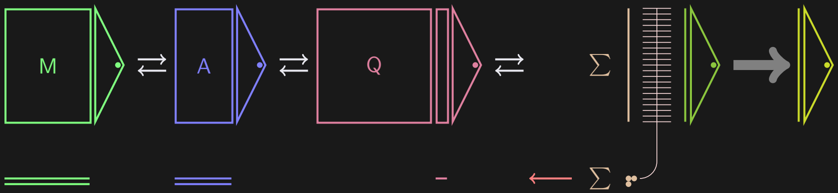
RAM cost remains! × × × ×

Obstacle 2

$\rightarrow$ decouple codeword from history

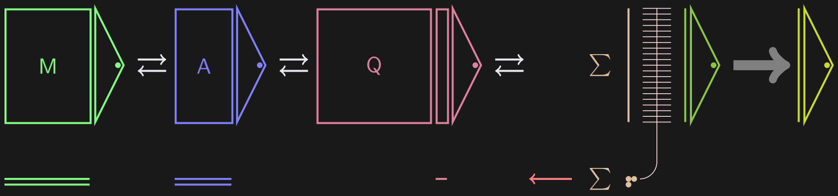
Obstacle 2

→ decouple codeword from history



Obstacle 2

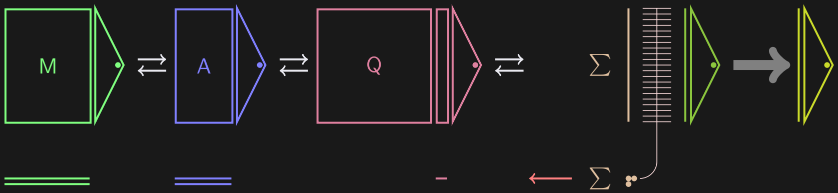
$\rightarrow$ decouple codeword from history



- verify authentic reduction to history-independent codeword
- drop AET from RAM
- batch history-independent codewords for FRI

Obstacle 2

→ decouple codeword from history



– verify authentic reduction to history-independent codeword

– drop AET from RAM

– batch history-independent codewords for FRI

HOW?

Decouple

$\mathcal{P}$ $h(X)$

$\mathcal{V}$ $[h(X)]$

Decouple

$\mathcal{P}$ $h(X)$

$\mathcal{V}$ $[h(X)]$

$z \xleftarrow{\$} \mathbb{F} \backslash D$

$\xleftarrow{\quad z \quad}$

Decouple

$\mathcal{P}$ $h(X)$ $\qquad\qquad$ $\mathcal{V}$ $[h(X)]$

$z \xleftarrow{\$} \mathbb{F} \backslash D$

$\xleftarrow{\quad z \quad}$

$y = h(\alpha)$

$\xrightarrow{\quad y \quad}$

Decouple

$\mathcal{P}$ $h(X)$ $\qquad\qquad$ $\mathcal{V}$ $[h(X)]$

$z \xleftarrow{\$} \mathbb{F} \backslash D$

$\xleftarrow{\quad z \quad}$

$y = h(\alpha)$

$\xrightarrow{\quad y \quad}$

$x_0, \ldots, x_{t-1} \xleftarrow{\$} D$

$\gamma \xleftarrow{\$} \mathbb{F}$

$\xleftarrow{\quad x_0, \ldots, x_{t-1}, \gamma \quad}$

Decouple

$\mathcal{P}$ $h(X)$ $\qquad\qquad$ $\mathcal{V}$ $[h(X)]$

$z \xleftarrow{\$} \mathbb{F} \setminus D$

$\xleftarrow{\quad z \quad}$

$y = h(\alpha)$

$\xrightarrow{\quad y \quad}$

$x_0, \ldots, x_{t-1} \xleftarrow{\$} D$

$\gamma \xleftarrow{\$} \mathbb{F}$

$\xleftarrow{\quad x_0, \ldots, x_{t-1}, \gamma \quad}$

$$q(X) = \underbrace{\frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1}}_{\text{degree-correction}} \cdot \frac{h(X) - \mathsf{Ans}(X)}{Z(X)}$$

