Towards Output-Optimal Uniform Sampling and Approximate Counting for Join-Project Queries

Xiao Hu[1] Jinchao Huang[2]

[1]University of Waterloo [2]The Chinese University of Hong Kong

PODS 2026 June 2026 ▪ Bengaluru, India

Why Sampling for Join-Project Queries?

Modern analytics queries are not bare joins. They look like

$$\pi_{\mathbf{y}} \left(R_1 \bowtie R_2 \bowtie \cdots \bowtie R_k \right).$$

Many applications need a *sample*, not the full result

- Approximate query processing & visualization
- Cardinality / selectivity estimation for the optimizer
- Training ML models without materializing the join
- Interactive exploration of warehouse-scale data

Prior breakthroughs: optimal sampling for **full joins** (Zhao'18; Deng-Lu-Tao'23; Kim et al.'23).

Reality: real queries **nearly always** project away non-output attributes.

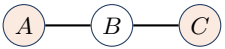
Join-Project Queries

Definition

$\mathcal{Q} = (\mathcal{V}, \mathcal{E}, \mathbf{y})$ with output attributes $\mathbf{y} \subseteq \mathcal{V}$; $\quad \mathcal{Q}(\mathcal{R}) \;=\; \pi_{\mathbf{y}}\Big(\bowtie_{e\in\mathcal{E}} R_e\Big)$.

Matrix $\mathcal{Q}_{\text{matrix}}$

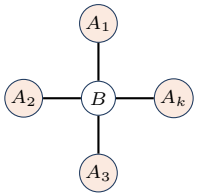
$\pi_{A,C}\big(R_1(A,B) \bowtie R_2(B,C)\big)$



red = output attr.

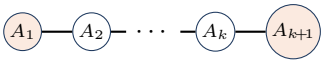
k-**Star** $\mathcal{Q}_{\text{star}}$

$\pi_{A_1,\ldots,A_k}\big(\bowtie_i R_i(A_i, B)\big)$



k-**Chain** $\mathcal{Q}_{\text{chain}}$

$\pi_{A_1,A_{k+1}}\big(\bowtie_i R_i(A_i, A_{i+1})\big)$



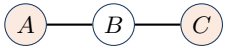
Notation: N = input size; $\mathrm{OUT} = |\mathcal{Q}(\mathcal{R})|$; $\mathrm{OUT}_{\bowtie}$ = full-join size; ρ^* = fractional edge-cover number (of the underlying full join query).

Prior Framework: "Propose-and-Verify" [Chen-Yi'20]

1. **Propose:** draw a candidate t from the *projection-induced full join* $\pi_{\mathbf{y}}(R_{e_1}) \bowtie \pi_{\mathbf{y}}(R_{e_2}) \bowtie \cdots$
2. **Verify:** check whether t extends to a real full-join tuple.

Prior Framework: "Propose-and-Verify" [Chen-Yi'20]

1. **Propose:** draw a candidate t from the *projection-induced full join* $\pi_{\mathbf{y}}(R_{e_1}) \bowtie \pi_{\mathbf{y}}(R_{e_2}) \bowtie \cdots$
2. **Verify:** check whether t extends to a real full-join tuple.



Two fundamental bottlenecks

1. **Low acceptance probability:** the projection-induced join is much larger than OUT.
2. **High verification cost:** essentially evaluating the *residual join* on non-output attrs.

Resulting sampling complexity: $\tilde{O}\left(\dfrac{N^{\rho^*}}{\mathrm{OUT}}\right)$.

Specializing:

$$\mathcal{Q}_{\mathsf{matrix}}:\ \tilde{O}(N^2/\mathrm{OUT}), \quad \mathcal{Q}_{\mathsf{star}}:\ \tilde{O}(N^k/\mathrm{OUT}), \quad \mathcal{Q}_{\mathsf{chain}}:\ \tilde{O}\left(N^{\lceil (k+1)/2 \rceil}/\mathrm{OUT}\right).$$

Are these optimal? — No.

