Acyclic Join Sampling under Selections

Dichotomy, Union Sampling, and Enumeration

Jinchao Huang Yufei Tao Sibo Wang
The Chinese University of Hong Kong

ICDT 2026 March 24–27, 2026 ▪ Tampere, Finland

1. Motivation

Join Sampling: Why?

Join evaluation on massive data can be prohibitively expensive — the result itself may be huge.

Many applications only need a sample

- Approximate query processing
- Query optimization (selectivity estimation)
- Interactive data exploration
- Training ML models without materializing the full join

Prior work: efficient sampling algorithms for joins **without** selection conditions.

Reality: queries **nearly always** include selection conditions!

Selections Are Everywhere

Example SQL query

```
SELECT * FROM Payment P, TaxPayer T
WHERE P.ssn = T.ssn AND
  ((T.job = 'prof')
    OR (T.job = 'lawyer')
    OR (T.gender = 'F' AND P.status = 'approved'))
```

Central question

Can we preprocess the join **once** into a near-linear space structure that supports **efficient uniform sampling** under any such runtime equality selection?

2. Background

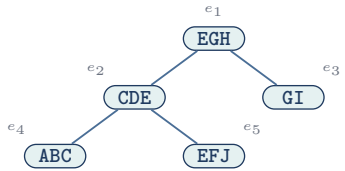
Background: Acyclic Joins

A schema graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: $\mathcal{V}$ = finite set of attributes, $\mathcal{E} \subseteq 2^{\mathcal{V}}$ = hyperedges.
A join instance $\mathcal{Q}$ of $\mathcal{G}$: for each $e \in \mathcal{E}$, a relation R_e with $schema(R_e) = e$.

Example:
Consider $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{\mathtt{A}, \mathtt{B}, ..., \mathtt{J}\}$ and 5 hyperedges: $e_1 = \mathtt{EGH}$ (shortform for $\{\mathtt{E}, \mathtt{G}, \mathtt{H}\}$), $e_2 = \mathtt{CDE}$, $e_3 = \mathtt{GI}$, $e_4 = \mathtt{ABC}$, $e_5 = \mathtt{EFJ}$.

$\mathcal{G}$ is acyclic if it admits a **join tree** $\mathcal{T}$:

- Nodes of $\mathcal{T} \leftrightarrow$ hyperedges in $\mathcal{E}$ (bijection)
- For each $X \in \mathcal{V}$, the nodes containing X form a connected subtree



Join tree $\mathcal{T}$ of $\mathcal{G}$

Prior Work: Join Sampling without Selections

Theorem (Zhao et al. 2018)

*Given a join instance $\mathcal{Q}$ with **acyclic** schema graph, one can build a structure of $O(\boldsymbol{IN})$ **space** supporting independent uniform sampling queries from $Join(\mathcal{Q})$ in $O(1)$ **time**.*

Key ideas:

1. Root the join tree $\mathcal{T}$ at an arbitrary node e^*
2. **Bottom-up:** compute **subjoin weights** at every node e:
   $w_e(\boldsymbol{u}) = |\boldsymbol{u} \bowtie Join(\mathcal{Q}_e)| = \prod_{i=1}^{t} |S_i(\boldsymbol{u})|$, where $S_i(\boldsymbol{u}) = \boldsymbol{u}[\mathbf{X}_i] \bowtie Join(\mathcal{Q}_{e_i})$
   (children-product lemma; $e_1, \ldots, e_t$ are children of e, $\mathbf{X}_i = e \cap e_i$)
3. **Top-down sampling:** at root, draw $\boldsymbol{u}^* \in R_{e^*}$ with prob. $\propto w_{e^*}(\boldsymbol{u}^*)$;
   at each child e_i, draw $\boldsymbol{v}$ from $R_{e_i}(\boldsymbol{u}) = \{\boldsymbol{v} \in R_{e_i} \mid \boldsymbol{v}[\mathbf{X}_i] = \boldsymbol{u}[\mathbf{X}_i]\}$ with prob. $\propto w_{e_i}(\boldsymbol{v})$, then recurse
4. **Alias structures** at each node $\Rightarrow O(1)$ time per level

Prior Work: Join Sampling without Selections

Theorem (Zhao et al. 2018)

*Given a join instance $\mathcal{Q}$ with **acyclic** schema graph, one can build a structure of $O(IN)$ **space** supporting independent uniform sampling queries from $Join(\mathcal{Q})$ in $O(1)$ **time**.*

