# Directed st-connectivity with few paths is in Quantum Logspace

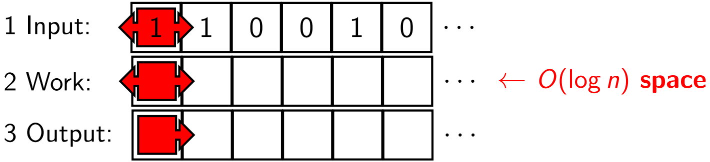
Simon Apers, Roman Edenhofer

Algorithms and Complexity, IRIF, Université Paris Cité
apers@irif.fr

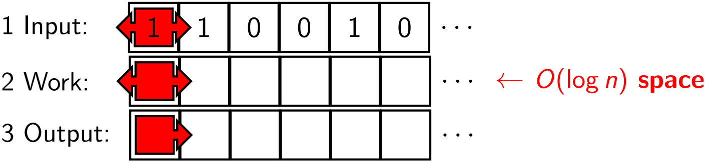
December 2024
arxiv:2408.12473

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

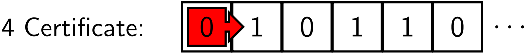
- **L** = deterministic logspace



- **L** = deterministic logspace




- **NL** = non-deterministic logspace



# Complexity Classes

- **L** = deterministic logspace

  1 Input: 1 1 0 0 1 0 ⋯

  2 Work: ⋯ ← $O(\log n)$ **space**

  3 Output: ⋯

- **NL** = non-deterministic logspace

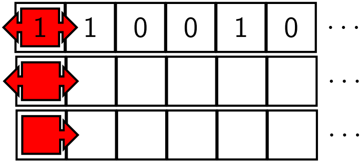
  4 Certificate: 0 1 0 1 1 0 ⋯

- **BPL** = bounded-error randomized logspace

  4' Coins: 1 0 0 1 0 1 ⋯

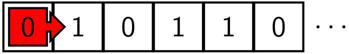
- **L** = deterministic logspace

1 Input:
2 Work: ← $O(\log n)$ **space**
3 Output:



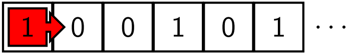

- **NL** = non-deterministic logspace

4 Certificate:



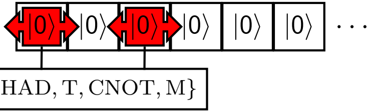

- **BPL** = bounded-error randomized logspace

4' Coins:

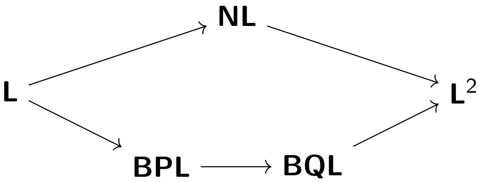


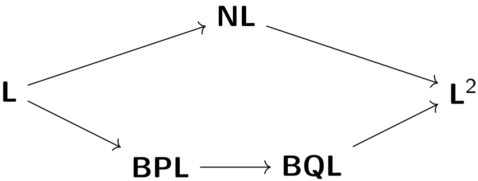

- **BQL** = bounded-error quantum logspace

4'' Quantum: ← $O(\log n)$ **space**



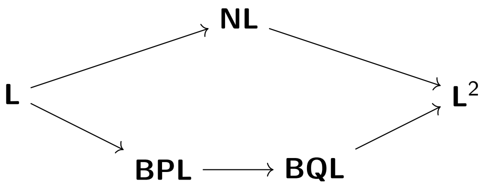
# Complexity Classes





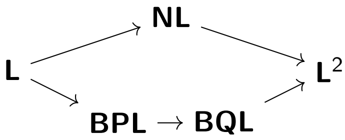

- **NL** $\rightarrow$ $\mathbf{L}^2$ [Savitch '70]

## Complexity Classes

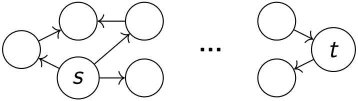


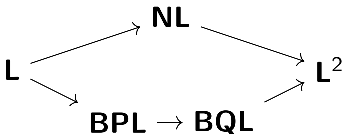

- **NL** $\rightarrow$ $\mathbf{L}^2$ [Savitch '70]
- **BQL** $\rightarrow$ $\mathbf{L}^2$ [Watrous '99]