Our Results at a Glance

		Matrix	k-Star	k-Chain	
				$k = 3$	$k \geq 4$
Uniform Sampling	Prior	$\tilde{O}(N^2/\mathrm{OUT})$	$\tilde{O}(N^k/\mathrm{OUT})$	$\tilde{O}(N^2/\mathrm{OUT})$	$\tilde{O}(N^{\lceil \frac{k+1}{2} \rceil}/\mathrm{OUT})$
	Our UB	$\tilde{O}(N/\sqrt{\mathrm{OUT}})$	$\tilde{O}(N/\mathrm{OUT}^{1/k})$	$\tilde{O}(\min\{N, N^2/\mathrm{OUT}\})$	$\tilde{O}(N)$
	Our LB	$\tilde{\Omega}(N/\sqrt{\mathrm{OUT}})$	$\tilde{\Omega}(N/\mathrm{OUT}^{1/k})$	$\tilde{\Omega}(\min\{N, N^2/\mathrm{OUT}\})$	$\tilde{\Omega}(N)$
Approx. Counting	Prior	$\tilde{O}(N^2/\mathrm{OUT})$	$\tilde{O}(N^k/\mathrm{OUT})$	$\tilde{O}(N^2/\mathrm{OUT})$	$\tilde{O}(N^{\lceil \frac{k+1}{2} \rceil}/\mathrm{OUT})$
	Our UB	$\tilde{O}(N/\sqrt{\mathrm{OUT}})$	$\tilde{O}(N/\mathrm{OUT}^{1/k})$	$\tilde{O}(\min\{N, N^2/\mathrm{OUT}\})$	$\tilde{O}(N)$
	Our LB	$\tilde{\Omega}(N/\sqrt{\mathrm{OUT}})$	$\tilde{\Omega}(N/\mathrm{OUT}^{1/k})$	$\tilde{\Omega}(\min\{N, N^2/\mathrm{OUT}\})$	$\tilde{\Omega}(N)$

- **Polynomial** speed-ups across all three query classes.
- Counting LBs via communication complexity — rule out *any* algorithm, incl. fast matrix mult.

Matrix Query: Two-Step Framework

$$\mathcal{Q}_{\text{matrix}} \;=\; \pi_{A,C}\big(R_1(A,B) \bowtie R_2(B,C)\big)$$

SampleMatrix — repeat until acceptance

1. **Step 1.** Draw $(a,b,c) \in R_1 \bowtie R_2$ uniformly at random.
2. **Step 2.** Accept (a,c) as the final sample with probability $\dfrac{1}{\deg(a,c)}$, where $\deg(a,c) = |\pi_B(R_1 \ltimes a) \cap \pi_B(R_2 \ltimes c)|$.

Matrix Query: Two-Step Framework

$$\mathcal{Q}_{\text{matrix}} \;=\; \pi_{A,C}\big(R_1(A,B) \bowtie R_2(B,C)\big)$$

SampleMatrix — repeat until acceptance

1. **Step 1.** Draw $(a,b,c) \in R_1 \bowtie R_2$ uniformly at random.
2. **Step 2.** Accept (a,c) as the final sample with probability $\dfrac{1}{\deg(a,c)}$,

   where $\deg(a,c) = |\pi_B(R_1 \ltimes a) \cap \pi_B(R_2 \ltimes c)|$.

Why uniform? A query result (a,c) appears in the full-join sample with probability $\frac{\deg(a,c)}{\mathrm{OUT}_\bowtie}$, hence

$$\mathbf{Pr}[\text{output } (a,c)] = \underbrace{\frac{\deg(a,c)}{\mathrm{OUT}_\bowtie}}_{\text{Step 1}} \cdot \underbrace{\frac{1}{\deg(a,c)}}_{\text{Step 2}} = \frac{1}{\mathrm{OUT}_\bowtie} \quad \text{— independent of } (a,c). \quad \checkmark$$

Matrix Query: Two-Step Framework

$$\mathcal{Q}_{\text{matrix}} \;=\; \pi_{A,C}\big(R_1(A,B) \bowtie R_2(B,C)\big)$$

SampleMatrix — repeat until acceptance

1. **Step 1.** Draw $(a,b,c) \in R_1 \bowtie R_2$ uniformly at random.
2. **Step 2.** Accept (a,c) as the final sample with probability $\dfrac{1}{\deg(a,c)}$,

   where $\deg(a,c) = |\pi_B(R_1 \ltimes a) \cap \pi_B(R_2 \ltimes c)|$.