Decouple

$\mathcal{P}$ $h(X)$ $\mathcal{V}$ $[h(X)]$

$z \xleftarrow{\$} \mathbb{F} \backslash D$

$\xleftarrow{\quad z \quad}$

$y = h(\alpha)$

$\xrightarrow{\quad y \quad}$

$x_0, \ldots, x_{t-1} \xleftarrow{\$} D$

$\gamma \xleftarrow{\$} \mathbb{F}$

$\xleftarrow{\quad x_0, \ldots, x_{t-1}, \gamma \quad}$

$\xrightarrow{\quad [\text{Fill}(X)] \quad}$

$$q(X) = \underbrace{\frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1}}_{\text{degree-correction}} \cdot \frac{h(X) - \text{Ans}(X)}{Z(X)}$$

$$[q(X)] = \begin{cases} \frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1} \cdot \frac{[h(X)] - \text{Ans}(X)}{Z(X)} & \Leftarrow X \notin \{x_i\} \\ [\text{Fill}(X)] & \Leftarrow X \in \{x_i\} \end{cases}$$

Soundness

$\sum f_j(X)$

true batch sum

$h(X)$

claimed batch sum

$q(X)$

result

$\Delta(\sum f_j(X), \mathsf{RS}) > \delta$

$\Delta(q(X), \mathsf{RS}) < \delta$

Soundness

$\sum f_j(X)$
true batch sum

$h(X)$
claimed batch sum

$q(X)$
result

$$\Delta\left(\sum f_j(X), \mathsf{RS}\right) > \delta$$

$$\Delta\left(q(X), \mathsf{RS}\right) < \delta$$

$$\Delta\left(h(X), \mathsf{RS}\right) > \delta$$

$$\Delta\left(h(X), \mathsf{RS}\right) < \delta$$

Soundness

$\sum f_j(X)$ | $h(X)$ | $q(X)$
true batch sum | claimed batch sum | result

$\Delta(\sum f_j(X), \mathsf{RS}) > \delta$

$\Delta\big(q(X), \mathsf{RS}\big) < \delta$

$\Delta\big(h(X), \mathsf{RS}\big) > \delta$ $\leftarrow$ trivial: by construction up to t points from $\mathsf{Fill}(X)$

$\Delta\big(h(X), \mathsf{RS}\big) < \delta$ $\leftarrow$ tricky

Soundness

$\sum f_j(X)$ true batch sum

$h(X)$ claimed batch sum

$q(X)$ result

$\Delta\left(\sum f_j(X), \mathsf{RS}\right) > \delta$

$\Delta\left(q(X), \mathsf{RS}\right) < \delta$

$\Delta\left(h(X), \mathsf{RS}\right) > \delta$ $\leftarrow$ trivial: by construction up to t points from $\mathsf{Fill}(X)$

$\Delta\left(h(X), \mathsf{RS}\right) < \delta$ $\leftarrow$ tricky

$p(X) \in \mathsf{RS}$

s.t. $\Delta\left(h(X), p(X)\right) < \delta$

and $p(z) = y$

Soundness

$\sum f_j(X)$ true batch sum

$h(X)$ claimed batch sum

$q(X)$ result

$\Delta(\sum f_j(X), \mathsf{RS}) > \delta$

$\Delta(q(X), \mathsf{RS}) < \delta$

$\Delta(h(X), \mathsf{RS}) > \delta$ ← trivial: by construction up to t points from $\mathsf{Fill}(X)$

$\Delta(h(X), \mathsf{RS}) < \delta$ ← tricky

$p(X) \in \mathsf{RS}$

s.t. $\Delta(h(X), p(X)) < \delta$

and $p(z) = y$

$\nexists\, p(X)$ ← trivial: quotient by wrong interpolant

Soundness

$\sum f_j(X)$ — true batch sum

$h(X)$ — claimed batch sum

$q(X)$ — result

$\Delta(\sum f_j(X), \mathsf{RS}) > \delta$

$\Delta(q(X), \mathsf{RS}) < \delta$

$\Delta(h(X), \mathsf{RS}) > \delta$ ← trivial: by construction up to t points from $\mathsf{Fill}(X)$

