Some recent results on high rate local codes

Shubhangi Saraf

Rutgers

Joint works with

Sivakanth Gopi, Swastik Kopparty, Or Meir, Rafael Oliveira, Noga Ron-Zewi, Mary Wootters

This talk

- Error-correcting codes with:
  - low redundancy
  - robust to large fraction of errors
  - *sublinear time* error-detection and error-correction algorithms

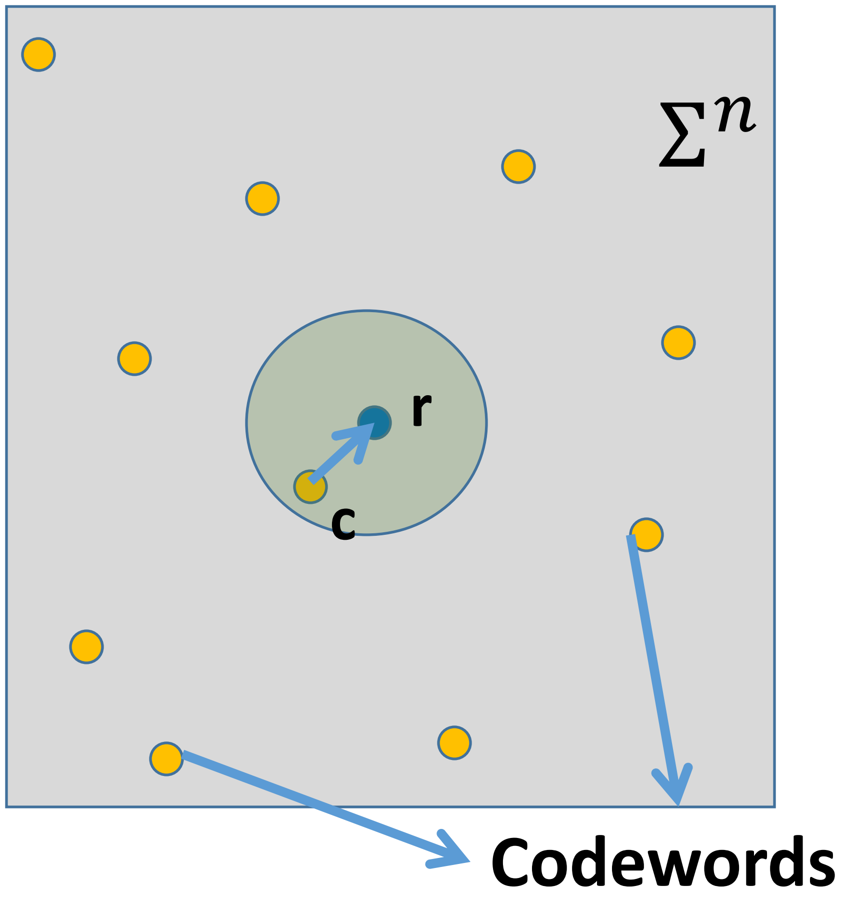
Error-correcting codes

- Alphabet Σ (often {0,1})

- Encoding:
  - E: $\Sigma^k \rightarrow \Sigma^n$
  - Maps data to "codeword"

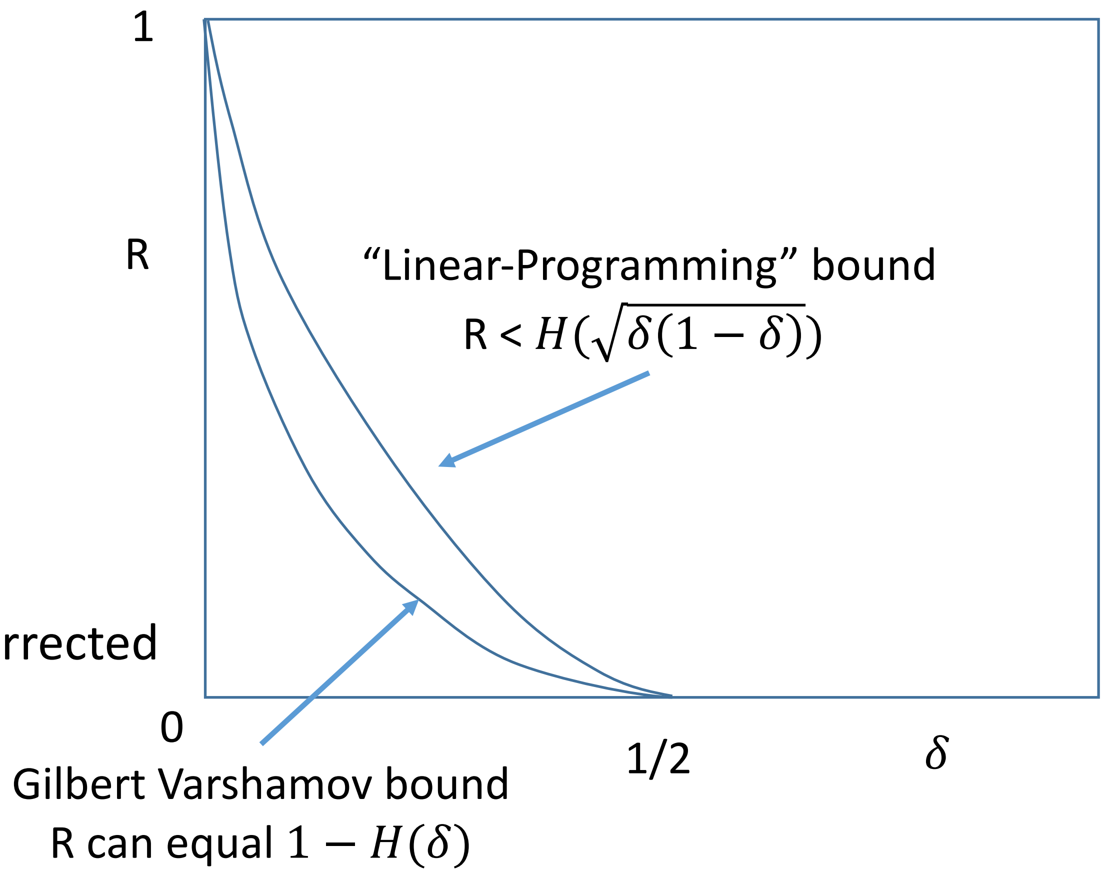
- Code C = Image(E)
  - Rate = k/n
  - (Hamming) Distance δ:
    Any 2 codewords differ on at least δ fraction coordinates,

$\frac{\delta}{2}$ fraction errors can be corrected



Binary Error-correcting codes

- C $\subseteq \{0,1\}^n$ (with Hamming metric)
- Rate R:
  - |C| = 2^{Rn}
- Distance δ:
  - $\Delta(x,y) \geq \delta n$ for distinct $x, y \in$ C
  - Implies $\delta/2$-fraction errors can be corrected
- Rate vs. Distance?
  - **OPEN**