Why uniform? A query result (a,c) appears in the full-join sample with probability $\frac{\deg(a,c)}{\mathrm{OUT}_\bowtie}$, hence

$$\mathbf{Pr}[\text{output } (a,c)] = \underbrace{\frac{\deg(a,c)}{\mathrm{OUT}_\bowtie}}_{\text{Step 1}} \cdot \underbrace{\frac{1}{\deg(a,c)}}_{\text{Step 2}} = \frac{1}{\mathrm{OUT}_\bowtie} \quad \text{— independent of } (a,c). \quad \checkmark$$

(i) Step 1: Easy. (ii) Step 2 *without* computing $\deg(a,c)$? The challenge.

Step 1: Sampling a Full-Join Tuple in $O(1)$

Auxiliary index (built in $O(N)$): for each $b \in \mathsf{adom}(B)$, store

$$W_b = |R_1 \ltimes b| \cdot |R_2 \ltimes b|, \qquad W = \sum_{b \in \mathsf{adom}(B)} W_b = \mathrm{OUT}_{\bowtie}.$$

JoinSampleMatrix-H

1. Sample $b \in \mathsf{adom}(B)$ with probability W_b/W.
2. $a \leftarrow$ random A-value in $R_1 \ltimes b$.
3. $c \leftarrow$ random C-value in $R_2 \ltimes b$.
4. Return (a, b, c).

Correctness:

$$\mathbf{Pr}[(a,b,c)] = \frac{W_b}{W} \cdot \frac{1}{|R_1 \ltimes b|} \cdot \frac{1}{|R_2 \ltimes b|} = \frac{1}{W} = \frac{1}{\mathrm{OUT}_{\bowtie}}.$$

$\Rightarrow$ uniform sample in $O(1)$ **time**. ✓

Step 2: Acceptance *without* Computing $\deg(a, c)$

Goal: accept (a, c) with probability $1/\deg(a, c)$ without enumerating all common B-witnesses.

Simulation Lemma

If $Y \in [M]$ and U is uniform on $[M]$, then

$$\mathbf{Pr}[U \leq Y] = \frac{\mathbf{E}[Y]}{M}.$$

Step 2: Acceptance *without* Computing $\deg(a, c)$

Goal: accept (a, c) with probability $1/\deg(a, c)$ without enumerating all common B-witnesses.

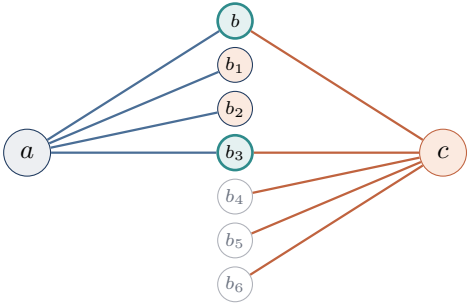
Simulation Lemma

If $Y \in [M]$ and U is uniform on $[M]$, then

$$\mathbf{Pr}[U \leq Y] = \frac{\mathbf{E}[Y]}{M}.$$

Constructing Proxy r.v.

Sample w/o replacement the smaller neighbor list until the next overlap or exhaust. Let F be the number of non-overlap values before stop.



Step 2: Acceptance *without* Computing $\deg(a, c)$

Goal: accept (a, c) with probability $1/\deg(a, c)$ without enumerating all common B-witnesses.

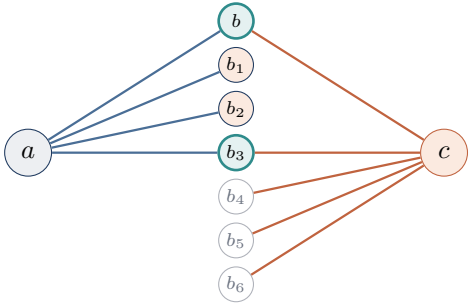
Simulation Lemma

If $Y \in [M]$ and U is uniform on $[M]$, then

$$\mathbf{Pr}[U \le Y] = \frac{\mathbf{E}[Y]}{M}.$$

Constructing Proxy r.v.