Key ideas:

1. Root the join tree $\mathcal{T}$ at an arbitrary node e^*
2. **Bottom-up:** compute **subjoin weights** at every node e:
   $w_e(\boldsymbol{u}) = |\boldsymbol{u} \ltimes Join(\mathcal{Q}_e)| = \prod_{i=1}^{t} |S_i(\boldsymbol{u})|$, where $S_i(\boldsymbol{u}) = \boldsymbol{u}[\mathbf{X}_i] \ltimes Join(\mathcal{Q}_{e_i})$
   (children-product lemma; $e_1, \ldots, e_t$ are children of e, $\mathbf{X}_i = e \cap e_i$)
3. **Top-down sampling:** at root, draw $\boldsymbol{u}^* \in R_{e^*}$ with prob. $\propto w_{e^*}(\boldsymbol{u}^*)$;
   at each child e_i, draw $\boldsymbol{v}$ from $R_{e_i}(\boldsymbol{u}) = \{\boldsymbol{v} \in R_{e_i} \mid \boldsymbol{v}[\mathbf{X}_i] = \boldsymbol{u}[\mathbf{X}_i]\}$ with prob. $\propto w_{e_i}(\boldsymbol{v})$, then recurse
4. **Alias structures** at each node $\Rightarrow O(1)$ time per level

What if we add selection conditions?

3. Problem 1: Conjunctive Selections

Problem 1: $(\mathcal{G}, \mathcal{Z})$-Sampling

Fix a schema graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a subset $\mathcal{Z} \subseteq \mathcal{V}$ of selection attributes.

Given a tuple $\boldsymbol{z}$ over $\mathcal{Z}$ **at runtime**, define:

$$\sigma_{\boldsymbol{z}}(Join(\mathcal{Q})) = \left\{ \boldsymbol{u} \in Join(\mathcal{Q}) \;\middle|\; \forall Z \in \mathcal{Z} :\ \boldsymbol{u}(Z) = \boldsymbol{z}(Z) \right\}$$

$(\mathcal{G}, \mathcal{Z})$-Sampling Problem

Preprocess $\mathcal{Q}$ into a data structure that, given any $\boldsymbol{z}$, can independently return a **uniformly random** tuple from $\sigma_{\boldsymbol{z}}(Join(\mathcal{Q}))$ in each operation.

Feasible structure: $\tilde{O}(\mathrm{IN})$ space, $\tilde{O}(1)$ sample time.

Key challenge: The values in $\boldsymbol{z}$ are **unknown at preprocessing time**.

Dichotomy Theorem

Definition (Ext-$\mathcal{Z}$-Connexity (Bagan, Durand, Grandjean 2007))

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is **ext-$\mathcal{Z}$-connex** if both $(\mathcal{V}, \mathcal{E})$ and $(\mathcal{V}, \mathcal{E} \cup \{\mathcal{Z}\})$ are acyclic.

Theorem (Dichotomy)

*Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be acyclic and $\mathcal{Z} \subseteq \mathcal{V}$. Subject to the SSD conjecture, the $(\mathcal{G}, \mathcal{Z})$-sampling problem admits a feasible structure **if and only if** $\mathcal{G}$ is ext-$\mathcal{Z}$-connex.*

Equivalent characterization [BDG07]:

- $\mathcal{G}$ is ext-$\mathcal{Z}$-connex $\iff$ $\mathcal{G}$ has **no** $\mathcal{Z}$-path
- A $\mathcal{Z}$-path: $Z_1, X_1, \ldots, X_\ell, Z_2$ where $Z_1, Z_2 \in \mathcal{Z}$ are non-neighbors, and $X_i \notin \mathcal{Z}$

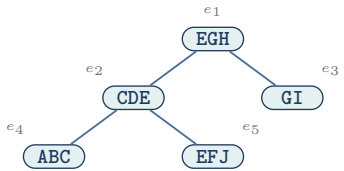
Dichotomy: Example

Example: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \{\mathtt{A}, \ldots, \mathtt{J}\}$ and 5 hyperedges.