# Complexity Classes

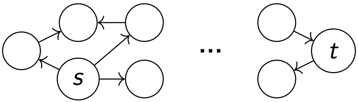



- STCON: *st*-connectivity on a *directed* graph

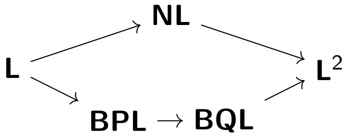




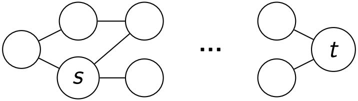

- STCON: *st*-connectivity on a *directed* graph

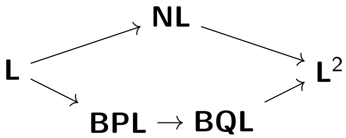



**NL**-complete

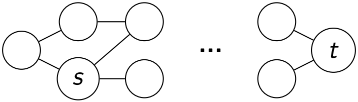



- USTCON: *st*-connectivity on an *undirected* graph

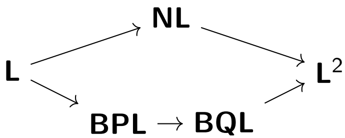




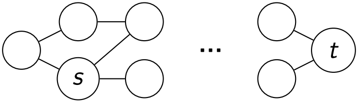

- USTCON: *st*-connectivity on an *undirected* graph




**L**-hard [Cook-McKenzie '87]

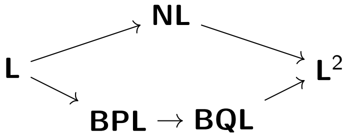



- USTCON: *st*-connectivity on an *undirected* graph

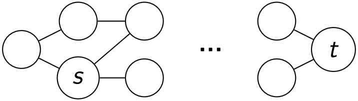



**L**-hard [Cook-McKenzie '87]
$\in$ **BPL** [AKL+ '79] (random walks mix fast!)

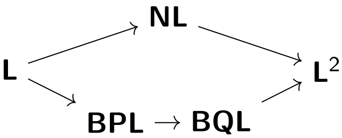
# Complexity Classes



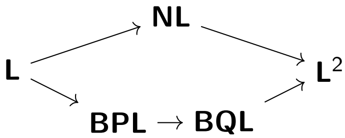

- USTCON: *st*-connectivity on an *undirected* graph




**L**-hard [Cook-McKenzie '87]
$\in$ **BPL** [AKL+ '79] (random walks mix fast!)
$\in$ **L** [Reingold '05] (derandomization)



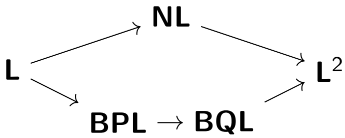

- pc-MATINV:




- pc-MATINV:

given $A \in \mathbb{C}^{n\times n}$ with $\|A\|, \|A^{-1}\| \leq \mathrm{poly}(n)$,
return $A^{-1}(i,j) \pm \frac{1}{\mathrm{poly}(n)}$

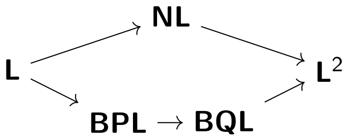
# Complexity Classes




- pc-MATINV:

given $A \in \mathbb{C}^{n\times n}$ with $\|A\|, \|A^{-1}\| \leq \mathrm{poly}(n)$,
return $A^{-1}(i,j) \pm \frac{1}{\mathrm{poly}(n)}$

$\in$ **BQL** [Ta-Shma '13] (HHL in logspace)



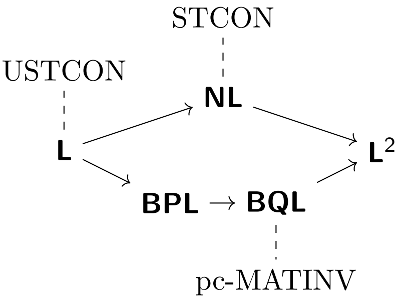

- pc-MATINV:

given $A \in \mathbb{C}^{n\times n}$ with $\|A\|, \|A^{-1}\| \leq \mathrm{poly}(n)$,
return $A^{-1}(i,j) \pm \frac{1}{\mathrm{poly}(n)}$

$\in$ **BQL** [Ta-Shma '13] (HHL in logspace)