Sample w/o replacement the smaller neighbor list until the next overlap or exhaust. Let F be the number of non-overlap values before stop.



Key calculation: $F \sim \texttt{NegHypGeo}(1,\ |S| - 1,\ \deg(a, c) - 1)$, so $\mathbf{E}[F + 1] = \dfrac{|S|}{\deg(a, c)}$.

$$\mathbf{Pr}[\texttt{RandInt}(1, |S|) \le F + 1] = \frac{\mathbf{E}[F + 1]}{|S|} = \frac{1}{\deg(a, c)}. \quad \checkmark$$

Matrix Query: $\tilde{O}(N/\sqrt{\mathrm{OUT}})$ Analysis

Step-2 cost per attempt: $\mathbf{E}[F+1] = \dfrac{|S|}{\deg(a,c)} = \dfrac{\min\{|R_1 \ltimes a|,\ |R_2 \ltimes c|\}}{\deg(a,c)}$.

Per (Step 1 + Step 2) iteration:

$$1 + \sum_{(a,c)\in\mathcal{Q}_{\text{matrix}}(\mathcal{R})} \frac{\deg(a,c)}{\mathrm{OUT}_{\bowtie}} \cdot \frac{\min\{|R_1 \ltimes a|, |R_2 \ltimes c|\}}{\deg(a,c)} \ \le\ 1 + \frac{N\sqrt{\mathrm{OUT}}}{\mathrm{OUT}_{\bowtie}}.$$

(using $\min\{x,y\} \le \sqrt{xy}$ and Cauchy–Schwarz.)

Matrix Query: $\tilde{O}(N/\sqrt{\mathrm{OUT}})$ Analysis

Step-2 cost per attempt: $\mathbf{E}[F+1] = \dfrac{|S|}{\deg(a,c)} = \dfrac{\min\{|R_1 \ltimes a|,\, |R_2 \ltimes c|\}}{\deg(a,c)}$.

Per (Step 1 + Step 2) iteration:

$$1 + \sum_{(a,c)\in\mathcal{Q}_{\text{matrix}}(\mathcal{R})} \frac{\deg(a,c)}{\mathrm{OUT}_{\bowtie}} \cdot \frac{\min\{|R_1 \ltimes a|, |R_2 \ltimes c|\}}{\deg(a,c)} \;\le\; 1 + \frac{N\sqrt{\mathrm{OUT}}}{\mathrm{OUT}_{\bowtie}}.$$

(using $\min\{x,y\} \le \sqrt{xy}$ and Cauchy–Schwarz.) **Iterations needed** (acceptance prob. $= \mathrm{OUT}/\mathrm{OUT}_{\bowtie}$): $\mathrm{OUT}_{\bowtie}/\mathrm{OUT}$.

Total expected time:

$$\frac{\mathrm{OUT}_{\bowtie}}{\mathrm{OUT}} \cdot \left(1 + \frac{N\sqrt{\mathrm{OUT}}}{\mathrm{OUT}_{\bowtie}}\right) \;=\; O\!\left(\frac{\mathrm{OUT}_{\bowtie}}{\mathrm{OUT}} + \frac{N}{\sqrt{\mathrm{OUT}}}\right) \;=\; \boldsymbol{O\!\left(\frac{N}{\sqrt{\mathrm{OUT}}}\right)}.$$

(using $\mathrm{OUT}_{\bowtie} \le N\sqrt{\mathrm{OUT}}$ for $\mathcal{Q}_{\text{matrix}}$, Amossen–Pagh 2009.)

Theorem

For $\mathcal{Q}_{\text{matrix}}$: $O(N)$ preprocessing, $O(N/\sqrt{\mathbf{OUT}})$ expected sampling time.

Matrix: Matching Lower Bounds

Thm A: Counting LB — info-theoretic

Any algorithm for $O(1)$-approx counting on $\mathcal{Q}_{\text{matrix}}$ needs $\tilde{\Omega}(N/\sqrt{\text{OUT}})$ time.

Reduction from 2-party Set Disjointness.