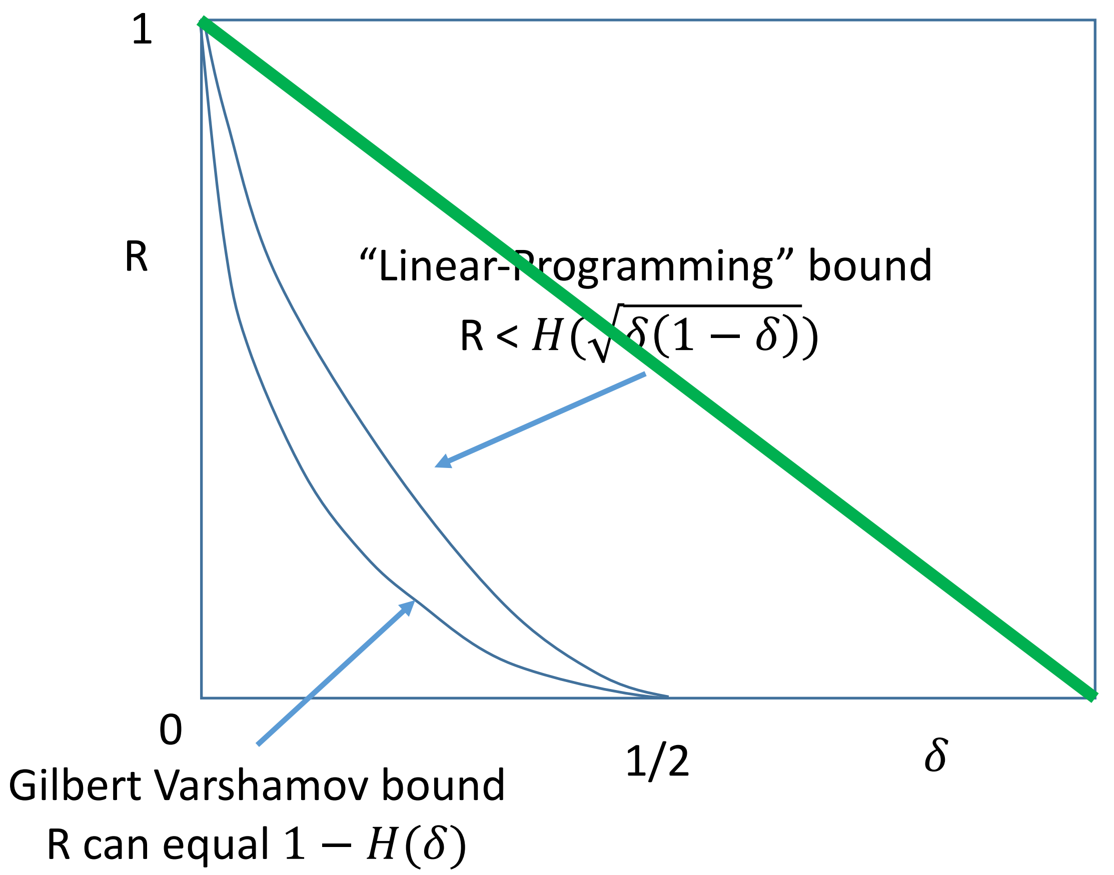
Gilbert Varshamov bound

- GV Bound: There exist codes with $R \geq 1 - H(\delta)$

Over large alphabets

R = 1 - δ is the optimal tradeoff
(a.k.a. SINGLETON BOUND)
Achieved explicitly

- Great open questions:
  - Is the GV bound tight?
  - Do there exist explicit codes meeting the GV bound?



Goals of classical coding theory

- Basic algorithmic tasks:
  - Encoding
  - Testing (error detection)
  - Decoding (error correction)

- Today we know codes with:
  - good rate-distance tradeoff
  - efficient encoding, testing, decoding
  - Linear/near-linear time

Local Codes

- Meanwhile, in early 90s complexity theory:
    - answers to questions that had never been asked

- Can we work with codes in sublinear time?

- In particular, what can we do with sublinear # queries?

Algorithmic Tasks associated with Error Correction

- **Error Detection:** Given $\mathrm{r} \in \Sigma^n$, determine if $r \in C$
  - Given $\mathrm{r} \in \Sigma^n$, with sublinear queries to r, distinguish between $r \in C$ and $\Delta(r, C) > \epsilon n$

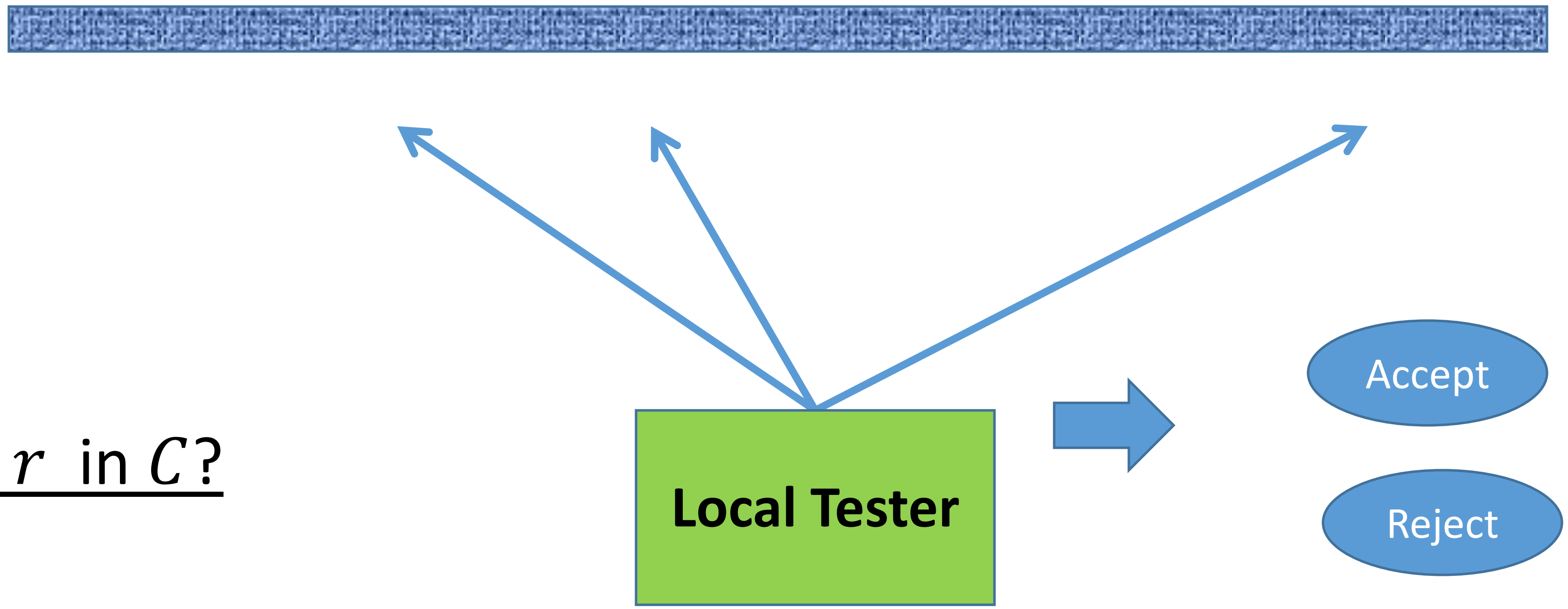
- **Error Correction:** Given $\mathrm{r} \in \Sigma^n$, if $\exists\ m$ such that

  $\Delta(r, E(m)) < \epsilon n$, find m
  - Given $\mathrm{r} \in \Sigma^n$ and $\mathrm{i} \in [k]$ if $\exists\ m$ such that $\Delta(r, E(m)) < \epsilon n$, with sublinear queries to r find m_i

