Coding for Interactive Communication

Ran Gelles
Princeton University
Coding for Interactive Communication

How to make conversations resilient to noise?

Ran Gelles
Princeton University
Why Interaction?
Why Interaction?
Why Interaction?
Why Interaction?
Why Interaction?
Why Interaction?

How does it work? - Interaction!
Why Interaction?

How does it work? - Interaction!
Why Interaction?

How does it work? - Interaction!
Why Interaction?

How does it work? - Interaction!
Why Interaction?

How does it work? - Interaction!
Without Interaction?

- Without interaction: Need to communicate the entire map
Without Interaction?
Without Interaction?
Without Interaction?
Without Interaction?
Without Interaction?
Without Interaction?

- Without interaction: Need to communicate the entire set of preferences
Why Interaction?
Why Interaction?

- Interactions are *everywhere!*
Why Interaction?

- Interactions are everywhere!
- Interaction makes a conversation short
  - No needless information
  - Can be exponentially faster
  - Saves time and energy!
The “Bad Guy”
The “Bad Guy”

Communication is Noisy!
Motivation

The Question:

How to error-correct conversations?!
Talk Outline

• Part 1: Coding in the Interactive Setting
• Part 2: The 2-party case
• Part 3: The multiparty case
• Part 4: Applications
Interactive Communication

\[ f(x, y) \] (output) \[ f(x, y) \]

\[ r \text{ rounds} \]
Noisy Interactive Communication

\[ \pi(x, y) \]

output: \( \pi(x, y) \)
Noisy Interactive Communication

$x$  

$y$  

\[ \vdots \]

$r$ rounds

output: ???
Coding for Interactive Comm.

\[ \pi(x, y) \]

\[ \pi'(x, y) = \pi(x, y) \]
Coding for Interactive Comm.

\[ \pi \]

\[ \pi(x,y) \]

\[ \pi' \]

\[ \pi'(x,y) = \pi(x,y) \]

The entire transcript

\( r \) rounds

\( R \) rounds
“Good Coding” Criteria

- Computation Efficiency (time)
- Noise Resilience
- Rate
“Good Coding” Criteria

- Efficiency
- Resilience
- Rate
“Good Coding” Criteria

- Efficiency: Efficient (pref. linear-time)
- Resilience
- Rate
“Good Coding” Criteria

- Efficiency: Efficient (pref. linear-time)
- Resilience: Success Prob: $1-2^{-\Omega(R)}$
- Rate
“Good Coding” Criteria

- **Efficiency**
  - Efficient (pref. linear-time)

- **Resilience**
  - Success Prob: $1-2^{-\Omega(R)}$

- **Rate**
  - $\lim_{r \to \infty} \frac{r}{R} > 0$
“Good Coding” Criteria

Efficiency

Efficient \ (\text{pref. linear-time})

Resilience

Success Prob: $1 - 2^{-\Omega(R)}$

Rate

\[
\lim_{r \to \infty} \frac{r}{R} > 0
\]

\[
R = O(r) \iff \frac{r}{R} = O(1)
\]
Simple Solutions?
Simple Solutions?

• Use Error-Correcting Codes!
Simple Solutions?

• Use Error-Correcting Codes!
• Encoding each message?
Simple Solutions?

• Use Error-Correcting Codes!
• Encoding each message?
  • 1 bit becomes $k = O(1)$ bits
Simple Solutions?

- Use Error-Correcting Codes!
- Encoding each message?
  - 1 bit becomes $k = O(1)$ bits
  - decoding still fails with const prob $2^{-O(k)}$
Simple Solutions?

- Use Error-Correcting Codes!
- Encoding each message?
  - 1 bit becomes $k = O(1)$ bits
  - decoding still fails with const prob $2^{-O(k)}$
  - $\approx r \cdot 2^{-O(k)}$ messages fail in expectation
Simple Solutions?

• Use Error-Correcting Codes!

• Encoding each message?
  • 1 bit becomes $k=O(1)$ bits
  • decoding still fails with const prob $2^{-O(k)}$
  • $\approx r \cdot 2^{-O(k)}$ messages fail in expectation

• To improve failure prob, $k$ must be super-constant
Simple Solutions?