**BQL**-complete [Fefferman-Lin '16]

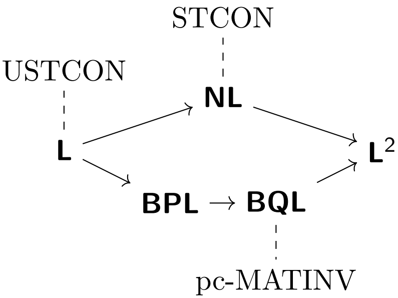
# Complexity Classes




## Motivating question

*quantum advantage for STCON?*

# Complexity Classes



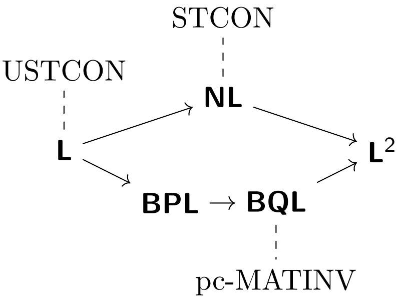

## Motivating question

*quantum advantage for STCON?*

## Main results

- Restricted version of STCON in **BQL**

# Complexity Classes



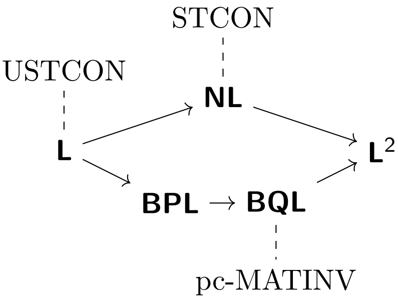

## Motivating question

*quantum advantage for STCON?*

## Main results

- Restricted version of STCON in **BQL** (and $1000 for putting it in **BPL**)

# Complexity Classes




## Motivating question

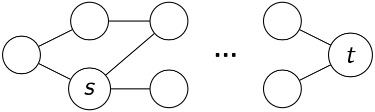
*quantum advantage for STCON?*

## Main results

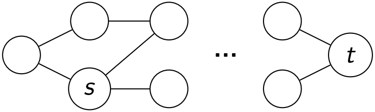
- Restricted version of STCON in **BQL** (and $1000 for putting it in **BPL**)
- First language in **BQL** not known to be in **L**

# Random walks

## Undirected graph



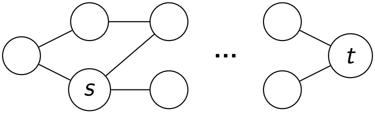
## Undirected graph




hitting time $O(n^3)$

# Random walks

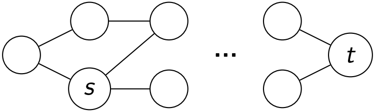
## Undirected graph




hitting time $O(n^3)$

random walk matrix $P$ **well-conditioned**

## Undirected graph



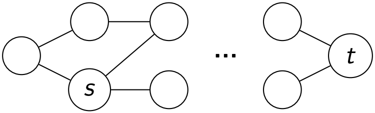

hitting time $O(n^3)$

random walk matrix $P$ **well-conditioned**

($\longrightarrow$ can use pc-MATINV in **BQL**)

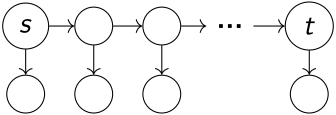
# Random walks

## Undirected graph



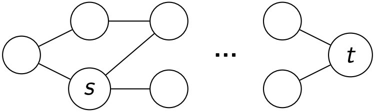

hitting time $O(n^3)$
random walk matrix $P$ **well-conditioned**
($\longrightarrow$ can use pc-MATINV in **BQL**)

## Directed graph



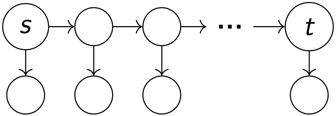
# Random walks

## Undirected graph




hitting time $O(n^3)$
random walk matrix $P$ **well-conditioned**
($\longrightarrow$ can use pc-MATINV in **BQL**)

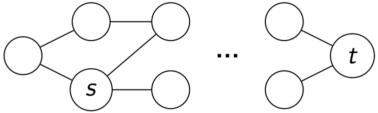
## Directed graph




hitting time can be exponential!