Locally Testable Code

Given: $r \in \Sigma^n$

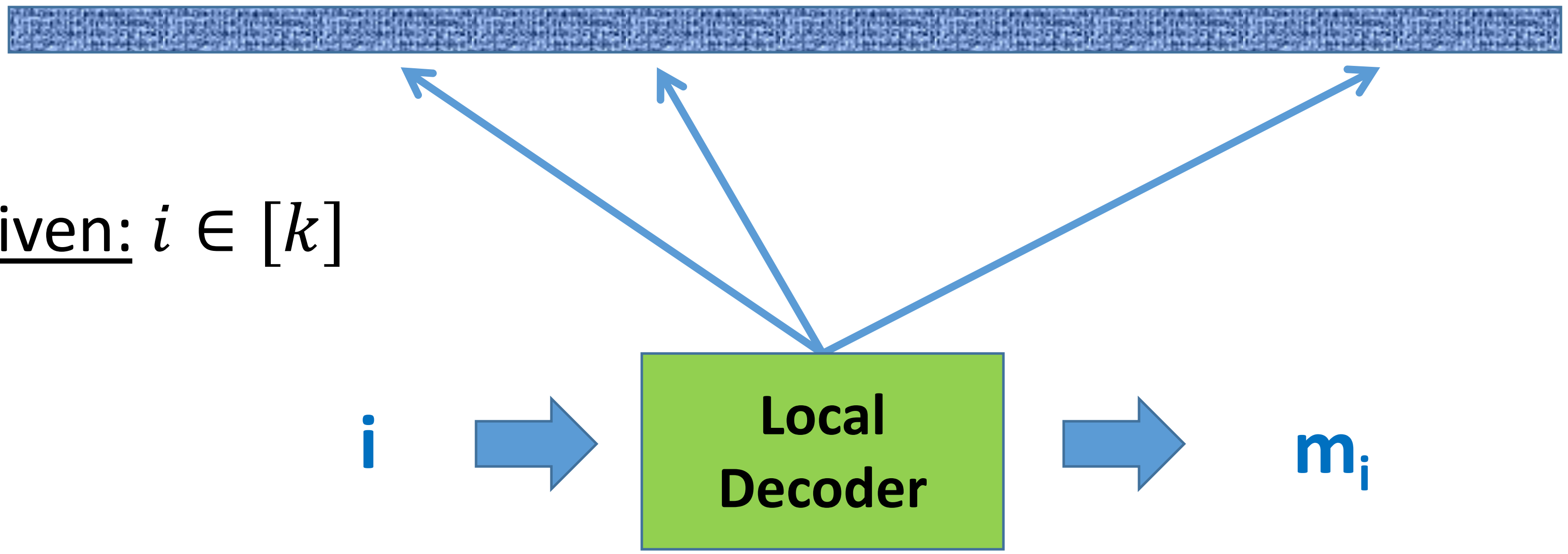
Is r in C?



Locally Decodable Codes

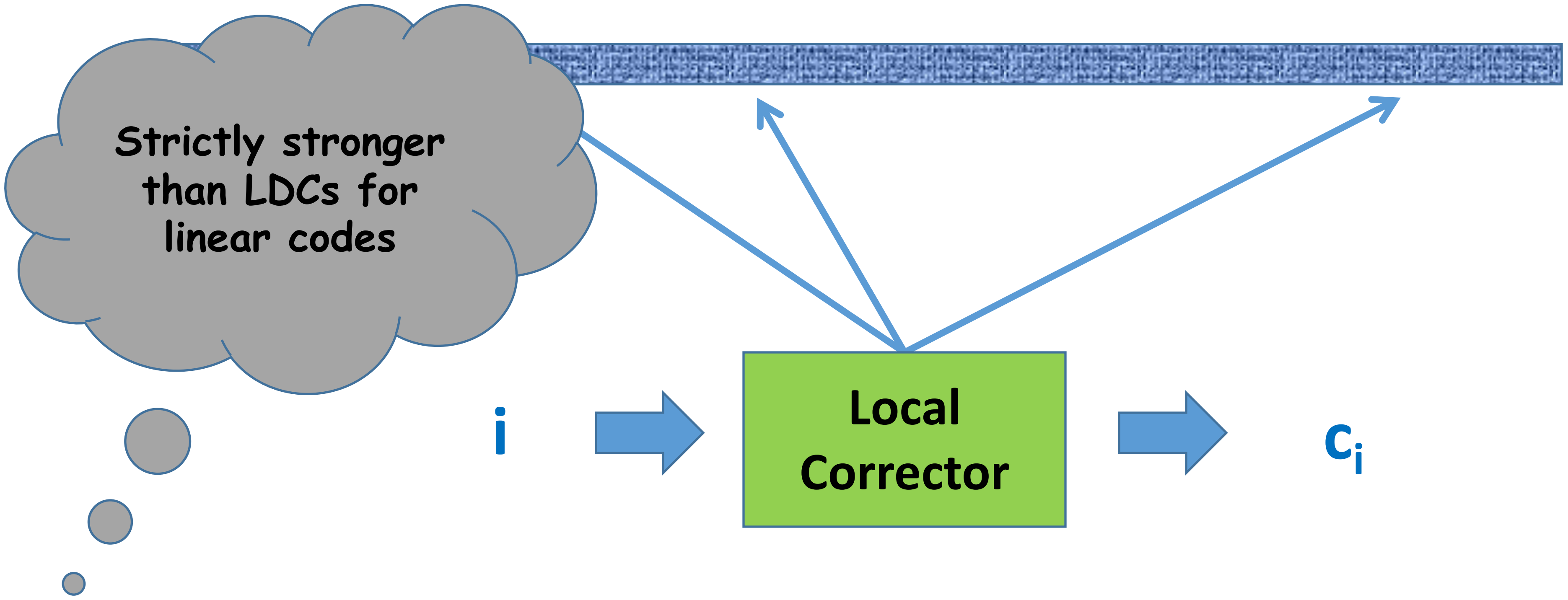
Given: $r \in \Sigma^n$ such that $\Delta(r, C) < \epsilon n$

Given: $i \in [k]$



Locally Correctable Codes

<u>Given:</u> $r \in \Sigma^n$ such that $\Delta(r, C) < \epsilon n$



Motivation for Local Decoding/Local Correcting

Many applications to cryptography and complexity theory

- Worst case to Average Case reductions
- Constructions of PRGs from One-Way functions
- Connections to Polynomial Identity Testing, Matrix Rigidity, Circuit Lower bounds
- Private information retrieval
- Learning theory

- Mathematically very interesting

- Interesting for coding theory in practice?

Motivation for Local Testing

- Implicit connections to the PCP theorem
  - Advances have led to improved PCPs
  - Limitations should lead to an understanding of limitations of PCPs

- Applications to Unique Games conjecture and hardness of approximation

- Many relations to testing of functions
  - Original [Blum-Luby-Rubinfeld] linearity tester $\approx$ testability of the Hadamard Code which led to the proof checking revolution

A nice local code

- Reed-Muller codes (multivariate polynomial evaluation codes)
  - constant rate, constant distance
  - O(n^{ϵ}) query locally testable
  - O(n^{ϵ}) query locally decodable

- Large finite field $\mathbf{F_q}$ of size q
- Interpret original data as a polynomial P(X,Y)
  - degree(P) = d = 0.1 q

- Encoding:
  - Evaluate P at each point of $\mathbf{F_q}^2$

- Rate = $\Omega(1)$
- Distance = 0.9
  - Two low degree polynomials cannot agree on many points of $\mathbf{F_q}^2$