Alice has S_A, Bob has $S_B \subseteq [m]$; decide $S_A \cap S_B \stackrel{?}{=} \emptyset$ (needs $\Omega(m)$ comm.).

Hard instance. $|\text{adom}(A)| = |\text{adom}(C)| = \sqrt{K}$;

- Alice: $R_1 = \text{adom}(A) \times S_A$
- Bob: $R_2 = S_B \times \text{adom}(C)$

$\text{OUT} = K$ iff $S_A \cap S_B \neq \emptyset$; else 0. $N = \sqrt{K}\, m/2$.

$O(1)$-approx count $\Rightarrow$ decides disjointness $\Rightarrow$ $\Omega(m) = \Omega(N/\sqrt{\text{OUT}})$.

$\Rightarrow$ **Info-theoretic:** rules out *any* algorithm, incl. fast matrix mult.

Matrix: Matching Lower Bounds

Thm A: Counting LB — info-theoretic

Any algorithm for $O(1)$-approx counting on $\mathcal{Q}_{\text{matrix}}$ needs $\tilde{\Omega}(N/\sqrt{\text{OUT}})$ time.

Reduction from 2-party Set Disjointness.

Alice has S_A, Bob has $S_B \subseteq [m]$; decide $S_A \cap S_B \stackrel{?}{=} \emptyset$ (needs $\Omega(m)$ comm.).

Hard instance. $|\text{adom}(A)| = |\text{adom}(C)| = \sqrt{K}$;

- Alice: $R_1 = \text{adom}(A) \times S_A$
- Bob: $R_2 = S_B \times \text{adom}(C)$

$\text{OUT} = K$ iff $S_A \cap S_B \neq \emptyset$; else 0. $N = \sqrt{K}\, m/2$.

$O(1)$-approx count $\Rightarrow$ decides disjointness $\Rightarrow$ $\Omega(m) = \Omega(N/\sqrt{\text{OUT}})$.

$\Rightarrow$ **Info-theoretic:** rules out *any* algorithm, incl. fast matrix mult.

Thm B: Sampling LB — conditional on comb. BMM

Assuming comb. hardness of Boolean mat. mult., no algorithm preprocesses in $O(N \cdot \text{OUT}^{1/2-\gamma})$ *and* samples in $O(N/\text{OUT}^{1/2+\gamma})$ for any $\gamma > 0$.

Reduction from listing. Combinatorial listing of $\mathcal{Q}_{\text{matrix}}(\mathcal{R})$ needs $\Omega(N\sqrt{\text{OUT}})$ time (Amossen–Pagh 2009).

Idea. A fast sampler + coupon-collector ($\tilde{O}(\text{OUT})$ samples) lists $\mathcal{Q}_{\text{matrix}}(\mathcal{R})$ in

$$\tilde{O}\Big(N\text{OUT}^{\frac{1}{2}-\gamma} + \text{OUT} \cdot \frac{N}{\text{OUT}^{\frac{1}{2}+\gamma}}\Big) = \tilde{O}(N\,\text{OUT}^{\frac{1}{2}-\gamma}),$$

beating $\Omega(N\sqrt{\text{OUT}})$. Contradiction.

$\Rightarrow$ **Conditional:** matches our UB *only* against combinatorial algorithms.

The info-theoretic counting LB also implies a sampling LB via reduction (Chen-Yi'20), but it only rules out simultaneously sublinear preprocessing and sublinear sampling — which does not cover our $O(N)$-preprocessing algorithm. The matching sampling LB therefore requires the separate combinatorial-hardness argument.

Star and Chain: Extensions and Dichotomy

Star $\mathcal{Q}_{\text{star}} = \pi_{A_1,\ldots,A_k}\left(\bowtie_i R_i(A_i, B)\right)$

Theorem (tight bound)

$\Theta(N/\mathrm{OUT}^{1/k})$ sampling *and* counting time.

Algorithm. Same two-step framework: weight b by $W_b = \prod_i |R_i \ltimes b|$, then accept via the simulation lemma on $\pi_B(R_{j^*} \ltimes a_{j^*})$ for $j^* = \arg\min_i |R_i \ltimes a_i|$.

Lower bounds.