$\mathcal{Z}$	$\emptyset$	E	AE	ACE	ACEH	ACEHI	ACEHF
ext-$\mathcal{Z}$-connex?	✓	✓	×	✓	✓	×	✓

$\mathcal{Z} = \{\mathtt{A}, \mathtt{E}\}$: $\mathcal{Z}$-path $\mathtt{A}, \mathtt{C}, \mathtt{E}$

$\mathcal{Z} = \{\mathtt{A}, \mathtt{C}, \mathtt{E}, \mathtt{H}, \mathtt{I}\}$: $\mathcal{Z}$-path $\mathtt{H}, \mathtt{G}, \mathtt{I}$



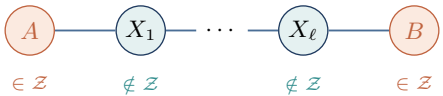
Join tree $\mathcal{T}$ of $\mathcal{G}$

Lower Bound: Reduction from Set Disjointness

SSD Conjecture: Given $m \geq 2$ sets $S_1, \ldots, S_m$ with $N = \sum_{i=1}^{m} |S_i|$, any data structure answering "$S_a \cap S_b \stackrel{?}{=} \emptyset$" in $\tilde{O}(1)$ time requires $\tilde{\Omega}(N^2)$ space.

Proof idea ("only if" direction — by contrapositive):

Not ext-$\mathcal{Z}$-connex $\Rightarrow \exists$ a $\mathcal{Z}$-path: $A, X_1, \ldots, X_\ell, B$

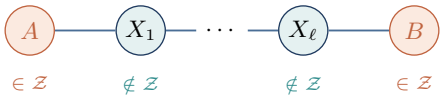


Lower Bound: Reduction from Set Disjointness

SSD Conjecture: Given $m \geq 2$ sets $S_1, \ldots, S_m$ with $N = \sum_{i=1}^{m} |S_i|$, any data structure answering "$S_a \cap S_b \stackrel{?}{=} \emptyset$" in $\tilde{O}(1)$ time requires $\tilde{\Omega}(N^2)$ space.

Proof idea ("only if" direction — by contrapositive):

Not ext-$\mathcal{Z}$-connex $\Rightarrow \exists$ a $\mathcal{Z}$-path: $A, X_1, \ldots, X_\ell, B$



Encode sets $S_1, \ldots, S_m$ into a join instance $\mathcal{Q}$ with $\mathrm{IN} = O(N)$:

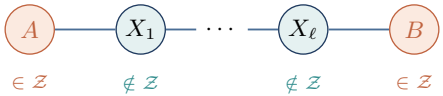
- $A, B \leftarrow$ set index, other $Z \in \mathcal{Z} \leftarrow \perp$, non-$\mathcal{Z}$ attributes $X_i \leftarrow$ element value
- Consecutive X_i, X_{i+1} share a hyperedge $\Rightarrow$ **forced equal** in any join result

Lower Bound: Reduction from Set Disjointness

SSD Conjecture: Given $m \geq 2$ sets $S_1, \ldots, S_m$ with $N = \sum_{i=1}^{m} |S_i|$, any data structure answering "$S_a \cap S_b \stackrel{?}{=} \emptyset$" in $\tilde{O}(1)$ time requires $\tilde{\Omega}(N^2)$ space.

Proof idea ("only if" direction — by contrapositive):

Not ext-$\mathcal{Z}$-connex $\Rightarrow \exists$ a $\mathcal{Z}$-path: $A, X_1, \ldots, X_\ell, B$



Encode sets $S_1, \ldots, S_m$ into a join instance $\mathcal{Q}$ with $\mathrm{IN} = O(N)$:

- $A, B \leftarrow$ set index, other $Z \in \mathcal{Z} \leftarrow \perp$, non-$\mathcal{Z}$ attributes $X_i \leftarrow$ element value
- Consecutive X_i, X_{i+1} share a hyperedge $\Rightarrow$ **forced equal** in any join result

Key Lemma

Query with $\boldsymbol{z}(A) = a,\ \boldsymbol{z}(B) = b,\ \boldsymbol{z}(Z) = \perp$ for other $Z \in \mathcal{Z}$: $\quad S_a \cap S_b \neq \emptyset \iff \sigma_{\boldsymbol{z}}(Join(\mathcal{Q})) \neq \emptyset$

$\Rightarrow$ A feasible sampling structure gives $\tilde{O}(N)$-space, $\tilde{O}(1)$-time set disjointness — breaks SSD!

Optimal Structure for $(\mathcal{G}, \mathcal{Z})$-Sampling

Theorem (Feasible Structure)

If $\mathcal{G}$ is ext-$\mathcal{Z}$-connex: ***O(IN)*** *space,* ***O(1)*** *sample time.*
Construction: $O(IN)$ expected time.