$\mathbf{F_q}^2$

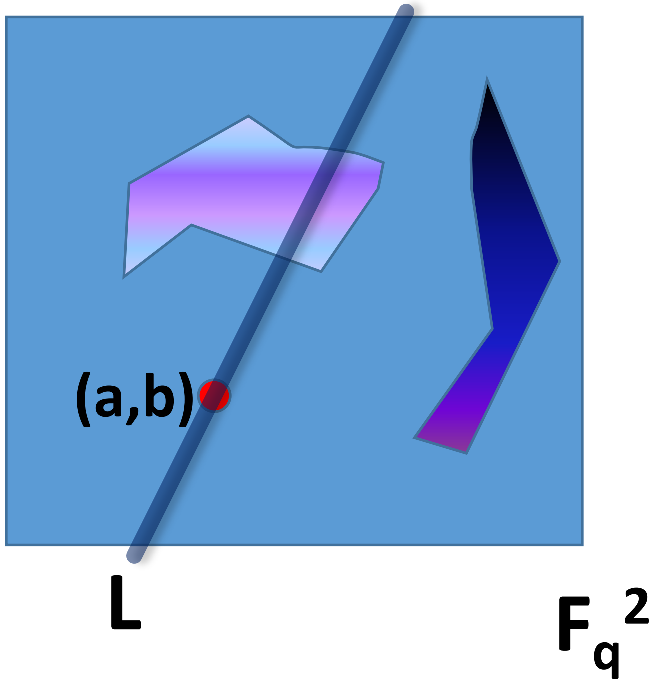
Local testing/correcting RM codes

- <u>Main idea:</u>
  - Restricting a low-degree multivariate polynomial to a line gives a low-degree univariate polynomial

- Local testing:
  - Check that restriction to a random line is a low-degree univariate polynomial
  - Analysis highly nontrivial [Rubinfeld-Sudan + others]

- Local correcting:
  - To recover P(a,b):
    - Pick random line L through (a,b)
    - Fit univariate polynomial through $r|_L$
    - Use it to recover value at (a,b)

- Query complexity
  - # points on a line = q = $O(\sqrt{n})$



Local codes of constant rate

- Reed-Muller codes (multivariate polynomial evaluation codes)
  - constant rate, constant distance
  - O(n^{ϵ}) query locally testable
  - O(n^{ϵ}) query locally decodable

- Since the 2010s, several improved codes:
  - Local testing:
    - tensor codes [BS, V], lifted codes [GKS]
  - Local decoding:
    - multiplicity codes [KSY], lifted codes [GKS], expander codes [HOW]

  - rate $\rightarrow 1$, better rate vs. distance vs. queries

Plan of talk

- Survey of some known results
- [Kopparty-Meir-RonZewi-*S* `16]
  - High rate LTCs/LCCs with improved query complexity
- [Gopi-Kopparty-Oliveira-RonZewi-*S* `17]
  - LTCs and LCCs approaching* Gilbert-Varshamov bound
- [Kopparty-RonZewi-*S*-Wootters `18]
  - Capacity achieving locally list decodable codes
- Some proofs

Locally decodable/correctab
Two regimes

Extensively studied
Many deep and amazing results (upper and lower bounds)
Many basic problems unanswered

- **Low query regime:**
  - Number of queries is small (2, 3, constant)
  - What is the best rate?

  - Theoretically very interesting
    - applications to *Cryptography, average-case complexity*
  - Too inefficient for codes in practice

- **High rate regime**
  - Let the rate be high (constant rate or rate ≈ 1)
  - What is the best query complexity that can be achieved?

  - Focus of more recent work.
  - Relevant regime for data storage and retrieval.
  - Even mild lower bounds would have very interesting consequences to rigidity, lower bounds [Dvir]

Low Query Regime (LCCs, LDCs)

- $\ell = 2$: Hadamard Code is best possible $n = 2^{\Omega(k)}$ [Goldreich-Karloff
- $\ell = 3$: $n = 2^{\sqrt{k}}$ (till not very long ago …)
- For any **constant** ℓ: Reed Muller code best known construction: $n =$ ong ago)

Matching Vector Codes:
LDCs with n = exp(exp(o(log k))
[Yekhanin, Efremenko, Dvir-Gopalan-Yekhanin]

Open question:

Can one get LDCs/LCCs with $O(1)$ queries and polynomial rate?

- **Lower bounds:**
  - $\ell = 3$: $n = \Omega(k^2)$ [Woodruff]
    - [Dvir-S-Wigderson] Over Real numbers, if code is linear then for LCCs $n = \Omega(k^{2+\epsilon})$
  - **General** ℓ: $n \geq k^{1+\frac{1}{\ell}}$ (too inefficient for codes in practice)

High rate regime (LCCs, LDCs)

- Till about 8 years ago:
  - Reed-Muller codes were the only example
  - To get query complexity $\ell = k^{\epsilon}$, Rate **R =** $\mathbf{exp}\left(\frac{1}{\epsilon}\right)$
- More recently:
  - [Kopparty-S-Yekhanin `11] Multiplicity Codes
  - [Guo-Kopparty-Sudan `13] Lifted Codes
  - [Hemenway-Ostrovsky-Wootters`13] Expander based c
  - Query complexity $\ell = k^{\epsilon}$, Rate **R=** $\mathbf{1} - \boldsymbol{\epsilon}$

  (locally decodable and correctable from a constant fract
- [Katz-Trevisan]:
  - Constant rate ⇒ must have query complexity $\Omega(\log n)$

Interesting question:

What is the best rate/query complexity tradeoff?

Can one get LDCs/LCCs with rate $\Omega(1)$ or $1 - \epsilon$ and with query complexity $k^{o(1)}$

Somewhat recent result:

[Kopparty-Meir-RonZewi-S `16]**:** There exists a family of codes of rate $1 - \epsilon$ that is locally decodable and locally correctable with $n^{o(1)}$ queries from a constant fraction of errors.

$\mathbf{2}^{\sqrt{(\log n \log \log n)}}$

What we know about constant rate LTCs

- As far as we know,
  - there could be 3-query LTCs of constant rate
- RM codes achieve:
  - For all R < $1/\exp(\frac{1}{\beta})$
  - Query complexity = $O(n^{\beta})$
- Recent progress beyond Reed-Muller codes:
  - For all R < 1
  - For all $\beta > 0$
  - Query complexity = $O(n^{\beta})$
  - Two familes of codes achieving this!
    - Tensor codes [BenSasson-Sudan], [Viderman]
    - Lifted Reed-Solomon codes [Guo-Kopparty-Sudan]

Constructions known with 3-queries and Rate = $\frac{1}{poly(\log n)}$ [BenSasson-Sudan`05, Dinur`06]

More recently:

[Kopparty-Meir-RonZewi-S `16]**:** There exists a family of codes of rate $1-\epsilon$ that are locally testable with $n^{o(1)}$ query complexity.

$\boldsymbol{(\log n)^{O(\log\log n)}}$

<u>KMRS Theorem for LCCs:</u> There exists a family of codes of rate $1-\epsilon$ that is locally decodable and locally correctable with $\mathbf{2}^{\sqrt{(\log n \log\log\ n\ )}}$ queries from a constant fraction of errors

<u>KMRS Theorem for LTCs:</u> There exists a family of codes of rate $1-\epsilon$ that is locally testable with $(\boldsymbol{log\, n})^{\boldsymbol{O(loglog\, n)}}$ queries from a constant fraction of errors.

LTCs and LCCs approaching the GV bound

- Theorem [Gopi-Kopparty-Oliveira-RonZewi-*S* `17]

(informal) We can construct LTCs and LCCs which achieve the best possible rate-distance tradeoff that we know how to achieve with general (nonlocal) codes.

Main Result: LTCs

[Gopi-Kopparty-Oliveira-RonZewi-S `17]

Theorem:

For all R, δ with:

$$R < 1 - H(\delta)$$

there exists an infinite family of codes C_n

such that:

- length(C_n) = n
- Rate $\geq$ R
- Distance $\geq$ δ
- C_n is locally testable with $(\log n)^{O(\log\log n)}$ queries

Local codes can be list decoded up to capacity

[Hemenway-RonZewi-Wootters`17, Kopparty-RonZewi-*S*-Wootters`18]

There exist codes that can be *locally list decoded* up to capacity

with query complexity $2^{(\log n)^{\frac{3}{4}}}$

[KMRS] result (and proof ideas) – an important ingredient in all these results.