$\Delta(h(X), \mathsf{RS}) < \delta$ ← tricky

$p(X) \in \mathsf{RS}$

s.t. $\Delta(h(X), p(X)) < \delta$

and $p(z) = y$

$\nexists\, p(X)$ ← trivial: quotient by wrong interpolant

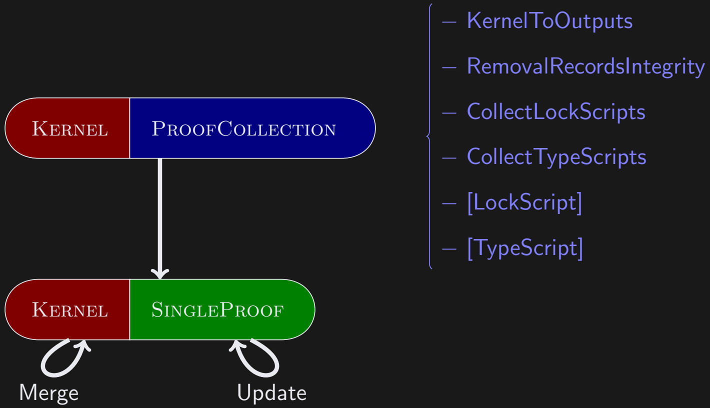
$\exists\, p(X)$ ← then:

- $p(X)$ is unique whp (DEEP technique)
- since $\Delta(\sum f_i(X), p(X)) > \delta$, wrong in-domain points $\{(x_i, \sum f_j(x_i))\}$ whp
- $\Rightarrow$ quotient by wrong interpolant $\square$