- Use Error-Correcting Codes!
- Encoding each message?
  - 1 bit becomes $k = O(1)$ bits
  - decoding still fails with const prob $2^{-O(k)}$
  - $\approx r \cdot 2^{-O(k)}$ messages fail in expectation
- To improve failure prob, $k$ must be super-constant

  ➤ bad resilience / bad communication!
Timeline
Timeline

- 1990's: [Schulman92, Schulman93, RajagopalanSchulman94]
Timeline

- 1990’s: [Schulman92, Schulman93, RajagopalanSchulman94]

- 2011: [BravermanRao11, GMoitraSahai11]
Timeline

- **1990's**: [Schulman92, Schulman93, RajagopalanSchulman94]

- **2011**: [BravermanRao11, GSmoitraSahai11]

Timeline

• 1990’s: [Schulman92, Schulman93, RajagopalanSchulman94]

• 2011: [BravermanRao11, GMoitraSahai11]

• 2012-2015: 
Timeline

- Players: 2-party, multiparty

• 2012-2015:
  [Braverman12, BrakerskiKalai12, KalaiLewkoRao12, BrakerskiNaor13, KolRaz13,
   FranklinG OstrovskySchulman13, AgrawalG Sahai13, ChungPassTelang13, Pankratov13
   GSahaiWadia14, BrassardNayakTappTouchetteUnger14, DodisLewko14,
   GhaffariHaeuplerSudan14, GhaffariHaeupler14, BravermanEfremenko14, Haeupler14,
   GHaeupler15, JainKalaiLewko15, EfremenkoGHaeupler15,
   AlonBravermanEfremenkoGHaeupler15, BravermanG MaoOstrovsky15,
   BravermanEfremenkoGHaeupler15, GHaeuplerKolRonZewiWigderson16, HozaSchulman16]
Timeline

- **Players:** 2-party, multiparty
- **Noise:** random, adversarial, erasures, indels, ...

- **2012-2015:**
Timeline

- Players: 2-party, multiparty
- Noise: random, adversarial, erasures, indels, ...
- Extensions: list decoding, adaptivity, ...

2012-2015:

[Braverman12, BrakerskiKalai12, KalaiLewkoRao12, BrakerskiNaor13, KolRaz13,
FranklinGOstrovskySchulman13, AgrawalGSahai13, ChungPassTelang13, Pankratov13,
GSahaiWadia14, BrassardNayakTappTouchetteUnger14, DodisLewko14,
GhaaffariHaeuplerSudan14, GhaaffariHaeupler14, BravermanEfremenko14, Haeupler14,
GHaeupler15, JainKalaiLewko15, EfremenkoGHaeupler15,
AlonBravermanEfremenkoGHaeupler15, BravermanGMaoOstrovsky15,
BravermanEfremenkoGHaeupler15, GHaeuplerKolRonZewiWigderson16, HozaSchulman16]
Timeline

- Players: 2-party, multiparty
- Noise: random, adversarial, erasures, indels, ...
- Extensions: list decoding, adaptivity, ...

• 2012-2015:
  [Braverman12, BrakerskiKalai12, KalaiLewkoRao12, BrakerskiNaor13, KolRaz13,
  FranklinGOSahai13, AgrawalGGSahai13, ChungPassTelang13, Pankratov13
  GSahaiWadia14, BrassardNayakTappTouchetteUnger14, DodisLewko14,
  GhaffariHaeuplerSudan14, GhaffariHaeupler14, BravermanEfremenko14, Haeupler14,
  GHaeupler15, BravermanEfremenkoGHaeupler15, GHaeuplerKolRonZewiWigderson16, HozaSchulman16]

Talk Outline

• Part 1: Coding in the Interactive Setting

• Part 2: The 2-party case

• Part 3: The multiparty case

• Part 4: Applications
Interactive Protocol
Interactive Protocol
Interactive Protocol
Interactive Protocol

Alice

Bob

Alice
Noisy Interactive Protocol
Noisy Interactive Protocol

Alice

Bob

Alice
Noisy Interactive Protocol
Noisy Interactive Protocol

Alice

Bob

Alice
Noisy Interactive Protocol

Need a way to verify consistency
Tree codes Encoding

[Schulman93]
Tree codes Encoding

[Schulman93]
Tree codes Encoding

[Schulman93]

\[ l_1 \]

\[ l_{10} \]
Tree codes Encoding

[Schulman93]
Tree codes Encoding

[Schulman93]

