Quantum Communication Advantage in TFNP

Sid Jain

joint with

Mika Göös Tom Gur Jiawei Li

Communication Complexity

Communication Complexity

Alice

x

Bob

y

$f(x,y)$

Communication Complexity

Why study it?

Alice x

Bob y

$f(x,y)$

Communication Complexity

Alice
x

Bob
y

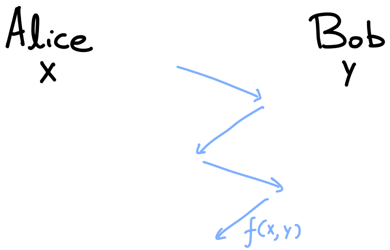
$f(x,y)$

Why study it?

→ expressive

circuits, streaming,
property testing,
time-space trade-offs,
query complexity

Communication Complexity



Why study it?

→ expressive

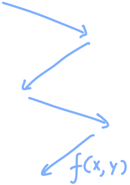
circuits, streaming, property testing, time-space trade-offs, query complexity

→ tractable

unconditional lower bounds for problems of interest

Communication Complexity

Alice	Bob
x	y



Models:

↳ Type
- Deterministic
- Randomized
- Quantum

↳ Interactivity
- SMP
- 1-way
- 2-way

Simultaneous Message Passing

Simultaneous Message Passing

$$
\begin{tikzcd}
\text{Alice} \\ x \arrow[dr, "|\psi_x\rangle"'] & & \text{Bob} \\ y \arrow[dl, "|\psi_y\rangle"] \\
& \text{Referee} &
\end{tikzcd}
$$

$f(x,y)$

Simultaneous Message Passing

Alice
x

Bob
y

$|\psi_x\rangle$

$|\psi_y\rangle$

Referee
$f(x,y)$

SMP $\preceq$ 1-way

Bob pretends to be the Referee.

Quantum Advantage

Goal: Design an experiment to demonstrate **unconditional** quantum advantage using Communication complexity.

Quantum Advantage

Two flavors:

Partial problems → promise on input

Total problems → NO promise

Quantum Advantage

Two flavors:

Partial problems $\longrightarrow$ promise on input

Total problems $\longrightarrow$ NO promise

Remark. Few separations for total problems

Impossible for query complexity of boolean fns

TFNP

A relation $R \subseteq X \times Y \times O$ is in communication-TFNP if

Totality: for all (x,y) there is a z s.t. $(x,y,z) \in R$

Verifiability: given (x,y,z) Alice and Bob can verify in $\mathrm{polylog}(|x|,|y|)$ communication if $(x,y,z) \in R$

Our Result

Our Result

$n = \mathrm{polylog}(N)$

There is a TFNP relation R s.t.

Alice $\quad\quad$ Bob

$x \quad\quad\quad\quad y$

$|\psi_x\rangle \quad\quad |\psi_y\rangle$

Referee

$(x, y, z) \in R$

$\#\text{qubits} = \mathrm{poly}(n)$

Our Result

$n = \mathrm{polylog}(N)$

There is a TFNP relation R s.t.

Alice x — Bob y

$|\psi_x\rangle$ — $|\psi_y\rangle$

Referee

$(x,y,z) \in R$

$\#\text{qubits} = \mathrm{poly}(n)$

yet

Alice x — Bob y

$(x,y,z) \in R$

$\#\text{bits} = 2^{n^{\Omega(1)}}$

Candidate problem	**Reference**	**Quantum u.b.**	**Classical l.b.**	**f / R**	**Totality**
Vector in Subspace	[Raz99, KR11]	**one-way**	**two-way**	**function**	**partial**
Gap Hamming Relation	[Gav21]	**SMP**	**two-way**	**relation**	**partial**
FORRELATION $\circ$ XOR	[GRT22]	**SMP**	**two-way**	**function**	**partial**
Hidden Matching	[BJK04]	**one-way**	**one-way**	**relation**	**total**
Lifted NULLCODEWORD	[YZ24a, GPW20]	**two-way**	**two-way**	**relation**	**total**
Bipartite NULLCODEWORD	This work	**SMP**	**two-way**	**relation**	**total**

Table 1: Several notable exponential quantum–classical separations. **Green text** indicates a strong result and **red text** indicates a weak result.

