D —— **dynamic reduction** ——→ D_Q

online

$Q(D)$ ←—— **approximation** ——— $Q(D_Q), \eta$

Bounded Approximation Scheme

Input: A resource ratio $\alpha \in (0, 1]$ and an access schema $\mathcal{A}$.

Scheme[6]: Given generic SQL Q and D, computes $(Q(D_Q), \eta)$:

- access a fraction D_Q of D with $|D_Q| \leq \alpha |D|$;

- $\textbf{accuracy}(Q, D, Q(D_Q)) \geq \eta$.

Chellenges:

- **Deterministic** bound for **generic** queries (even non-aggregate)
- Compute **both** answers and accuracy bound by accessing $\leq \alpha|D|$ tuples

decided by available resources

approximate answers

accuracy bound

<u>Flexible trade-offs</u>: available resources vs. accuracy bound

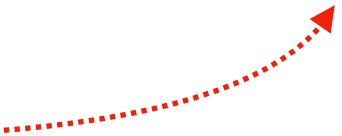[6] Y. Cao, W. Fan: Data Driven Approximation with Bounded Resources, VLDB 2017

not *one-size-fit-all* reduction

access $\leq \alpha|D|$ tuples

Bounded Approximation Scheme
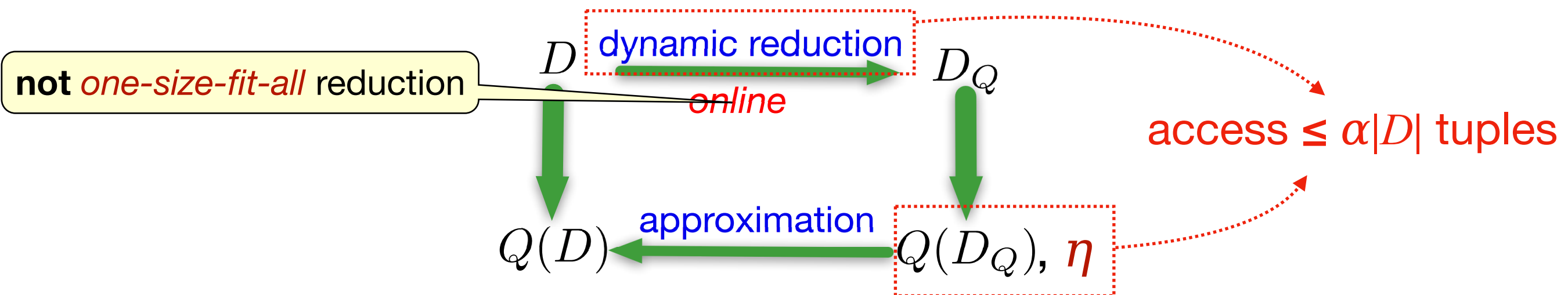
approximate answers

Input: A resource ratio $\alpha \in (0, 1]$ and an access schema $\mathcal{A}$.

Scheme[6]: Given generic SQL Q and D, computes $(Q(D_Q), \eta)$:

- access a fraction D_Q of D with $|D_Q| \leq \alpha|D|$;

  accuracy bound

- $\text{accuracy}(Q, D, Q(D_Q)) \geq \eta$.

  deterministic!

not *one-size-fit-all* reduction

dynamic reduction

D dynamic reduction D_{Q^2}

online

access $\leq \alpha|D|$ tuples

approximation

approximation

$Q(D)$ $Q(D_Q)$ η

Flexible trade-offs: available resources vs. accuracy bound

Chellenges:

- **Deterministic** bound for **generic** queries (even non-aggregate)

- Compute **both** answers and accuracy bound by accessing $\leq \alpha|D|$ tuples

[6] Y. Cao, W. Fan: Data Driven Approximation with Bounded Resources, VLDB 2017

How does it work?