Rest of talk – sketch of proof of KMRS result for LCCs

KMRS Theorem for LCCs: There exists a family of codes of rate $1 - \epsilon$ that is locally decodable and locally correctable with $2^{\sqrt{(\log n \log\log n)}}$ queries from a constant fraction of errors

KMRS Theorem for LTCs: There exists a family of codes of rate $1 - \epsilon$ that is locally testable with $(\boldsymbol{log\ n})^{\boldsymbol{O(loglog\ n)}}$ queries from a constant fraction of errors.

Proof of KMRS result: *2 components*

- **Component 1:** High rate codes with *sub-polynomial query complexity* but only tolerating a tiny *sub-constant fraction of errors*

- **Component 2:** "Distance Amplification"
  - Takes code as above and transforms it to a code that can tolerate many more errors

Component 1

- High rate codes with *sub-polynomial query complexity* but only tolerating a tiny *sub-constant fraction of errors*

Can be achieved by Multiplicity Codes!

(In a regime of parameters not studied before)

Multiplicity Codes
[Kopparty-S-Yekhanin`11]

Theorem (original)

For every $\epsilon > 0$,
for inf. many k, there are codes encoding

k bits -> $(1+\epsilon)$ k bits (symbols)
decodable in $O(k^{\epsilon})$ time (+queries)
from $\delta(\epsilon) > 0$ fraction errors.

Theorem (sub-constant distance)

For every $\epsilon > 0$
for inf. many k, there are codes encoding

k bits -> $(1+\epsilon)$ k bits (symbols)
decodable in $O(2^{\sqrt{\log k \log\log k}})$ time (+queries)
from $\approx \sqrt{(\log\log k)/\log k}$ fraction errors.

Construction of Mult. Codes

- Reed Muller Codes
- Augment it with “derivatives”

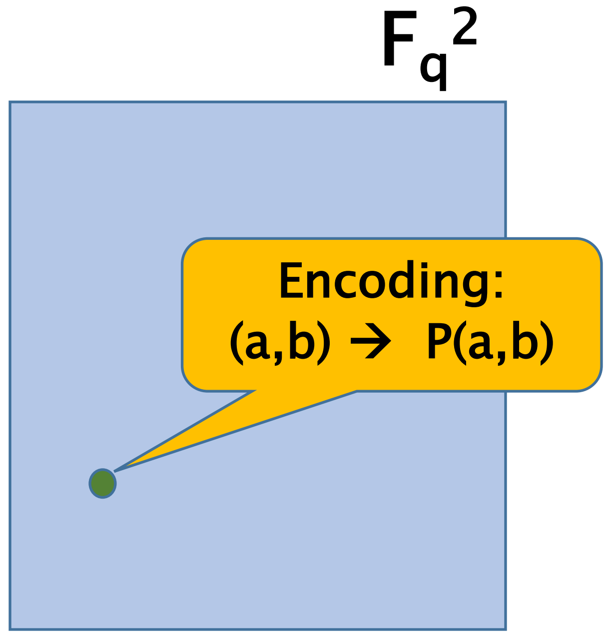
Reed-Muller Codes

Bivariate **Reed-Muller**

- Large finite field of size q

- Interpret original data as a polynomial P(X,Y)
  - degree(P) $\cdot$ d = (1- δ) q

- *Encoding:* Enc(P)
  - At each point (a,b) $\in F_q^2$,
    Evaluate P(a,b)



Key observations

- Schwartz-Zippel Lemma
  - 2 polynomials of degree < (1 - δ) q   differ on at least δ fraction of points
- So:
  - Any two codewords are at least δn apart

Decoding Reed-Muller Codes

- **<u>Given:</u>**
  - noisy encoding of P(X,Y)
    - Deg(P) = q $(1 - \delta)$
  - point (a,b) in $\mathbf{F}_q^2$

- **<u>Goal:</u>** recover P(a,b)

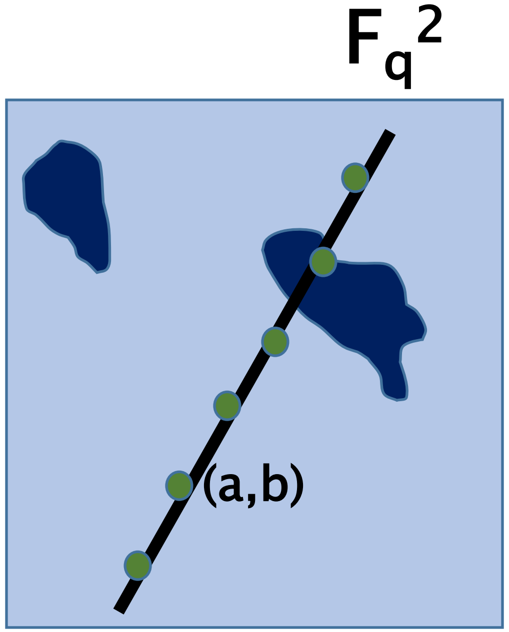
<u>Algorithm</u>

- Take random line L through (a,b)

- Query points on L
  - Should have small error
  - Noisy encoding of $P|_L$ (univariate polynomial)

- Recover $P|_L$
  - "Reed Solomon" decoding

- Compute $P|_L$ (a,b)
  = P(a,b)



Parameters of Reed-Muller Codes

- Bivariate Reed Muller:
  - k = (d+2) choose 2 $\approx \frac{(1-\delta)^2 q^2}{2}$
  - $n = q^2$
  - **Rate** $\approx \frac{1}{2} - \delta$
  - **# Queries:** $\ell \approx$ **$O(k^{1/2})$**

- Improve query complexity → increase # of variables

More variables

- Polynomials of deg · (1-δ) q in ***m variables***
- k = (d+m) choose m $\approx \frac{(1-\delta)^m q^m}{m!}$
- n = q^m
- **Rate** $\approx \boldsymbol{\frac{(1-\delta)^m}{m!}}$
- **Queries** = q $\approx n^{1/m} \approx$ **O($k^{1/m}$)**

- Decodable from $\Omega(\delta)$ errors

- Bottleneck for rate: Degree needs to be small

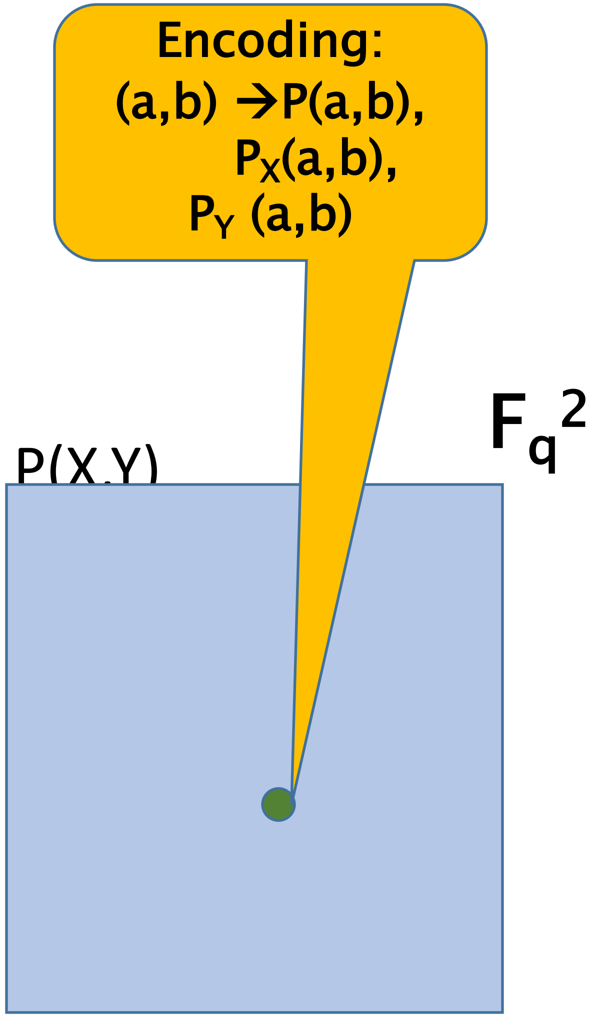
Multiplicity Codes

- Key idea: Derivatives
- Higher degree polynomials
  - (too high for Reed-Muller)