What's the problem?

Null Codeword

Yamakawa-Zhandry's relation

Null Codeword

Yamakawa-Zhandry's relation

Fix a code $C_n \subseteq \Sigma^n$

Notation: $H(x) = H_1(x_1) \cdots H_n(x_n)$, $H_i : \Sigma \to \{0,1\}$

Null Codeword

Yamakawa-Zhandry's relation

Fix a code $C_n \subseteq \Sigma^n$

Notation: $H(x) = H_1(x_1)\cdots H_n(x_n)$, $H_i : \Sigma \to \{0,1\}$

Then

$$\mathrm{NullCodeword}_n^C \subseteq \{0,1\}^{n|\Sigma|} \times C$$
$$= \{(H,c) \mid c \in C_n ,\ H(c) = 0^n\}$$

$$\begin{array}{ccccc} c_1 & c_2 \cdots & c_{\frac{n}{2}} & c_{\frac{n}{2}+1} \cdots & c_n \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

Null Codeword

Yamakawa-Zhandry's relation

Fix a code $C_n \subseteq \Sigma^n$

Notation: $H(x) = H_1(x_1) \cdots H_n(x_n)$, $H_i : \Sigma \to \{0,1\}$

Then

$$\mathrm{NullCodeword}_n^C \subseteq \{0,1\}^{n|\Sigma|} \times C$$
$$= \{(H,c) \mid c \in C_n,\ H(c) = 0^n\}$$

"Invert H on some codeword in C"

Null Codeword

Yamakawa-Zhandry's relation

Fix a code $C_n \subseteq \Sigma^n$

Notation: $H(x) = H_1(x_1) \cdots H_n(x_n)$, $H_i : \Sigma \to \{0,1\}$

Then

$$\mathrm{NullCodeword}_n^C \subseteq \{0,1\}^{n|\Sigma|} \times C$$
$$= \{(H,c) \mid c \in C_n, H(c) = 0^n\}$$

Yamakawa-Zhandry: if C is a certain folded Reed-Solomon code
H is uniform random

$$Q^{dt}(\mathrm{NullCodeword}) = O(n) \text{ yet } R^{dt}(\mathrm{NullCodeword}) = 2^{n^{\Omega(1)}}$$

Bipartite Nullcodeword

$$\underbrace{H_1 \; H_2 \cdots H_{\frac{n}{2}}}_{x} \; \underbrace{H_{\frac{n}{2}+1} \cdots H_n}_{y}$$

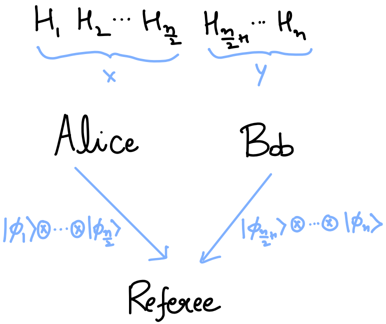
$$\underbrace{H_1 \; H_2 \cdots H_{\frac{n}{2}}}_{x} \;\; \underbrace{H_{\frac{n}{2}+1} \cdots H_n}_{y}$$

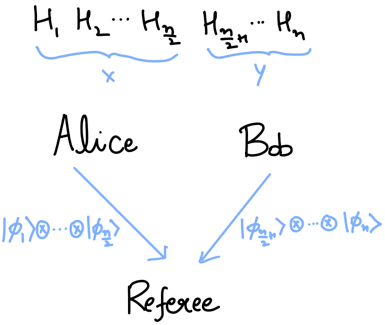
AKA

Alice Knows

$$\begin{array}{ccc|ccc} c_1 & \cdots & c_{\frac{n}{2}} & c_{\frac{n}{2}+1} & \cdots & c_n \\ \downarrow & & \downarrow & \downarrow & & \downarrow \\ 0/1 & & 0/1 & 0/1 & & 0/1 \end{array}$$