Depend on the entire path!
Tree codes Encoding

[Schulman93]

• In order to be useful:
labels must behave like a good code:

• Labels along any two divergent paths need to have large Hamming distance

(below the point of divergence)
Tree codes Encoding noisy example

Alice

Bob

Alice
Tree codes Encoding

noisy example

Alice

Bob

Alice
Tree codes Encoding
noisy example
Tree codes Encoding
noisy example

Alice: something is wrong here, expecting $l_{10}$ or $l_{11}$!
Tree codes
Tree codes

- Each label $l_{\text{path}}$ must be short $|l_{\text{path}}| = O(1)$ (otherwise, the rate will not be constant)
Tree codes

- Each label $l_{\text{path}}$ must be short \( |l_{\text{path}}| = O(1) \) (otherwise, the rate will not be constant)

- [Schulman93]: Infinite tree codes exist
Tree codes

- Each label $l_{\text{path}}$ must be short $|l_{\text{path}}| = O(1)$ (otherwise, the rate will not be constant)

- [Schulman93]: Infinite tree codes exist

\[ \text{takes } 2^{O(N)} \text{ time to construct!} \]
Tree codes

• Each label $l_{\text{path}}$ must be short $|l_{\text{path}}|=O(1)$ (otherwise, the rate will not be constant)

• [Schulman93]: Infinite tree codes exist

• But how to construct them efficiently?
Tree codes

- Each label $l_{\text{path}}$ must be short $|l_{\text{path}}| = O(1)$ (otherwise, the rate will not be constant)

- [Schulman93]: Infinite tree codes exist

- But how to construct them efficiently?

Even a randomized construction almost surely fails!
Tree codes

- Each label $l_{\text{path}}$ must be short $|l_{\text{path}}| = O(1)$ (otherwise, the rate will not be constant)

- [Schulman93]: Infinite tree codes exist

- But how to construct them efficiently?
Tree codes

- Each label $l_{\text{path}}$ must be short $|l_{\text{path}}| = O(1)$ (otherwise, the rate will not be constant)

- [Schulman93]: Infinite tree codes exist

- But how to construct them efficiently?
Potent Tree Codes

• Replace tree code with a potent relaxation:

  • *Not all* paths have large Hamming distance, but *most* of them do.
Potent Tree Codes

“potent” tree-code
Potent Tree Codes

“A colliding path is “equivalent” to additional noise

“potent” tree-code

[GMoitraSahai11]
Potent Tree Codes
Potent Tree Codes

• Since potent trees are more relaxed:
  
  • a randomized **efficient** construction exists
  
  • succeeds with prob $1 - 2^{-\Omega(N)}$
Potent Tree Codes

[GMoitraSahai11]
Potent Tree Codes

- Potent trees + techniques from [Schulman93] give:

  - Efficiency
  - Resilience
  - Rate
Potent Tree Codes

- Potent trees + techniques from [Schulman93] give:

  - Efficiency
  - Resilience
  - Rate
  - efficient
Potent Tree Codes

- Potent trees + techniques from [Schulman93] give:

  - Efficiency: **efficient**
  - Resilience: success prob: \(1 - 2^{-\Omega(R)}\)
  - Rate: over a BSC_\(\varepsilon\)
Potent Tree Codes

- Potent trees + techniques from [Schulman93] give:

  - **Efficiency**: efficient

  - **Resilience**: success prob: \(1 - 2^{-\Omega(R)}\)
    over a BSC_\(\varepsilon\)

  - **Rate**: \(\frac{r}{R} = O(1)\)
Potent Tree Codes

- Potent trees + techniques from [Schulman93] give:

  - **Efficiency**: efficient
  - **Resilience**: success prob: $1 - 2^{-\Omega(R)}$
    over a BSC$_\varepsilon$
  - **Rate**: $\frac{r}{R} = O(1)$

- Potent tree codes can be used in other schemes:
  [RajagopalanSchulman94, BravermanRao11, etc.]
Determinism?
Determinism?