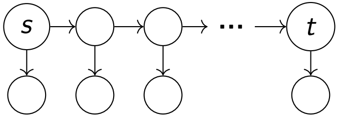
# Random walks

## Undirected graph




hitting time $O(n^3)$
random walk matrix $P$ **well-conditioned**
($\longrightarrow$ can use pc-MATINV in **BQL**)

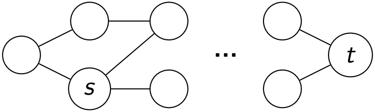
## Directed graph




hitting time can be exponential!
random walk matrix $P$ can be **ill-conditioned**

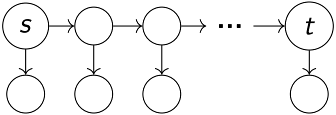
# Random walks

## Undirected graph




hitting time $O(n^3)$
random walk matrix $P$ **well-conditioned**
($\longrightarrow$ can use pc-MATINV in **BQL**)

## Directed graph




hitting time can be exponential!
random walk matrix $P$ can be **ill-conditioned**
($\longrightarrow$ cannot use pc-MATINV in **BQL**)

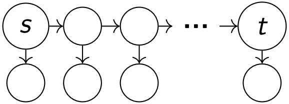
# A new hope

Let $N(i, j) := \#\text{paths}(i, j)$.

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

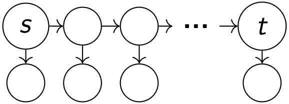
If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.



Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.

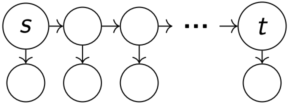



**Proof.** Assume $G$ DAG (wlog).

Let $N(i,j) := \#\text{paths}(i,j)$.

**Few paths in BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.

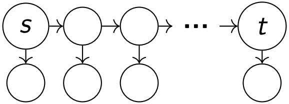



**Proof.** Assume $G$ DAG (wlog).
Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




**Proof.** Assume $G$ DAG (wlog).

Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

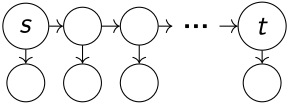
$\Rightarrow\ \mathcal{L}^{-1} = I + A + \ldots + A^{n-1}$ because $A^n = 0$

# A new hope

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




**Proof.** Assume $G$ DAG (wlog).

Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

$\Rightarrow\ \mathcal{L}^{-1} = I + A + \ldots + A^{n-1}$ because $A^n = 0$

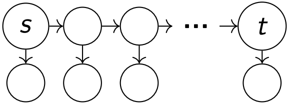
$\Rightarrow\ \mathcal{L}^{-1}(i,j) = N(i,j)$

# A new hope

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




**Proof.** Assume $G$ DAG (wlog).

Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

$\Rightarrow\ \mathcal{L}^{-1} = I + A + \ldots + A^{n-1}$ because $A^n = 0$

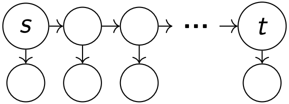
$\Rightarrow\ \mathcal{L}^{-1}(i,j) = N(i,j)$

Further, $\begin{cases} \|\mathcal{L}\|_2 \leq n\|\mathcal{L}\|_{\max} = n \\ \end{cases}$

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \text{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




**Proof.** Assume $G$ DAG (wlog).

Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

$\Rightarrow\ \mathcal{L}^{-1} = I + A + \ldots + A^{n-1}$ because $A^n = 0$

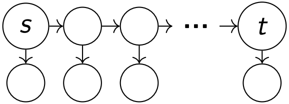
$\Rightarrow\ \mathcal{L}^{-1}(i,j) = N(i,j)$

Further, $\begin{cases} \|\mathcal{L}\|_2 \leq n\|\mathcal{L}\|_{\max} = n \\ \|\mathcal{L}^{-1}\|_2 \leq n\|\mathcal{L}^{-1}\|_{\max} = n \max_{i,j} N(i,j) \end{cases}$

Let $N(i,j) := \#\text{paths}(i,j)$.

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




**Proof.** Assume $G$ DAG (wlog).
Consider $\mathcal{L} := I - A$ for adjacency matrix $A$ of a DAG (wlog).

$\Rightarrow\ \mathcal{L}^{-1} = I + A + \ldots + A^{n-1}$ because $A^n = 0$

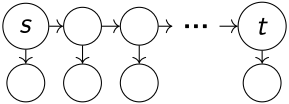
$\Rightarrow\ \mathcal{L}^{-1}(i,j) = N(i,j)$

Further, $$\begin{cases} \|\mathcal{L}\|_2 \leq n\|\mathcal{L}\|_{\max} = n \\ \|\mathcal{L}^{-1}\|_2 \leq n\|\mathcal{L}^{-1}\|_{\max} = n \max_{i,j} N(i,j) \end{cases}$$

$\Rightarrow \mathcal{L}$ is poly-conditioned iff $\forall i,j : N(i,j) \leq \mathrm{poly}(n)$.