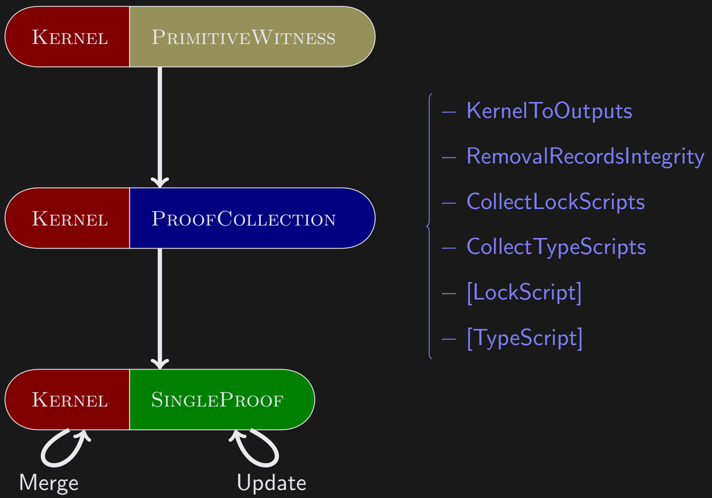
Application: Transactions in Neptune

Application: Transactions in Neptune

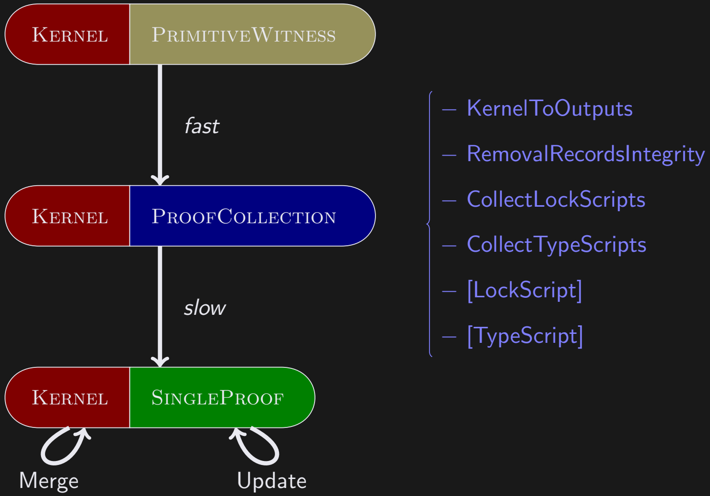
Application: Transactions in Neptune



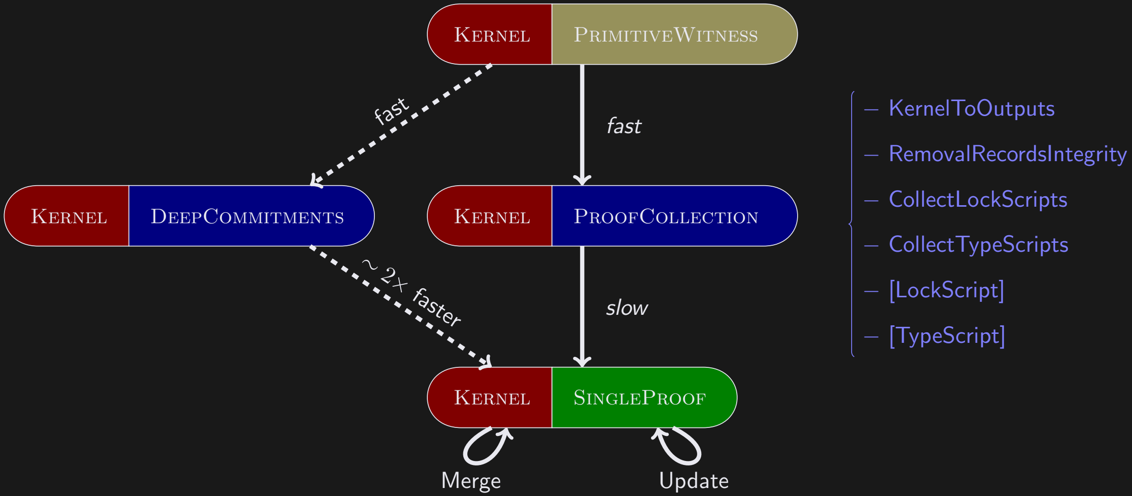
Application: Transactions in Neptune



Application: Transactions in Neptune

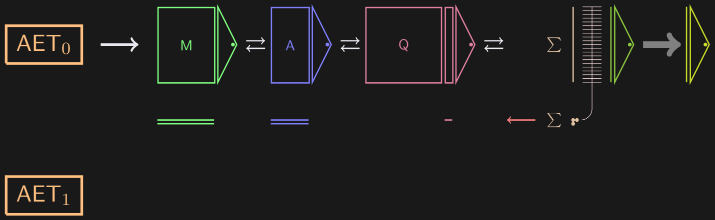


Application: Transactions in Neptune

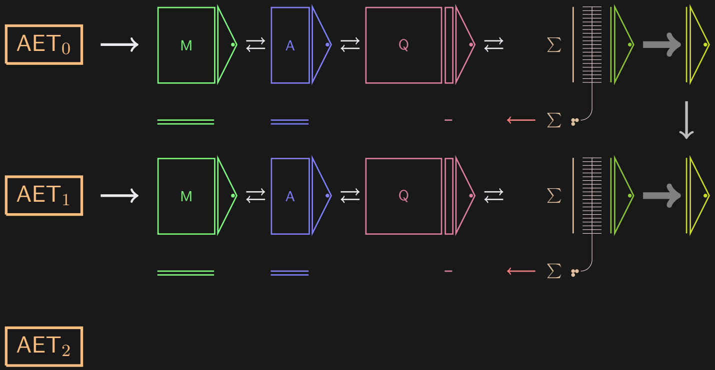


Application: Incrementally Verifiable Computation (IVC)

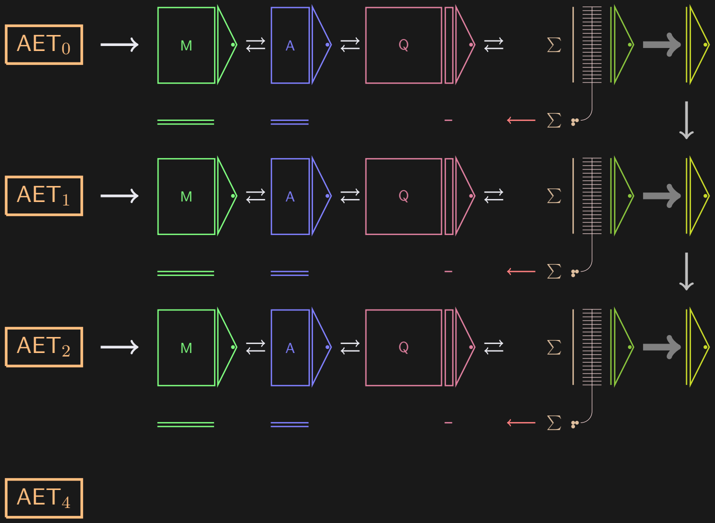
Application: Incrementally Verifiable Computation (IVC)