• All efficient coding schemes (with good parameters) are randomized

• Can we get a scheme with good parameters which is both efficient and deterministic?
Towards a Deterministic Scheme

A new coding scheme achieves:

- Efficiency
- Resilience
- Rate

[GHaeuplerKolRonZewiWigderson16]
Towards a Deterministic Scheme

- A new coding scheme achieves: efficient and deterministic

[GHaeuplerKolRonZewiWigderson16]
Towards a Deterministic Scheme

A new coding scheme achieves: efficient and deterministic

- Efficiency
- Resilience
- Rate

success prob: \(1 - 2^{-\Omega(R/\log R)}\)

over a BSC_\varepsilon
Towards a Deterministic Scheme

A new coding scheme achieves:

- **Efficiency**: efficient and **deterministic**
- **Resilience**: success prob: $1 - 2^{-\Omega(R/\log R)}$ over a $\text{BSC}_\varepsilon$
- **Rate**

[GHaeuplerKolRonZewiWigderson16]
Towards a Deterministic Scheme

- A new coding scheme achieves:

  - Efficiency: efficient and **deterministic**
  - Resilience: success prob: $1 - 2^{-\Omega(R/\log R)}$
  - Rate: over a BSC$_\varepsilon$
  \[
  \frac{r}{R} = 1 - O(\sqrt{H(\varepsilon)})
  \]

[GHaeuplerKolRonZewiWigderson16]
Towards a Deterministic Scheme

[GHaeuplerKolRonZewiWigderson16]
Towards a Deterministic Scheme

• Based on ideas from one-way *concatenation codes*

[GHaeuplerKolRonZewiWigderson16]

[Forney65]
Towards a Deterministic Scheme

Based on ideas from one-way *concatenation codes*

Efficient and deterministic construction of a tree code (relaxation) of depth $R$:

- Large alphabet, yet small rate
  \[ \Sigma_{in} = \{0, 1\}^{\log R} \quad \Sigma_{out} = \{0, 1\}^{(1+\varepsilon) \log R} \]

- Distance: $O(1/\log R)$
Towards a Deterministic Scheme

- Based on ideas from one-way *concatenation codes* [Forney65]
- Efficient and deterministic construction of a tree code (relaxation) of depth $R$:
  - Large alphabet, yet small rate
    $$\Sigma_{in} = \{0, 1\}^{\log R} \quad \Sigma_{out} = \{0, 1\}^{(1+\varepsilon) \log R}$$
  - Distance: $O(1/\log R)$
Towards a Deterministic Scheme

- Based on ideas from one-way *concatenation codes* [Forney65]
- Efficient and deterministic construction of a tree code (relaxation) of depth $R$:
  - Large alphabet, yet small rate
    $$\Sigma_{in} = \{0, 1\}^{\log R} \quad \Sigma_{out} = \{0, 1\}^{(1+\varepsilon) \log R}$$
  - Distance: $O(1/\log R)$
• Part 1: Coding in the Interactive Setting
• Part 2: The 2-party case
  • Part 3: The multiparty case
• Part 4: Applications
Multiparty Coding
Multiparty Coding

• Modern systems: everybody talks with everybody!
Multiparty Coding

• Modern systems: everybody talks with everybody!

• 2-party solutions do not generalize:
  • An error at one party affects the entire network
  • Even when each channel is encoded via a 2-party scheme
Multiparty Coding

- A network $G=(V,E)$ with $n=|V|$ parties, connected in an **arbitrary** topology.
- Each node gets an input $x_i$.
- The goal: compute $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$. 
Multiparty Coding

- A network $G=(V,E)$ with $n=|V|$ parties, connected in an \textit{arbitrary} topology
- Each node gets an input $x_i$
- The goal: compute $f(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$.