- Counting: k-party set disjointness (info-theoretic)
- Sampling: combinatorial hardness of star listing (conditional)

Star and Chain: Extensions and Dichotomy

Star $\mathcal{Q}_{\text{star}} = \pi_{A_1,\ldots,A_k}\left(\bowtie_i R_i(A_i, B)\right)$

Theorem (tight bound)

$\Theta(N/\text{OUT}^{1/k})$ sampling *and* counting time.

Algorithm. Same two-step framework: weight b by $W_b = \prod_i |R_i \ltimes b|$, then accept via the simulation lemma on $\pi_B(R_{j^*} \ltimes a_{j^*})$ for $j^* = \arg\min_i |R_i \ltimes a_i|$.

Lower bounds.

- Counting: k-party set disjointness (info-theoretic)
- Sampling: combinatorial hardness of star listing (conditional)

Chain $\mathcal{Q}_{\text{chain}} = \pi_{A_1,A_{k+1}}\left(\bowtie_i R_i(A_i, A_{i+1})\right)$

Theorem (sharp phase transition)

$$k = 3: \quad \Theta\big(\min\{N,\, N^2/\text{OUT}\}\big)$$
$$k \geq 4: \quad \Theta(N)$$

Algorithm. For $k = 3$: combine prior $\tilde{O}(N^2/\text{OUT})$ with new $O(N)$ via the k-minimum-values (KMV) summary [Cohen 2008].

Both LBs are info-theoretic.

Approximate Counting: Heavy/Light Hybrid Reduction

Naive sampling-to-counting reduction [Chen-Yi'20] gives $\tilde{O}(N + N^{\rho^*}/\mathrm{OUT}) = \tilde{O}(N)$ for $\mathcal{Q}_{\mathsf{matrix}}$ — not sublinear in OUT.

Our refinement. Geometric guess $\Lambda \in [\mathrm{OUT}, 2\mathrm{OUT}]$ and partition:

$$B^{\mathsf{heavy}} = \{b : |R_2 \ltimes b| \geq \sqrt{\Lambda}\}, \quad B^{\mathsf{light}} = \mathsf{adom}(B) \setminus B^{\mathsf{heavy}}.$$

Three estimators, then combine

- **Heavy:** SampleMatrix-H — efficient because $|B^{\mathsf{heavy}}| = O(N/\sqrt{\Lambda})$.
- **Light:** SampleMatrix-L with rejection threshold $\Delta = \sqrt{\Lambda}$.
- **Intersection:** estimate $|R^{\mathsf{heavy}} \cap R^{\mathsf{light}}|$ via sampling.
- **Combine:** $\widehat{\mathrm{OUT}} = s^{\mathsf{heavy}} + s^{\mathsf{light}} - \min\{s^{\mathsf{heavy}}, \beta\, s^{\mathsf{light}}\}$.

Theorem

$\tilde{O}(N/\sqrt{\mathrm{OUT}})$ for ϵ-approx counting on $\mathcal{Q}_{\mathsf{matrix}}$ — matches the info-theoretic LB.

Analogous partition (threshold $\mathrm{OUT}^{1/k}$) gives $\tilde{O}(N/\mathrm{OUT}^{1/k})$ for $\mathcal{Q}_{\mathsf{star}}$; chain counting uses KMV.

Summary & Open Problems

Query	Sampling & Counting	Improvement	Lower-bound type
Matrix	$\Theta(N/\sqrt{\mathrm{OUT}})$	poly.	Counting: info-th.; Sampling: comb.
k-Star	$\Theta(N/\mathrm{OUT}^{1/k})$	poly.	Counting: info-th.; Sampling: comb.
3-Chain	$\Theta(\min\{N, N^2/\mathrm{OUT}\})$	tight	Both: info-theoretic
k-Chain ($k \geq 4$)	$\Theta(N)$	tight (hard)	Both: info-theoretic

Two reusable techniques:

- **Rejection + simulation lemma** for output-optimal sampling.
- **Heavy/light hybrid reduction** for output-optimal counting.

Open problems: combining $\sigma + \pi$; other join-project query classes; other operators; dynamic settings under updates.

Thank you — questions?