### Few paths in **BQL**

If $N(i, j) \leq \mathrm{poly}(n)\ \forall i, j$, we can return $N(s, t)$ in **BQL**.



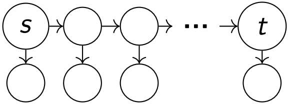

### Corollary

$\mathrm{STCON}_{\mathrm{sf}} := \{\langle G, s, t, 1^k \rangle \mid \forall i, j : N(i, j) \leq k \text{ and } N(s, t) \geq 1\} \in$ **BQL**.

# A language

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.



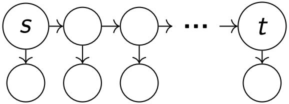

## Corollary

$\mathrm{STCON}_{\mathrm{sf}} := \{\langle G, s, t, 1^k \rangle \mid \forall i,j : N(i,j) \leq k \text{ and } N(s,t) \geq 1\} \in$ **BQL**.

! first (non-promise) **language** in **BQL**, not known to be in **BPL**

## Few paths in **BQL**

If $N(i,j) \leq \mathrm{poly}(n)\ \forall i,j$, we can return $N(s,t)$ in **BQL**.




## Corollary

$\mathrm{STCON}_{\mathrm{sf}} := \{\langle G, s, t, 1^k\rangle \mid \forall i,j : N(i,j) \leq k \text{ and } N(s,t) \geq 1\} \in$ **BQL**.

! first (non-promise) **language** in **BQL**, not known to be in **BPL**

For comparison:

- pc-MATINV [Ta-Shma '13]
- pc-MATPOW [Fefferman-Remscrim '20]
- quantum state testing [Le Gall-Liu-Wang '23]

all **promise problems**

# A weaker promise

## Few outgoing/incoming paths in **BQL**

If $N(s, \cdot)$, $N(\cdot, t) \leq \mathrm{poly}(n)$, we can return $N(s, t)$ in **BQL**.

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j \left|u_j\right\rangle \left\langle v_j\right|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

**Few outgoing/incoming paths in BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j \left|u_j\right\rangle \left\langle v_j\right|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{???} + \underbrace{\sum_{\sigma_j \geq \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{\text{easy}}$$

## Few outgoing/incoming paths in **BQL**

If $N(s, \cdot)$, $N(\cdot, t) \leq \mathrm{poly}(n)$, we can return $N(s, t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j |u_j\rangle \langle v_j|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|}_{???} + \underbrace{\sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|$

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j |u_j\rangle\langle v_j|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j<\kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|}_{???} + \underbrace{\sum_{\sigma_j\geq\kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j\geq\kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j |u_j\rangle \langle v_j|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|}_{???} + \underbrace{\sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle \langle u_j|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

**!** $N(s,\cdot) = \| \langle s| \mathcal{L}^{-1} \|_1 \leq \mathrm{poly}(n)$

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j \left|u_j\right\rangle \left\langle v_j\right|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{???} + \underbrace{\sum_{\sigma_j \geq \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \geq \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

**!** $N(s,\cdot) = \| \left\langle s\right| \mathcal{L}^{-1} \|_1 \leq \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} \left|\langle s|v_j\rangle\right| \leq \mathrm{poly}(n)$

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot), N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j |u_j\rangle\langle v_j|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|}_{???} + \underbrace{\sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \geq \kappa} \sigma_j^{-1} |v_j\rangle\langle u_j|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

**!** $N(s,\cdot) = \|\langle s|\mathcal{L}^{-1}\|_1 \leq \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle s|v_j\rangle| \leq \mathrm{poly}(n)$

**!** $N(\cdot,t) = \|\mathcal{L}^{-1}|t\rangle\|_1 \leq \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle u_j|t\rangle| \leq \mathrm{poly}(n)$