Case 1: $\exists e \in \mathcal{E}$ covering $\mathcal{Z}$

- **Root** join tree $\mathcal{T}$ at e
- **Group** R_e by $\boldsymbol{z}$-values (hashing)
- Build **alias structure** on each group $R_e(\boldsymbol{z})$
  $\Rightarrow$ weighted sample in $O(1)$
- Recurse down the tree (as in Zhao et al.)

Optimal Structure for $(\mathcal{G}, \mathcal{Z})$-Sampling

Theorem (Feasible Structure)

If $\mathcal{G}$ is ext-$\mathcal{Z}$-connex: ***O(IN)*** *space,* ***O(1)*** *sample time.*
Construction: $O(IN)$ expected time.

Case 1: $\exists e \in \mathcal{E}$ covering $\mathcal{Z}$

- **Root** join tree $\mathcal{T}$ at e
- **Group** R_e by $\boldsymbol{z}$-values (hashing)
- Build **alias structure** on each group $R_e(\boldsymbol{z})$ $\Rightarrow$ weighted sample in $O(1)$
- Recurse down the tree (as in Zhao et al.)

Case 2: no single e covers $\mathcal{Z}$

- Split $\mathcal{T}$ into subtrees $\mathcal{G}_1, \ldots, \mathcal{G}_s$ s.t. each $\mathcal{G}_i$'s $\mathcal{Z}$-attributes are covered by one hyperedge
- Each $\mathcal{G}_i$ falls into Case 1 $\Rightarrow$ sample $\boldsymbol{u}_i$ independently
- **Combine:** $\boldsymbol{u}_1 \bowtie \cdots \bowtie \boldsymbol{u}_s$
  (components are independent under $\boldsymbol{z}$)

4. Problem 2: Disjunctive Selections

Problem 2: $(\mathcal{G}, \vee_{i=1}^{m} \mathcal{Z}_i)$-Sampling

Disjunctive selections — the common case in practice:

Example

```
SELECT * FROM Payment P, TaxPayer T
WHERE P.ssn = T.ssn AND
  ((T.job = 'prof') OR (T.job = 'lawyer')
   OR (T.gender = 'F' AND P.status = 'approved'))
```

$\mathcal{Z}_1 = \{\texttt{job}\},\ \mathcal{Z}_2 = \{\texttt{job}\},\ \mathcal{Z}_3 = \{\texttt{gender}, \texttt{status}\}$ — sets $\sigma_{\boldsymbol{z}_i}(Join(\mathcal{Q}))$ **can overlap**!

Goal: Sample uniformly from $\sigma_{\boldsymbol{z}_1 \vee \cdots \vee \boldsymbol{z}_m}(Join(\mathcal{Q})) = \bigcup_{i=1}^{m} \sigma_{\boldsymbol{z}_i}(Join(\mathcal{Q}))$

Core challenge: $\Rightarrow$ the Set Union Sampling problem.

Set Union Sampling

Problem

Given $m \geq 2$ sets $S_1, \ldots, S_m$, each supporting $O(1)$-time:

1. **Size:** return $|S_i|$
2. **Membership:** check if a given element is in S_i
3. **Sampling:** return a uniformly random element of S_i

Goal: Sample uniformly from $\bigcup_{i=1}^{m} S_i$.

Let $n = |\bigcup_{i=1}^{m} S_i|$ and $N = \sum_{i=1}^{m} |S_i|$.

	Expected Sample Time	Remarks
Prior best	$O(\min\{m^2, m \log^2 N\})$	[CZB+22, AHM+22]
Our result	**O(m)**	(optimal: $\Omega(m)$ to read input)

Baseline Method

Repeat until success:

1. Draw $X \in [m]$ with $\mathbf{Pr}[X = x] = |S_x|/N$
2. Sample Y from S_X uniformly
3. Compute $\deg(Y)$ and **Accept** Y with probability $1/\deg(Y)$ where $\deg(y) = |\{x : y \in S_x\}|$

Baseline Method

Repeat until success:

1. Draw $X \in [m]$ with $\mathbf{Pr}[X = x] = |S_x|/N$
2. Sample Y from S_X uniformly
3. Compute $\deg(Y)$ and **Accept** Y with probability $1/\deg(Y)$ where $\deg(y) = |\{x : y \in S_x\}|$

Correctness: For any $y \in \bigcup S_i$:

$$\mathbf{Pr}[\text{output } y] = \sum_{x \in \text{InvS}(y)} \underbrace{\mathbf{Pr}[X = x]}_{|S_x|/N} \cdot \underbrace{\mathbf{Pr}[Y = y \mid X = x]}_{1/|S_x|} \cdot \frac{1}{\deg(y)} = \frac{1}{N}$$

$\Rightarrow$ uniform!