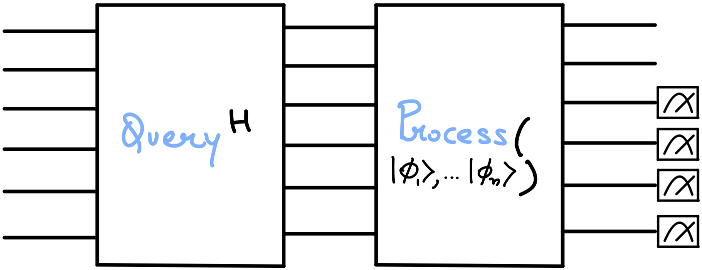
Bob Knows


$H_1 \; H_2 \cdots H_{\frac{n}{2}}$
$H_{\frac{n}{2}+1} \cdots H_n$
x
y
Alice
Bob
$|\phi_1\rangle \otimes \cdots \otimes |\phi_{\frac{n}{2}}\rangle$
$|\phi_{\frac{n}{2}+1}\rangle \otimes \cdots \otimes |\phi_n\rangle$
Referee

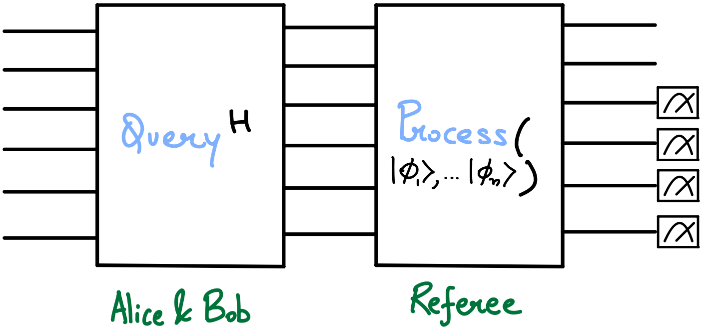


$$|\phi_i\rangle \propto \sum_{\substack{a \in \Sigma : \\ H_i(a)=0}} |a\rangle$$

Yamakawa-Zhandry algorithm



Yamakawa-Zhandry algorithm



Classical LB

Intuition A good code is pseudorandom.

Moreover, $\Pr[H(c) = 0^n] = 2^{-n}$.

Classical LB

Intuition A good code is pseudorandom.

Moreover, $\Pr[H(c) = 0^n] = 2^{-n}$.

→ Every codeword is unlikely to be a sol^n.

→ Querying one codeword does not reveal much information about many other codewords.

Classical LB

List-Recoverability, Simplified $C \subseteq \Sigma^n$ is l.r.

if for any $S_1, S_2, \ldots S_n \subseteq \Sigma$ s.t. $\sum_i |S_i| \leq \ell$,

$$\left| \{ (x_1, \ldots x_n) \in C : |\{ i \in [n] : x_i \in S_i \}| \geq 0.4n \} \right| \leq 2^{o(n)}$$

Classical LB

List-Recoverability, Simplified $C \subseteq \Sigma^n$ is ℓ.r.

if for any $S_1, S_2, \dots S_n \subseteq \Sigma$ s.t. $\sum_i |S_i| \leq \ell$,

$$\left| \{ (x_1, \dots x_n) \in C : |\{ i \in [n] : x_i \in S_i \}| \geq 0.4n \} \right| \leq 2^{o(n)}$$

$\sum_i |S_i| \leftarrow$ number of inputs bits revealed

Subcube Protocols

Defn $X \subseteq \{0,1\}^N$ is a subcube if $\exists\, I \subseteq [n]$
$X_I := \{ x_I \in \{0,1\}^{|I|} : x \in X \} = \{\alpha\}$
and $X_{\bar{I}}$ contains all possible strings

Subcube Protocols

Defn $X \subseteq \{0,1\}^N$ is a subcube if $\exists\, I \subseteq [n]$

$X_I := \{ x_I \in \{0,1\}^{|I|} : x \in X \} = \{\alpha\}$

and $X_{\bar{I}}$ contains all possible strings

Defn A protocol Π is a subcube protocol if

for every $v \longleftrightarrow R_v = X \times Y$

X, Y are subcubes

~~Subcube Protocols~~ Decision Trees?