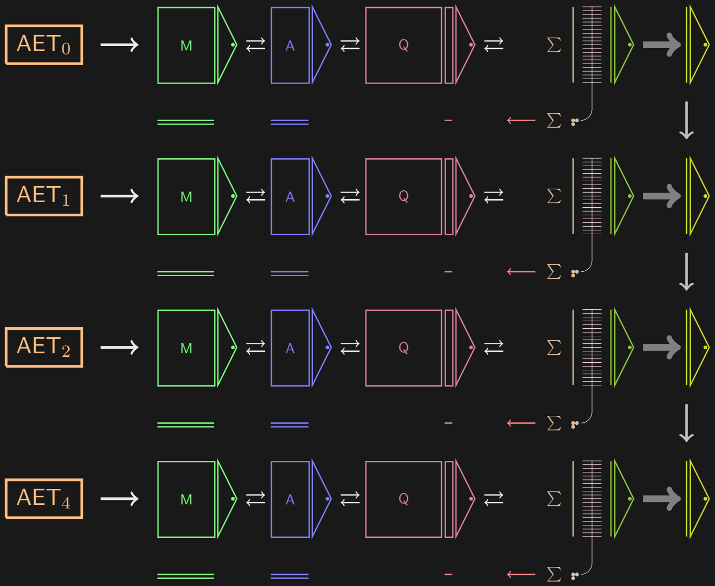
Application: Incrementally Verifiable Computation (IVC)



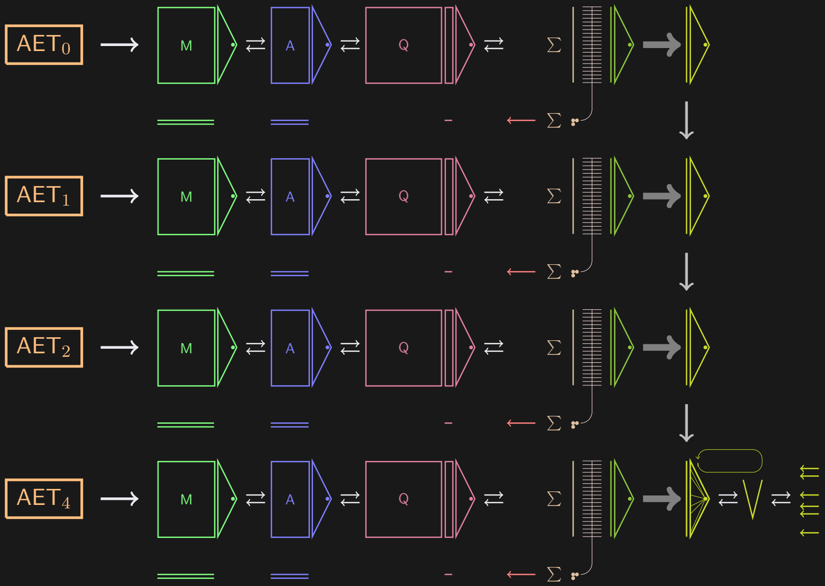
Application: Incrementally Verifiable Computation (IVC)



Application: Incrementally Verifiable Computation (IVC)



Application: Incrementally Verifiable Computation (IVC)



Application: Better LDT

Application: Better LDT

0. decouple

Application: Better LDT

0. decouple

1. shift domains $D \to D'$

Application: Better LDT

0. decouple

1. shift domains $D \to D'$

2. shrink rate faster than domain size

Application: Better LDT

$0.$ decouple

$1.$ shift domains $D \to D'$

$2.$ shrink rate faster than domain size

$\Rightarrow$ fewer queries

Application: Better LDT

0. decouple

1. shift domains $D \to D'$

2. shrink rate faster than domain size

   $\Rightarrow$ fewer queries