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot), N(\cdot,t) \le \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j \left|u_j\right\rangle \left\langle v_j\right|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{???} + \underbrace{\sum_{\sigma_j \ge \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \ge \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

$$\textbf{!}\quad N(s,\cdot) = \| \left\langle s\right| \mathcal{L}^{-1} \|_1 \le \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle s|v_j\rangle| \le \mathrm{poly}(n)$$

$$\textbf{!}\quad N(\cdot,t) = \| \mathcal{L}^{-1} \left|t\right\rangle \|_1 \le \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle u_j|t\rangle| \le \mathrm{poly}(n)$$

and so

$$\sigma_j^{-1} | \langle s|v_j\rangle \langle u_j|t\rangle | \le \sigma_j \cdot \mathrm{poly}(n)$$

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot), N(\cdot,t) \le \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

**Proof.** Consider $\mathcal{L} := I - A = \sum_j \sigma_j \left|u_j\right\rangle \left\langle v_j\right|$. Pick $\kappa = 1/\mathrm{poly}(n)$.

$$\Rightarrow \mathcal{L}^{-1} = \underbrace{\sum_{\sigma_j < \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{???} + \underbrace{\sum_{\sigma_j \ge \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|}_{\text{easy}}$$

$\Rightarrow$ return "effective" inverse $\widetilde{\mathcal{L}}^{-1} = \sum_{\sigma_j \ge \kappa} \sigma_j^{-1} \left|v_j\right\rangle \left\langle u_j\right|$

**?** $\langle s|\widetilde{\mathcal{L}}^{-1}|t\rangle \approx \langle s|\mathcal{L}^{-1}|t\rangle = N(s,t)$ **:**

$$! \quad N(s,\cdot) = \| \langle s| \mathcal{L}^{-1} \|_1 \le \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle s|v_j\rangle| \le \mathrm{poly}(n)$$

$$! \quad N(\cdot,t) = \| \mathcal{L}^{-1} |t\rangle \|_1 \le \mathrm{poly}(n) \Rightarrow \sigma_j^{-1} |\langle u_j|t\rangle| \le \mathrm{poly}(n)$$

and so

$$\sigma_j^{-1} | \langle s|v_j\rangle \langle u_j|t\rangle | \le \sigma_j \cdot \mathrm{poly}(n) \le 1/\mathrm{poly}(n) \quad \text{(for right choice of } \kappa\text{).}$$

**Few outgoing/incoming paths in BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

# Comparison

## Few outgoing/incoming paths in **BQL**

If $N(s,\cdot)$, $N(\cdot,t) \leq \mathrm{poly}(n)$, we can return $N(s,t)$ in **BQL**.

## Few outgoing paths in $\mathbf{DSPACE}(o(\log^2 n))$

If $N(s,\cdot) \leq \mathrm{poly}(n)$, can decide STCON in $\mathbf{DSPACE}\left(\frac{\log^2(n)}{\log\log(n)}\right)$.

[Allender-Lange '98], [Garvin-Stolee-Tewari-Vinodchandran '11]

## Few outgoing/incoming paths in **BQL**

If $N(s, \cdot)$, $N(\cdot, t) \leq \mathrm{poly}(n)$, we can return $N(s, t)$ in **BQL**.

## Few outgoing paths in $\mathbf{DSPACE}(o(\log^2 n))$

If $N(s, \cdot) \leq \mathrm{poly}(n)$, can decide STCON in $\mathbf{DSPACE}\left(\frac{\log^2(n)}{\log\log(n)}\right)$.

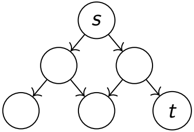
[Allender-Lange '98], [Garvin-Stolee-Tewari-Vinodchandran '11]

## Few outgoing paths in $\mathbf{SC}^2$

If $N(s, \cdot) \leq \mathrm{poly}(n)$, can decide STCON in $\mathrm{poly}(n)$ time and $O(\log^2 n)$ space.

[Lange '97], [Garvin-Stolee-Tewari-Vinodchandran '11]

**!** can connect to existing complexity classes: $\langle G, s, t \rangle$ is called ...



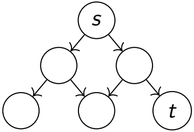

unambiguous

$:\Leftrightarrow N(s,t) \leq 1$

few unamb.

$\cdots \leq \mathrm{poly}(n)$

**!** can connect to existing complexity classes: $\langle G, s, t\rangle$ is called ...

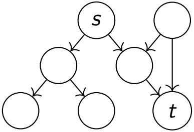



unambiguous

$:\Leftrightarrow N(s,t) \leq 1$

few unamb.