~~Subcube Protocols~~ Decision Trees?

No, they are more expressive.

~~Subcube Protocols~~ Decision Trees?

No, they are more expressive.

Alice $\quad\quad\quad\quad\quad\quad\quad\quad$ Bob

$x \in \{0,1\}^N \xrightarrow{i}$

length = $\log(N+1)$

$N+1$ subcubes

$i \in [N]$ s.t. $x_i = 1$

$x_j = 0 \ \forall j < i$

~~Subcube Protocols~~ Decision Trees?

No, they are more expressive.

Alice $\qquad\qquad\qquad$ Bob

$x \in \{0,1\}^N \xrightarrow{\quad i \quad}$

$i \in [N]$ s.t. $x_i = 1$
$x_j = 0 \;\forall\, j < i$

length $= \log(N+1)$

$N+1$ subcubes

Can't be efficiently simulated by queries/decision trees!

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

Say that $x \in C$ is dangerous for rectangle R if

$$\left| \{ i \in [n] : x_i \text{ is fixed in } R \} \right| \leq 0.4n$$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

Say that $x \in C$ is dangerous for rectangle R if

$$\left| \{ i \in [n] : x_i \text{ is fixed in } R \} \right| \leq 0.4n$$

By list-recoverability, if $|\Pi| = o(\ell)$ then

$$\# \text{dangerous } x \in C \leq 2^{o(n)}$$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability, if $|\Pi| = o(\ell)$ then

$$\#\text{dangerous } x \in C \leq 2^{o(n)}$$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability, if $|\Pi| = o(\ell)$ then

$$\#\text{ dangerous } x \in C \leq 2^{o(n)}$$

Say $x \in C$ becomes dangerous when Alice speaks at v

x has at least $0.1n$ unfixed bits in Bob's half of R_v

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability, if $|\Pi| = o(\ell)$ then

$$\#\text{ dangerous } x \in C \leq 2^{o(n)}$$

Say $x \in C$ becomes dangerous when Alice speaks at v

x has at least $0.1n$ unfixed bits in Bob's half of R_v

$$\Pr\left[H(x) = 0^n \,\middle|\, H \in R_v \right] \leq 2^{-0.1n}$$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability, if $|\Pi| = o(\ell)$ then

$$\#\text{ dangerous } x \in C \leq 2^{o(n)}$$

By union bound, the chance of any dangerous x sol^n

$$\Pr\left[\exists \text{ dangerous } x,\ H(x) = 0^n \,\middle|\, H \in R_0\right] \leq 2^{-0.1n}\, 2^{o(n)}$$

$$= 2^{-\Omega(n)}$$

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability,

We need parameters stronger
than we can prove for the
YZ code

Lower Bound for Subcube Protocols

Any subcube protocol solving BiNC has complexity $\Omega(\ell)$

By list-recoverability,

We need parameters stronger than we can prove for the YZ code

(*) we generalize to a p-biased input distribution to tradeoff upper and lower bounds

Lower Bound

- how can we lift the lower bound for Subcube Protocals?

Lower Bound

- how can we lift the lower bound for Subcube Protocals?

Structure-vs-Randomness

Lower Bound

- how can we lift the lower bound for Subcube Protocals?

Structure-vs-Randomness

- how do we convert to a total relation?

Lower Bound

- how can we lift the lower bound for Subcube Protocals?

Structure-vs-Randomness

- how do we convert to a total relation?

employ trick: find short certificates → TFNP

Thanks

for your

attention!