Success probability: n/N per repeat $\Rightarrow$ N/n repeats expected.

Baseline Method

Repeat until success:

1. Draw $X \in [m]$ with $\mathbf{Pr}[X = x] = |S_x|/N$
2. Sample Y from S_X uniformly
3. Compute $\deg(Y)$ and **Accept** Y with probability $1/\deg(Y)$ where $\deg(y) = |\{x : y \in S_x\}|$

Correctness: For any $y \in \bigcup S_i$:

$$\mathbf{Pr}[\text{output } y] = \sum_{x \in \text{InvS}(y)} \underbrace{\mathbf{Pr}[X = x]}_{|S_x|/N} \cdot \underbrace{\mathbf{Pr}[Y = y \mid X = x]}_{1/|S_x|} \cdot \frac{1}{\deg(y)} = \frac{1}{N}$$

$\Rightarrow$ uniform!

Success probability: n/N per repeat $\Rightarrow$ N/n repeats expected.

Bottleneck

Computing $\deg(Y)$ requires querying membership in all m sets: $O(m)$ per repeat.
Total: $O(m \cdot N/n)$ — can be as bad as $O(m^2)$ when all the sets are the same.

Key Idea: Simulating Acceptance Step using Random Events

Goal: Simulate "Accept Y with probability $1/\deg(Y)$" in $O(m/\deg(Y))$ expected time.
(instead of $O(m)$ computing).

Proposition (Simulation)

For any random variable $\Gamma \in [m]$, and U uniform on $[m]$ independent of Γ:

$$\mathbf{Pr}[U \leq \Gamma] = \frac{1}{m}\,\mathbf{E}[\Gamma]$$

Key Idea: Simulating Acceptance Step using Random Events

Goal: Simulate "Accept Y with probability $1/\deg(Y)$" in $O(m/\deg(Y))$ expected time.
(instead of $O(m)$ computing).

Proposition (Simulation)

For any random variable $\Gamma \in [m]$, and U uniform on $[m]$ independent of Γ:

$$\mathbf{Pr}[U \leq \Gamma] = \frac{1}{m}\,\mathbf{E}[\Gamma]$$

The `find-2nd` procedure: Given $x \in [m]$ and $y \in S_x$:

- Sample indices i from $[m] \setminus \{x\}$ **without replacement**
- Stop when finding another S_i containing y
- Let $F =$ number of failures before success (or $m-1$ if $\deg(y) = 1$)

Set $\Gamma = F + 1$. Then $\mathbf{E}[\Gamma] = m/\deg(y)$.
Result: Accepting when $U \leq \Gamma$ succeeds with prob. $1/\deg(y)$ in expected cost $O(m/\deg(y))$.

Why the Total Cost Is $O(m)$

Cost of one repeat (acceptance step):

$$\sum_{x\in[m]}\sum_{y\in S_x}\underbrace{\mathbf{Pr}[X=x,Y=y]}_{=1/N}\cdot O\left(\frac{m}{\deg(y)}\right)=\frac{m}{N}\sum_{y\in\cup S_i}\underbrace{\sum_{x\in\mathrm{InvS}(y)}\frac{1}{\deg(y)}}_{=1}=O\left(\frac{mn}{N}\right)$$

Why the Total Cost Is $O(m)$

Cost of one repeat (acceptance step):

$$\sum_{x\in[m]}\sum_{y\in S_x}\underbrace{\mathbf{Pr}[X=x,Y=y]}_{=1/N}\cdot O\Big(\frac{m}{\deg(y)}\Big)=\frac{m}{N}\sum_{y\in\cup S_i}\underbrace{\sum_{x\in\mathrm{InvS}(y)}\frac{1}{\deg(y)}}_{=1}=O\Big(\frac{mn}{N}\Big)$$

Total expected sample time:

- Each repeat costs $O(mn/N)$ expected
- Number of repeats: N/n expected

$$\mathbf{E}[T_{\text{total}}]=O\Big(\frac{mn}{N}\Big)+\Big(1-\frac{n}{N}\Big)\cdot\mathbf{E}[T_{\text{total}}]\quad\Longrightarrow\quad\mathbf{E}[T_{\text{total}}]=\mathbf{O(m)}$$