Multiplicity Codes

***Bivariate* Multiplicity codes**

- Large finite field of size q
- Interpret original data as a (high) degree polynomial P(X,Y)
  - degree(P): d = $\mathbf{2} \times (1-\delta)$ q
- *Encoding:* Enc(P)
  - At each point (a,b) $\in F_q^2$, evaluate:
    - $\langle P(a,b), P_X(a,b), P_Y(a,b) \rangle$



Schwartz–Zippel with Multiplicities [Dvir–Kopparty–*S*–Sudan'10]

- 2 polynomials of degree < 2q $(1-\delta)$ cannot agree on their evaluations and evaluations of derivatives in more than $(1-\delta)$ fraction points

- # roots of P *counted with multiplicity* · deg(P) $|F|^{n-1}$

- Multiplicity Codes have good distance

Decoding Multiplicity Codes

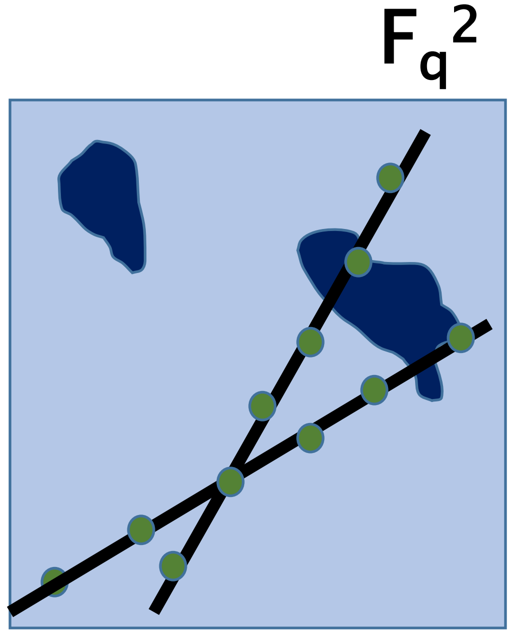
Given:

- noisy encoding of <P, P_X, P_Y>
  - Deg(P) = $2 \times q\,(1-\delta)$
- point (a,b) in $\mathbf{F}_q^2$

Goal: recover <P(a,b), P_X(a,b), P_y(a,b)>

Algorithm

- Take random line L through (a,b)
  - Should have small error
- Query points on L
  - P_X, P_Y give directional derivative of P along L
  - Noisy encoding of $P|_L$ (univariate polynomial), and of der($P|_L$)
- Recover $P|_L$
- Repeat above steps
- We thus know P(a,b), der($P|_{L1}$) (a,b), der($P|_{L2}$) (a,b)
- This gives us P(a,b), P_X(a,b), P_Y(a,b)



Parameters of Multiplicity Codes

- Bivariate Multiplicity Codes of order 2:
  - $k = (d+2)$ choose $2 / 3 \approx (2(1-\delta)q)^2 / 6$
  - $n = q^2$
  - **Rate $\approx$ 2/3 - δ**
  - **# Queries: $\approx$ O($k^{1/2}$)**

- Improve Rate → increase order of derivatives

- Improve query complexity → increase # variables

More variables, many derivatives

- **m – variate, derivatives up to order s**
- Polynomials of **degree (1-δ)sq**
  - **Query Complexity: $\approx k^{1/m}$**
- **Rate $\approx$ (s/ m+s)^m $\times$ (1-δ)^m**
  - so if s >> m,   rate →1
- Decoding as before …
  - (+ some "robustification")

Reed-Muller Codes

- Messages: Low degree polynomials

- Encoding: Evaluation of polynomial on full domain

- #queries: Decreases with increase in # variables

- Rate: Decreases exponentially with increase in #vars

Multiplicity Codes

- Messages: High degree polynomials

- Encoding: Evaluation of polynomial *and its derivatives* on full domain

- #queries: Decreases with increase in # variables

- Rate: 1

Multiplicity codes in low distance regime

To make *queries sub-polynomial*, choose m to be super-constant. For *constant rate* this *forces distance to be sub-constant*.

<u>Theorem (sub-constant distance)</u>

For every $\epsilon > 0$

for inf. many k, there are codes encoding

k bits -> $(1+\epsilon)$ k bits (symbols)

decodable in $O(2^{\sqrt{\log k \log\log k}})$ time (+queries)

from $\approx \sqrt{(\log\log k)/\log k}$ fraction errors.

Component 2

- **Distance amplification**
  - Similar technique used by [Alon-Luby'96] and then by others [GI'05, GR'08]

<u>Theorem (sub-constant distance)</u>

For every $\epsilon > 0$
for inf. many k, there are codes encoding

k bits -> $(1+\epsilon)$ k bits (symbols)
decodable in $O(2^{\sqrt{\log k \log\log k}})$ time (+queries)
from $\approx \sqrt{(\log\log k)/\log k}$ fraction errors.

Component 2

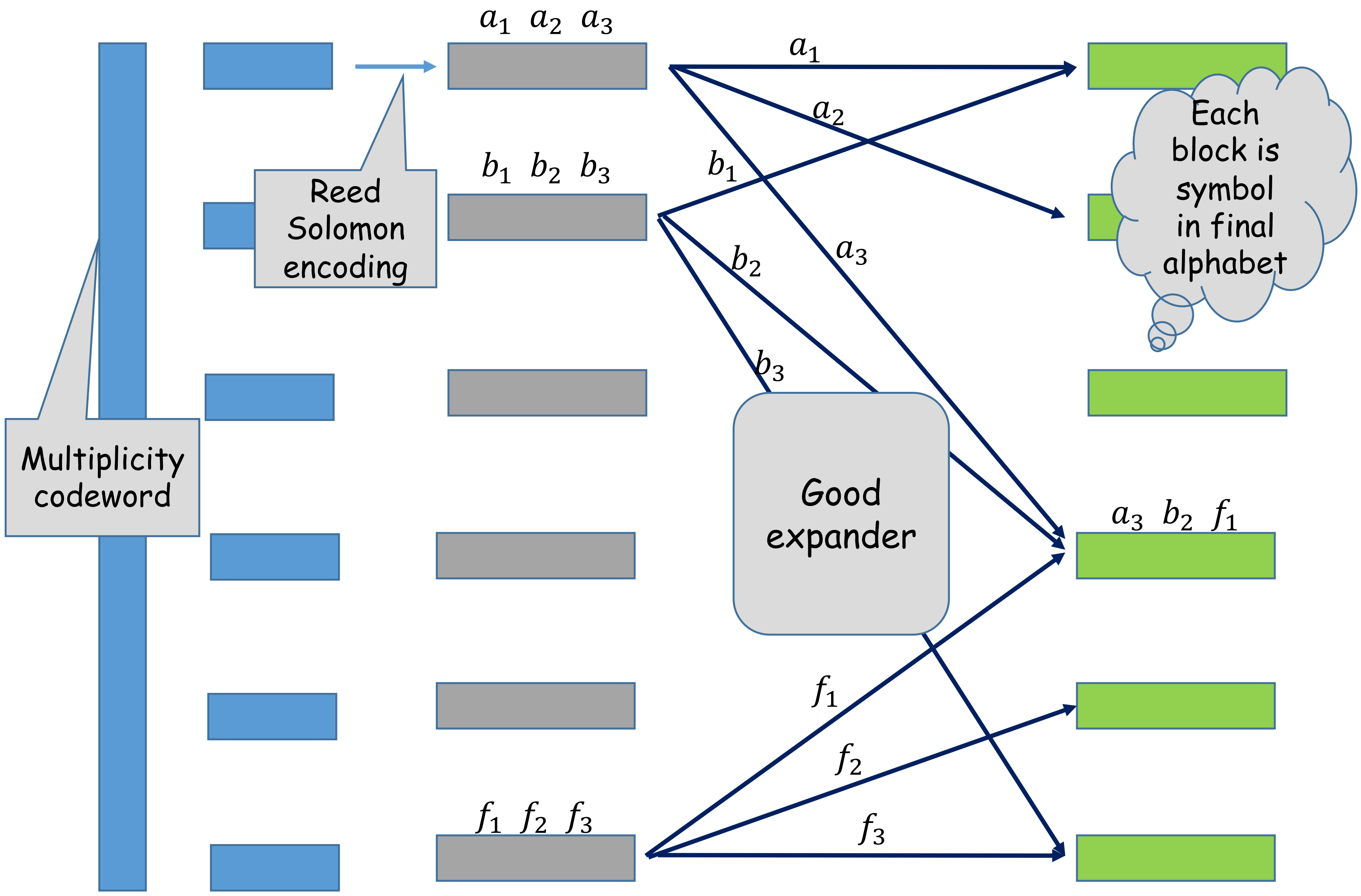
- **Distance amplification**
  - Similar technique used by [Alon-Luby'96] and then by others [GI'05, GR'08]