- A \textit{synchronous} protocol:
  - Each \textit{round}, each party sends a single bit on each edge connected to it.
Multiparty Coding

- A coding scheme by [RajagopalanSchulman94] achieves:
  - Efficiency
  - Resilience
  - Rate
Multiparty Coding

- A coding scheme by [RajagopalanSchulman94] achieves:

  - **Efficiency**: inefficient
  - **Resilience**: (efficient version via potent tree codes [GMoitraSahai11])
  - **Rate**
Multiparty Coding

- A coding scheme by [RajagopalanSchulman94] achieves:

  - **Efficiency**: inefficient
    (efficient version via potent tree codes [GMoitraSahai11])
  - **Resilience**: success prob: $1 - n2^{-\Omega(R)}$
Multiparty Coding

- A coding scheme by [RajagopalanSchulman94] achieves:
  - **Efficiency**: inefficient
    (efficient version via **potent** tree codes [GoMoitraSahai11])
  - **Resilience**: success prob: $1 - n2^{-\Omega(R)}$
  - **Rate**: $O(1/\log d)$, $d = \text{maximal degree in } G$
Multiparty Coding

- A coding scheme by [RajagopalanSchulman94] achieves:

  - **Efficiency**
    - inefficient (efficient version via potent tree codes [GMoitraSahai11])

  - **Resilience**
    - success prob: $1 - n2^{-\Omega(R)}$

  - **Rate**
    - $O(1/\log d)$ \quad d = \text{maximal degree in } G$
    \quad \Rightarrow O(1/\log n)$ \quad \text{e.g., complete graph}$
Multiparty Coding
Multiparty Coding

• Can rate $O(1)$ be achieved for $G$ with degree $\Omega(n)$?
Multiparty Coding

- Can rate $O(1)$ be achieved for $G$ with degree $\Omega(n)$?
- **YES** (sometimes)

$O(1)$ scheme for complete graphs (and others)

[AlonBravermanEfremenkoGHaupepler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]

- **Theorem** (following [RajagopalanSchumlan94]):

  $O(1)$ simulation of any $r$-round protocol reduces to:

  simulating a **single** noiseless round with high probability, using $O(1)$ noisy rounds.
Multiparty O(1) Coding
[AlonBravermanEfremenkoGHauepler15]
Multiparty O(1) Coding

[AlonBravermanEfremenkoGHaeuepler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHaeupler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHaeuepler15]
Multiparty O(1) Coding

[AlonBravermanEfremenkoGHaupepler15]
Multiparty $O(1)$ Coding

Directly:
- $\Omega(\log n)$ times
- Prob. $1 - 2^{-\Omega(\log n)}$
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]

• Directly:
  • $\Omega(\log n)$ times
  • Prob. $1-2^{-\Omega(\log n)}$
Multiparty O(1) Coding

[Ronen Alon, Shay Braverman,羿斐伦, Huanyu Fu, Gill Hauepler 2015]

- Relay
Multiparty $O(1)$ Coding

- Relay

[AlonBravermanEfremenkoGHauepler15]
Multiparty $O(1)$ Coding

- Relay
Multiparty O(1) Coding

- Relay:
  - $O(1)$ rounds
  - Prob. $1 - 2^{-\Omega(n)}$
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]

• Can we send $n$ bits in $O(1)$ rounds?
Multiparty O(1) Coding

[AlonBravermanEfremenkoGHaepler15]

• Can we send $n$ bits in $O(1)$ rounds?

• **YES**, as simple as sending 1 bit:
  
  • use error-correcting code (ECC) to obtain $O(n)$ bits, such that decoding succeeds w.p. $1 - 2^{-\Omega(n)}$ even if each bit is $\epsilon$-noisy  
    
    [Shannon’48]

• relay through the network
  (each party relays $|\text{ECC}(\text{msg})|/n$ bits)
Multiparty O(1) Coding

[AlonBravermanEfremenkoGHauepler15]
Multiparty $O(1)$ Coding

[AlonBravermanEfremenkoGHauepler15]

- When $p_1$ talks with $p_2$, only $n$ links in $G$ are used.
Multiparty $O(1)$ Coding

$\text{[AlonBravermanEfremenko\textsubscript{G}Hauepler15]}$

- When $p_1$ talks with $p_2$, only $n$ links in $G$ are used.

- But the network $G$ has $\approx n^2$ links!

$\Rightarrow$ All parties communicate in parallel, each party is source/target at most once
Multiparty O(1) Coding

• When $p_1$ talks with $p_2$, only $n$ links in $G$ are used.

• But the network $G$ has $\approx n^2$ links!

  All parties communicate in parallel, each party is source/target at most once

• Using the above idea on “subnetworks” of size $\sqrt{n}$ allows all parties to deliver 1 bit to neighbors in $O(1)$ rounds with high probability $1-2^{-\Omega(\sqrt{n})}$
O(1) coding for any topology?