$\cdots \leq \mathrm{poly}(n)$



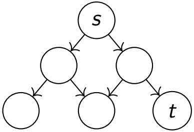

reach-unambiguous

$:\Leftrightarrow \forall j : N(s,j) \leq 1$

reach-few

$\cdots \leq \mathrm{poly}(n)$

**!** can connect to existing complexity classes: $\langle G, s, t\rangle$ is called ...

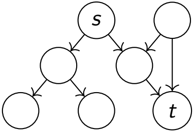



unambiguous

$:\Leftrightarrow N(s,t) \leq 1$

few unamb.

$\cdots \leq \mathrm{poly}(n)$

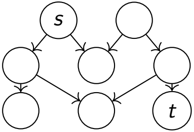



reach-unambiguous

$:\Leftrightarrow \forall j : N(s,j) \leq 1$

reach-few

$\cdots \leq \mathrm{poly}(n)$



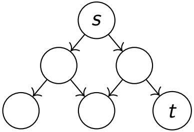

strongly unambiguous

$:\Leftrightarrow \forall i,j : N(i,j) \leq 1$

strongly-few

$\cdots \leq \mathrm{poly}(n)$

**!** can connect to existing complexity classes: $\langle G, s, t \rangle$ is called ...

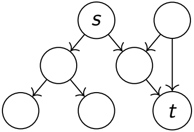



unambiguous

$:\Leftrightarrow N(s,t) \leq 1$

few unamb.

$\dots \leq \mathrm{poly}(n)$

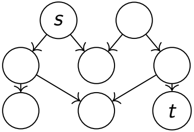



reach-unambiguous

$:\Leftrightarrow \forall j : N(s,j) \leq 1$

reach-few

$\dots \leq \mathrm{poly}(n)$



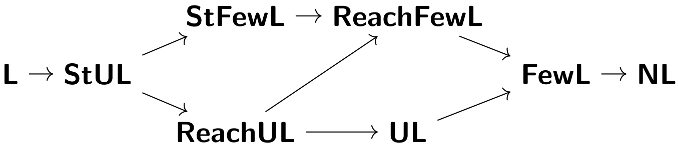

strongly unambiguous

$:\Leftrightarrow \forall i,j : N(i,j) \leq 1$

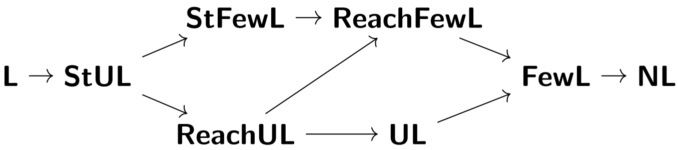
strongly-few

$\dots \leq \mathrm{poly}(n)$

$\Rightarrow$ corresponding complexity classes:



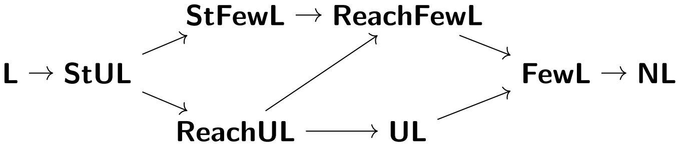
# Unambiguity & Fewness




**Corollary:**

**StFewL** $\subseteq$ **BQL**

**Further Results:**

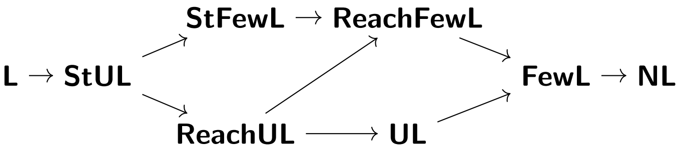



**Corollary:**

**StFewL** $\subseteq$ **BQL**

**Further Results:**

1. [AL98]: **ReachUL** $\subseteq \mathrm{DSPACE}(\frac{\log^2 n}{\log \log n})$

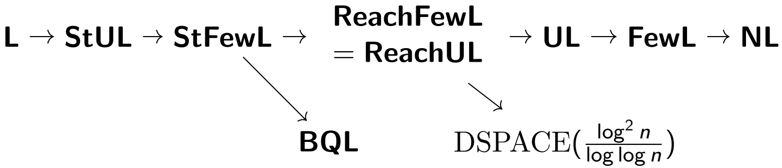



## Corollary:

**StFewL** $\subseteq$ **BQL**

## Further Results:

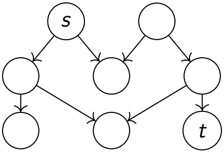
1. [AL98]: **ReachUL** $\subseteq \mathrm{DSPACE}(\frac{\log^2 n}{\log\log n})$
2. [GSTV11]: **ReachUL** $=$ **ReachFewL**



# Open Questions

## \$1000 reward for a dequantization [All23]

STCON on StU graphs in DSPACE$(o(\log^2 n/\log\log n))$?