Theorem (~~sub-constant distance~~)

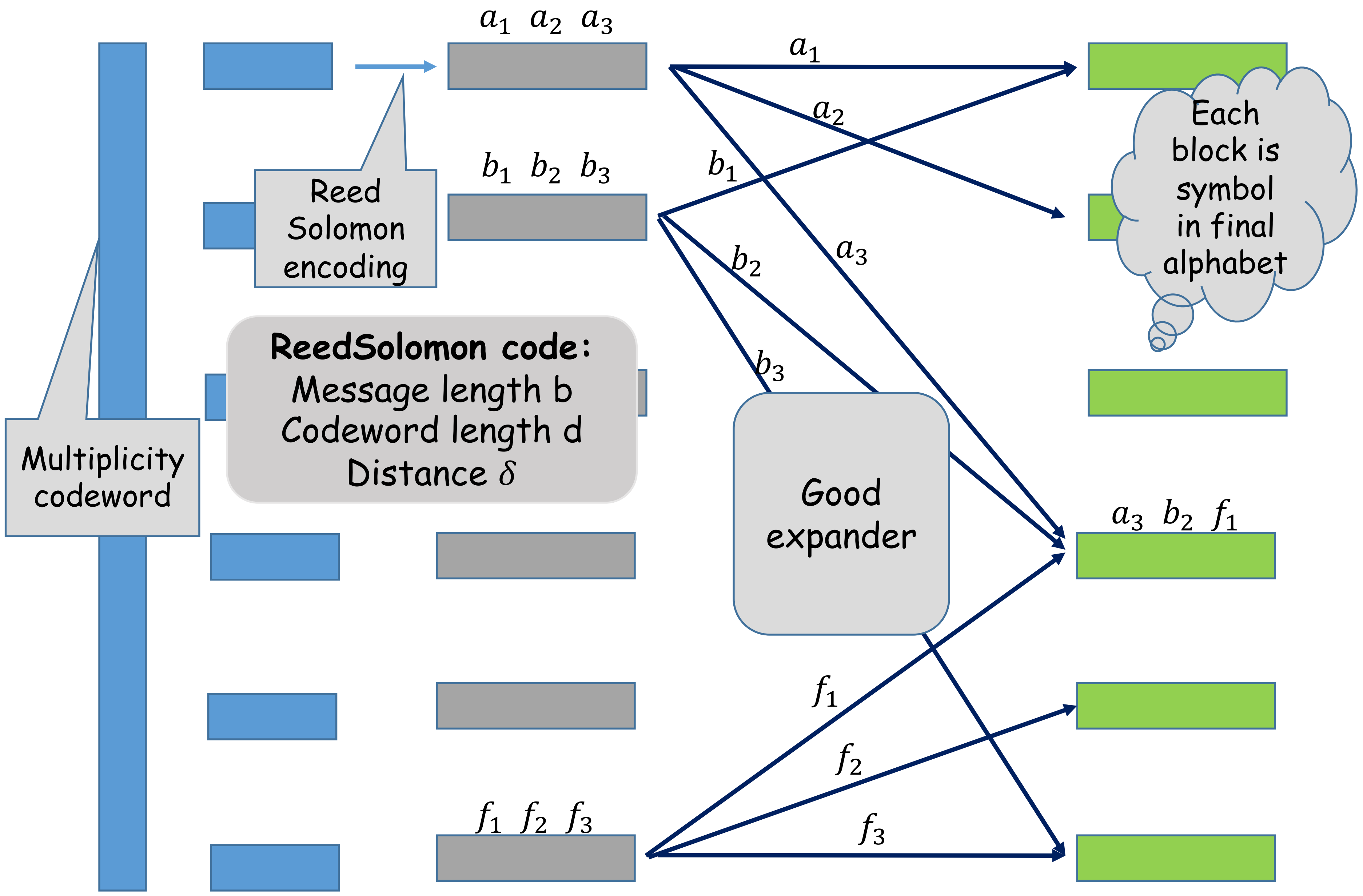
For every $\epsilon > 0$
for inf. many k, there are codes encoding

k bits -> $(1+2\epsilon)$ k bits (symbols)
decodable in $O(2^{2\sqrt{\log k \log\log k}})$ time (+queries)
from $\approx \cancel{\sqrt{(\log\log k)/\log k}}$ fraction errors.