• Can rate $O(1)$ be achieved for all $G$ with degree $\Omega(n)$?
O(1) coding for any topology?

- Can rate $O(1)$ be achieved for all $G$ with degree $\Omega(n)$?
  - **NO**
O(1) coding for any topology?

• Can rate $O(1)$ be achieved for all $G$ with degree $\Omega(n)$?

• NO

$\tilde{O}(1/\log n)$-outer bound for star graphs

[BravermanEfremenkoGHauhepler15]
$\tilde{O}(1/\log n)$-outer bound over star
\( \tilde{O}(1/\log n) \)-outer bound over star

- Why previous ideas don’t work for all topologies?
  - [RS94] Direct coding fails when degree is large
  - [ABEGH15] Relay fails when no multiple (short) paths
\( \tilde{O}(1/\log n) \)-outer bound over star

- Why previous ideas don’t work for all topologies?
  - [RS94] Direct coding fails when degree is large
  - [ABEGH15] Relay fails when no multiple (short) paths

- Bad topology:
\(\tilde{O}(1/\log n)-\text{outer bound over star}\)

**Theorem:**
Any coding scheme for \(r\)-round noiseless protocol over a star network with \(n\) parties, must take

\[ R \geq \Omega \left( r \frac{\log n}{\log \log n} \right) \]

rounds to succeed with high probability.

[BravermanEfremenkoGHaupepler15]
Theorem:
Any coding scheme for $r$-round noiseless protocol over a star network with $n$ parties, must take
\[ R \geq \Omega \left( r \frac{\log n}{\log \log n} \right) \] rounds to succeed with high probability.
\( \tilde{O}(1/\log n) \)-outer bound over star

- **Theorem:**
  Any coding scheme for \( r \)-round noiseless protocol over a star network with \( n \) parties, must take
  \[
  R \geq \Omega \left( r \frac{\log n}{\log \log n} \right)
  \]
  rounds to succeed with high probability.

- \([RS94]\) Rate \( \frac{r}{R} = O \left( \frac{\log \log n}{\log n} \right) \)

\[\tilde{O}(1/\log n) \]-outer bound over star network.
\(\tilde{O}(1/\log n)\)-outer bound over star

[BravermanEfremenko\(G\)Hauepler15]

- **Theorem:** Any coding scheme for \(r\)-round noiseless protocol over a star network with \(n\) parties, must take

\[ R \geq \Omega \left( r \frac{\log n}{\log \log n} \right) \]

rounds to succeed with high probability.

- In fact, holds for *erasure* channels!
\(\tilde{O}(1/\log n)\)-outer bound over star

\[ p_1 \quad p_2 \quad p_3 \]

[BravermanEfremenkoGHaeuepler15]
$\tilde{O}(1/\log n)$-outer bound over star

$\begin{align*}
p_1 & \quad 100\overline{0}100 \\
p_2 & \quad 010\overline{0}10 \\
p_3 & \quad 1\overline{0}0\overline{0}10
\end{align*}$

[BravermanEfremenkoGHauhepler15]
\[\tilde{O}(1/\log n)\text{-outer bound over star}\]

\[\begin{array}{c}
p_1 & 10\emptyset100 \\
p_2 & 010\emptyset10 \\
p_3 & 1\emptyset0\emptyset10 \\
\end{array}\]

[BravermanEfremenkoGHauepler15]
\[\tilde{O}(1/\log n)\text{-outer bound over star}\]

\[0.1 \cdot \log n\]

\begin{align*}
p_1 & \quad 10\emptyset10010 \\
p_2 & \quad 010\emptyset10\emptyset \emptyset \\
p_3 & \quad 1\emptyset0\emptyset10\emptyset \emptyset
\end{align*}

rounds
$\tilde{O}(1/\log n)$-outer bound over star

0.1 \cdot \log n

$p_1 \quad 1\emptyset 10010$
$p_2 \quad 010\emptyset 1\emptyset \emptyset$
$p_3 \quad 1\emptyset 0\emptyset 10\emptyset \emptyset$

rounds

[BravermanEfremenkoGHaeupler15]
\[\tilde{O}(1/\log n)\)-outer bound over star

\[0.1 \cdot \log n\]

\[\sqrt{n} \text{ parties are completely erased (w.h.p):}\]

\[
\begin{array}{c}
p_1 \quad 10010010 \\
p_2 \quad 01001000 \\
p_3 \quad 10001000 \\
\end{array}
\]
\( \tilde{O}(1/\log n) \)-outer bound over star

\[ t \]