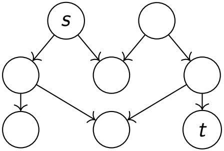



$\forall i, j : N(i, j) \leq 1$

# Open Questions

## \$1000 reward for a dequantization [All23]

STCON on StU graphs in $\mathrm{DSPACE}(o(\log^2 n / \log\log n))$?



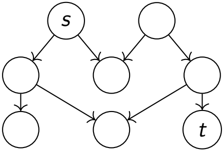

$\forall i, j : N(i, j) \leq 1$

- **BQL**-hardness of STCON-variant?

# Open Questions

## \$1000 reward for a dequantization [All23]

STCON on StU graphs in $\mathrm{DSPACE}(o(\log^2 n / \log\log n))$?




$\forall i, j : N(i, j) \leq 1$

- **BQL**-hardness of STCON-variant?
- Further investigation of quantum algorithms on directed graphs!

# References I


[AKL+79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, Laszlo Lovasz, and Charles Rackoff.
Random walks, universal traversal sequences, and the complexity of maze problems.
In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, 1979.

[AL98] Eric Allender and Klaus-Joern Lange.
$\mathsf{RUSPACE}(\log n) \subseteq \mathsf{DSPACE}(\log^2 n / \log\log n)$.
*Theory of Computing Systems*, 31, 1998.

[All23] Eric Allender.
Guest column: Parting thoughts and parting shots (read on for details on how to win valuable prizes!).
*SIGACT News*, 54(1), 2023.

[CM87] Stephen A Cook and Pierre McKenzie.
Problems complete for deterministic logarithmic space.
*Journal of Algorithms*, 8(3):385–394, 1987.

[FL16] Bill Fefferman and Cedric Yen-Yu Lin.
A complete characterization of unitary quantum space, 2016.

[FR21] Bill Fefferman and Zachary Remscrim.
Eliminating intermediate measurements in space-bounded quantum computation.
In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '21, 2021.

[GLW23] François Le Gall, Yupan Liu, and Qisheng Wang.
Space-bounded quantum state testing via space-efficient quantum singular value transformation.
*arXiv preprint arXiv:2308.05079*, 2023.

[GSTV11] Brady Garvin, Derrick Stolee, Raghunath Tewari, and N. V. Vinodchandran.
$\mathsf{ReachFewL} = \mathsf{ReachUL}$.
In Bin Fu and Ding-Zhu Du, editors, *Computing and Combinatorics*, pages 252–258, 2011.

[Lan97] Klaus-Jörn Lange.
An unambiguous class possessing a complete set.
In *STACS 97*. Springer Berlin Heidelberg, 1997.

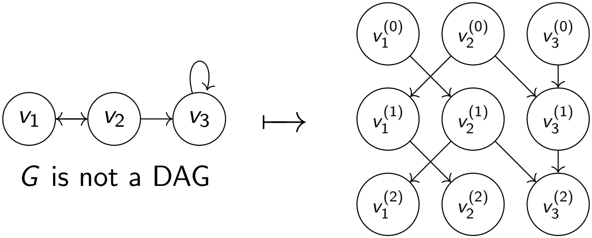
[Rei08] Omer Reingold.
Undirected connectivity in log-space.
*Journal of the ACM (JACM)*, 55(4):1–24, 2008.

[Sav70] Walter J Savitch.
Relationships between nondeterministic and deterministic tape complexities.
*Journal of computer and system sciences*, 4(2):177–192, 1970.

[TS13] Amnon Ta-Shma.
Inverting well conditioned matrices in quantum logspace.
In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, 2013.

[Wat99] John Watrous.
Space-bounded quantum complexity.
*Journal of Computer and System Sciences*, 59(2):281–326, 1999.

## DAG reduction




$G$ is not a DAG

$G'$ is a DAG