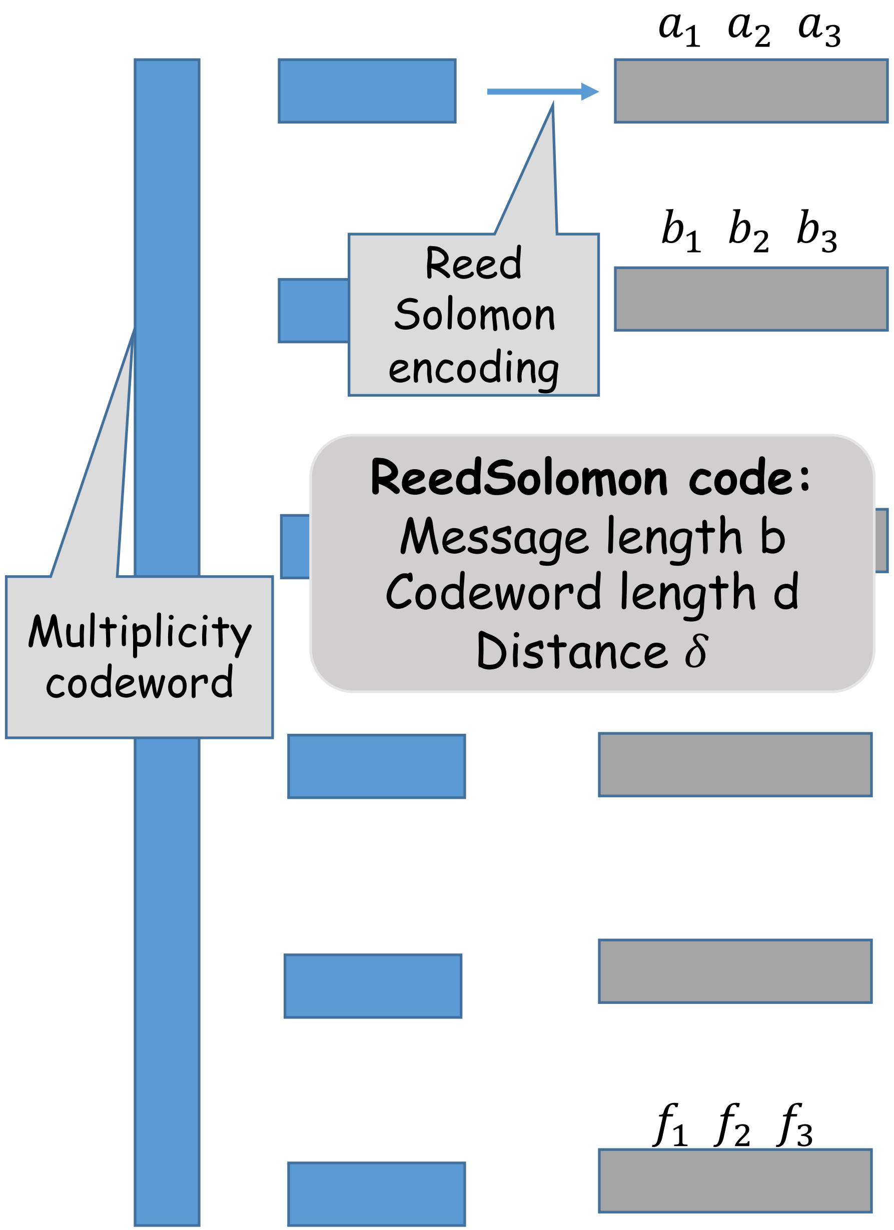
$\boldsymbol{\Omega(1)}$



Multiplicity codeword
Reed Solomon encoding
a_1 a_2 a_3
b_1 b_2 b_3
f_1 f_2 f_3
a_1
a_2
a_3
b_1
b_2
b_3
f_1
f_2
f_3
Good expander
Each block is symbol in final alphabet
a_3 b_2 f_1



a_1 a_2 a_3
b_1 b_2 b_3
f_1 f_2 f_3
Reed Solomon encoding
Multiplicity codeword
ReedSolomon code:
Message length b
Codeword length d
Distance δ
Good expander
a_1
a_2
a_3
b_1
b_2
b_3
f_1
f_2
f_3
Each block is symbol in final alphabet
a_3 b_2 f_1



Decoding from random errors:

Suppose $\frac{\delta}{2} - \epsilon$ fraction of random errors

Most (1-o(1)) grey blocks have at most $\frac{\delta}{2}$ corruptions

Those Reed-Solomon codewords can be correctly decoded

Thus 1-o(1) fraction of the blue blocks can be correctly recovered. This is low enough error for multiplicity codes to handle

Everything can be done locally

Decoding from adversarial errors:

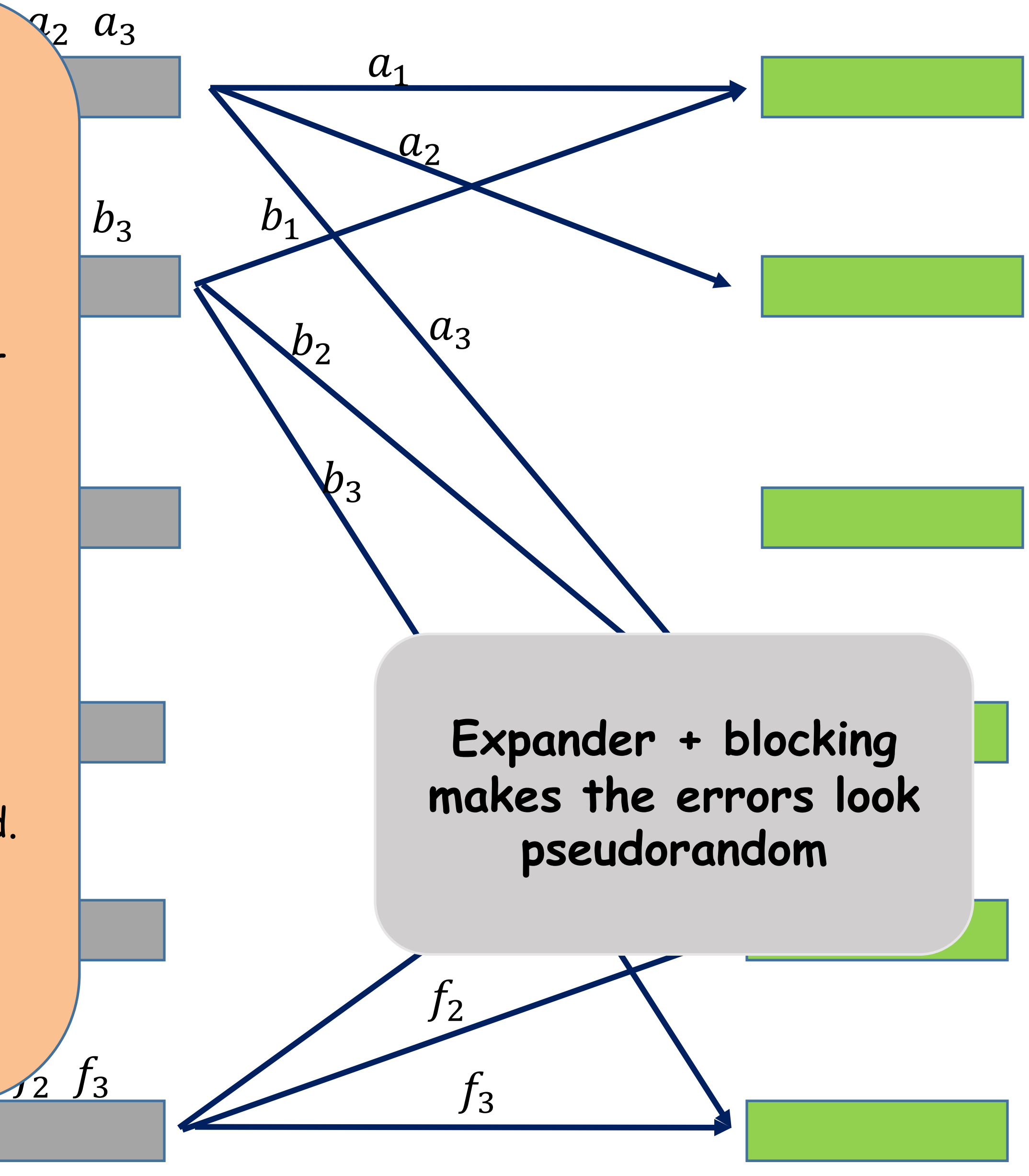
Suppose $\frac{\delta}{2} - \epsilon$ fraction of green blocks get corrupted

Most (1-o(1)) grey blocks have at most $\delta/2$ corrupt neighbors (expander mixing lemma).

Those Reed-Solomon codewords have at most $\frac{\delta}{2}$ errors and can be correctly decoded

Thus 1-o(1) fraction of the blue blocks can be correctly recovered. This is low enough error for multiplicity codes to handle

Everything can be done locally



Expander + blocking makes the errors look pseudorandom

Open questions

- Best possible query complexity for high rate LDCs and LTCs?
  - LTCS – potentially high rate 3 query LTCs!
  - LDCs/LCCs – potentially high rate log n query LCCs
- Explicit codes meeting the GV bound?
  - Almost solved by Ta-Shma!
- Is the GV bound tight?

Thanks!