Disjunctive Sampling & Necessity

Reduction to set union sampling:

- Build a $(\mathcal{G}, \mathcal{Z}_i)$-sampling structure for each **distinct** $\mathcal{Z}_i$
- Only $O(1)$ distinct subsets $\Rightarrow$ total $O(\mathrm{IN})$ space
- Each $S_i = \sigma_{\boldsymbol{z}_i}(Join(\mathcal{Q}))$ supports size/membership/sampling in $O(1)$

Theorem (Disjunctive Sampling)

If $\mathcal{G}$ is ext-$\mathcal{Z}_i$-connex for all i: $O(\boldsymbol{IN})$ space, $O(m)$ expected sample time.

Disjunctive Sampling & Necessity

Reduction to set union sampling:

- Build a $(\mathcal{G}, \mathcal{Z}_i)$-sampling structure for each **distinct** $\mathcal{Z}_i$
- Only $O(1)$ distinct subsets $\Rightarrow$ total $O(\mathrm{IN})$ space
- Each $S_i = \sigma_{\boldsymbol{z}_i}(Join(\mathcal{Q}))$ supports size/membership/sampling in $O(1)$

Theorem (Disjunctive Sampling)

If $\mathcal{G}$ is ext-$\mathcal{Z}_i$-connex for all i: $O(\boldsymbol{IN})$ space, $O(m)$ expected sample time.

Theorem (Necessity)

If $\mathcal{G}$ is not ext-$\mathcal{Z}_i$-connex for some i and $m \leq IN^{0.49}$: no structure of $\tilde{O}(IN)$ space can guarantee $\tilde{O}(m)$ expected sample time, subject to the SSD conjecture.

5. Random Enumeration

Random Enumeration

Random enumeration of $\sigma_{\boldsymbol{z}}(Join(\mathcal{Q}))$: output a uniformly random permutation.

Theorem (Random Enumeration)

*If $\mathcal{G}$ is ext-$\mathcal{Z}$-connex: $O(IN)$ space, $O(1)$ **expected delay**.*
(Prior best [CZB+22]: $O(\log IN)$ delay for $\mathcal{Z} = \emptyset$)

High-level idea: Let $S = \sigma_{\boldsymbol{z}}(Join(\mathcal{Q}))$.

1. **Sample phase:** Draw random samples from S using our $O(1)$-time sampler until half of S is discovered. Output these elements one by one.
2. **Enumerate phase:** Enumerate all of S, extract the undiscovered half, randomly permute, and output.

Key technique: de-amortization

The enumerate phase has a bulk startup cost. We **buffer** sampled elements from Phase 1 and interleave their output to mask this cost $\Rightarrow$ $O(1)$ **delay** throughout.

6. Summary

Summary of Results

Problem	Space	Time	Condition
$(\mathcal{G}, \mathcal{Z})$-sampling	$O(\mathrm{IN})$	$O(1)$	ext-$\mathcal{Z}$-connex
$\hookrightarrow$ *Lower bound: no $\tilde{O}(IN)$-space, $\tilde{O}(1)$-time str. if* ***not*** *ext-$\mathcal{Z}$-connex*			
Set union sampling	N/A	$O(m)$	—
$(\mathcal{G}, \bigvee \mathcal{Z}_i)$-sampling	$O(\mathrm{IN})$	$O(m)$	all ext-$\mathcal{Z}_i$-connex
$\hookrightarrow$ *Lower bound: no $\tilde{O}(IN)$-space, $\tilde{O}(m)$-time str. if* ***any not*** *ext-$\mathcal{Z}_i$-connex*			
Random enumeration	$O(\mathrm{IN})$	$O(1)$ delay	ext-$\mathcal{Z}$-connex

Open questions:

- Other forms of selection conditions (inequalities, LIKE, ...)?
- Set union sampling k elements in $O(m + k)$ time? (nontrivial to improve over $O(mk)$!)
- Sampling over joins with other relational operators? (projection is recently studied in PODS'26)
- Other sampling semantics over joins? (Poisson/Subset Sampling in SIGMOD/PODS'26)

Thank You

Questions?

Acyclic Join Sampling under Selections:
Dichotomy, Union Sampling, and Enumeration

Jinchao Huang, Yufei Tao, Sibo Wang ▪ ICDT 2026