\( \sqrt{n} \) parties are \textit{completely} erased (w.h.p):

- single party erased w.p. \( \varepsilon^{0.1 \log n} \approx 1/\sqrt{n} \)
\(\tilde{O}(1/\log n)\)-outer bound over star

- \(0.1 \cdot \log n\) parties are completely erased (w.h.p):
  - single party erased w.p. \(\varepsilon^{0.1 \log n} \approx 1/\sqrt{n}\)
  - \(\approx \sqrt{n}\) parties are erased in expectation

[BravermanEfremenkoGHauepler15]
$\tilde{O}(1/\log n)$-outer bound over star

Small information on correct continuation (even given history)

$p_1 \ 10010010$
$p_2 \ 01001000$
$p_3 \ 10001000$

0.1 \cdot \log n$

$\Omega(1/\log n)$-outer bound over star

[BravermanEfremenkoGHaeupler15]
\( \tilde{O}(1/\log n) \)-outer bound over star

\[ O(\log \log n) \]

\[ 0.1 \cdot \log n \]

[BravermanEfremenkoGHauepler15]
Talk Outline

• Part 1: Coding in the Interactive Setting
• Part 2: The 2-party case
• Part 3: The multiparty case
• Part 4: Applications
Summary so far
Summary so far

• Coding for Interactive Communication
Summary so far

• Coding for Interactive Communication

• 2-Party:
  - efficient coding schemes
  - deterministic constructions
Summary so far

• Coding for Interactive Communication

• 2-Party:
  - efficient coding schemes
  - deterministic constructions

• Multiparty:
  - efficient coding schemes
  - upper and lower bounds on the rate
Summary so far

• Coding for Interactive Communication

• 2-Party:
  - efficient coding schemes
  - deterministic constructions

• Multiparty:
  - efficient coding schemes
  - upper and lower bounds on the rate

• Applications?
Applications 1
Applications 1

• Noise-Resilient Circuits [KalaiLewkoRao12]
Applications 1

• Noise-Resilient Circuits [KalaiLewkoRao12]
Applications 1

- Noise-Resilient Circuits [KalaiLewkoRao12]

[KarchmerWigderson90]
Applications 1

• Noise-Resilient Circuits  [KalaiLewkoRao12]

[KarchmerWigderson90]
Applications 1

- Noise-Resilient Circuits  
  \[\text{KalaiLewkoRao12}\]

\[\text{KarchmerWigderson90}\]
Applications 1

• Noise-Resilient Circuits  [KalaiLewkoRao12]

Setting:
• players: 2-party
• channel: BSC w/ feedback
• noise: adversarial

[KalaiLewkoRao12]
[GHaeupler15]
[EfremenkoGHaeupler15]
Applications 2
Applications 2

• Noise-Resilient *data stream* communication

[FranklinG.OstrovskySchulman13]
Applications 2

• Noise-Resilient *data stream* communication

[FranklinG OostrovskySchulman13]

• Allows communicating big (infinite) data stream
Applications 2

• Noise-Resilient *data stream* communication

  [Franklin, G. Ostrovsky, S. Schulman 2013]

• Allows communicating big (infinite) data stream

• Robust to a fraction $1-\varepsilon$ of adversarial noise
Applications 2

- Noise-Resilient *data stream* communication

- Allows communicating big (infinite) data stream

- Robust to a fraction $1 - \varepsilon$ of adversarial noise

- Efficient, randomized, success prob $1 - 2^{-\Omega(|S|)}$ assuming a shared key

[Franklin GOstrovskySchulman13]
Applications 3
Applications 3

• Private noisy interactive communication?
Applications 3

- Private noisy interactive communication?

- **Impossibility:** (adversarial noise)
  no coding scheme can be both *private* and *noise-resilient*

[ChungPassTelang13]    [GSahaiWadia14]
Interactive Coding

• Young and Active Field!


• Survey: [G15]
Interactions are welcome!