Time and Space Efficient Deterministic Decoders

Joshua Cook, Dana Moshkovitz

The University of Texas at Austin

What is an Error Correcting code?

Definition:

Function C: $\Sigma^k \rightarrow \Sigma^n$ such that for $x \neq y$, we have

$$\Pr_i[C(x)_i \neq C(y)_i] \geq \boldsymbol{\delta}^*.$$

Relative distance $\boldsymbol{\delta}^*$, rate $r = k/n$. Good if $\boldsymbol{\delta}^*$, $r = \boldsymbol{\Omega}(1)$ and $|\Sigma| = O(1)$.

The correcting radius is $\boldsymbol{\delta} = \boldsymbol{\delta}^*/2$.

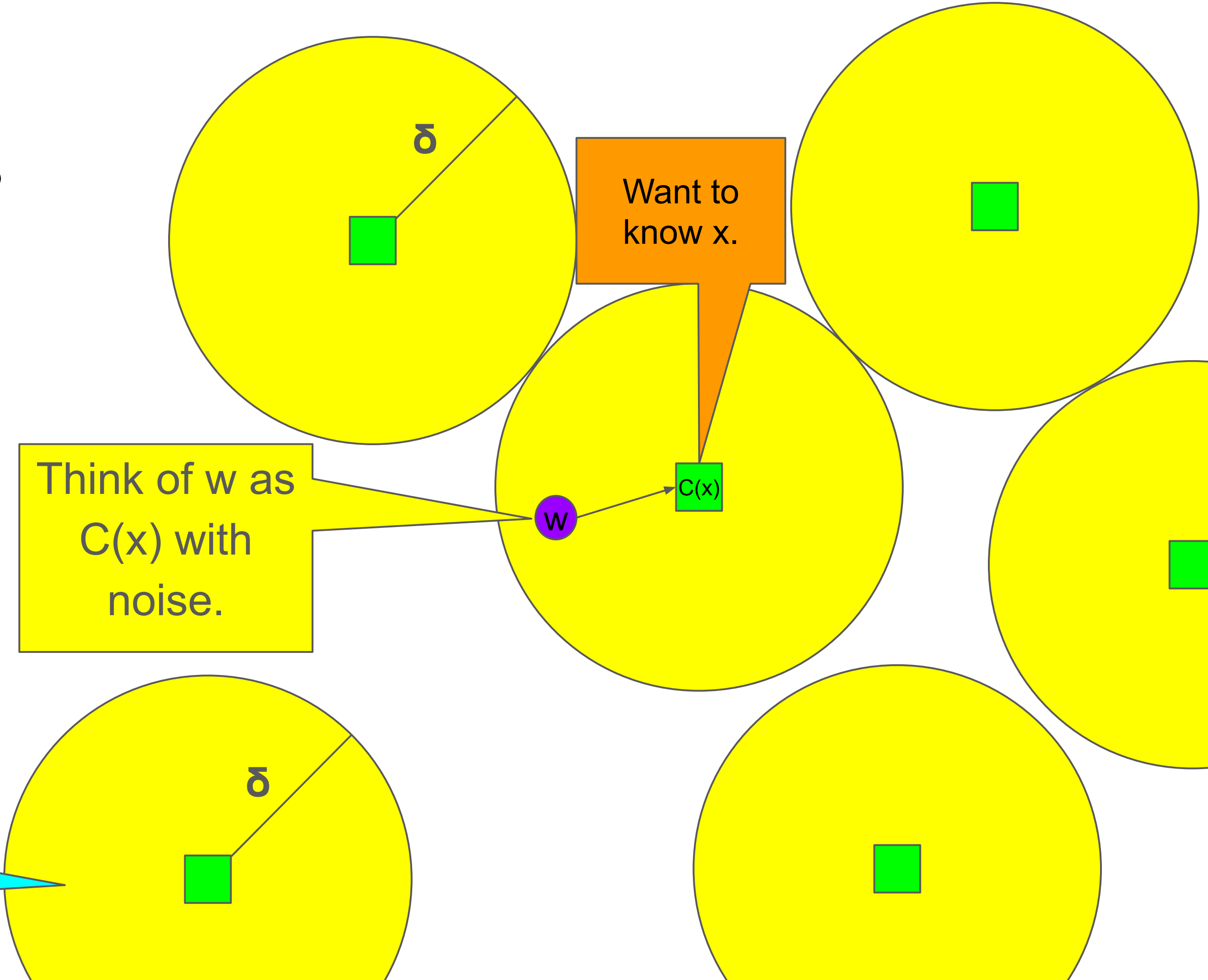
Many, many applications,

some of which are in the small space regime.

What is decoding?

Given w close to C(x), output x.

If w is within **δ** of C(x), we decode x.



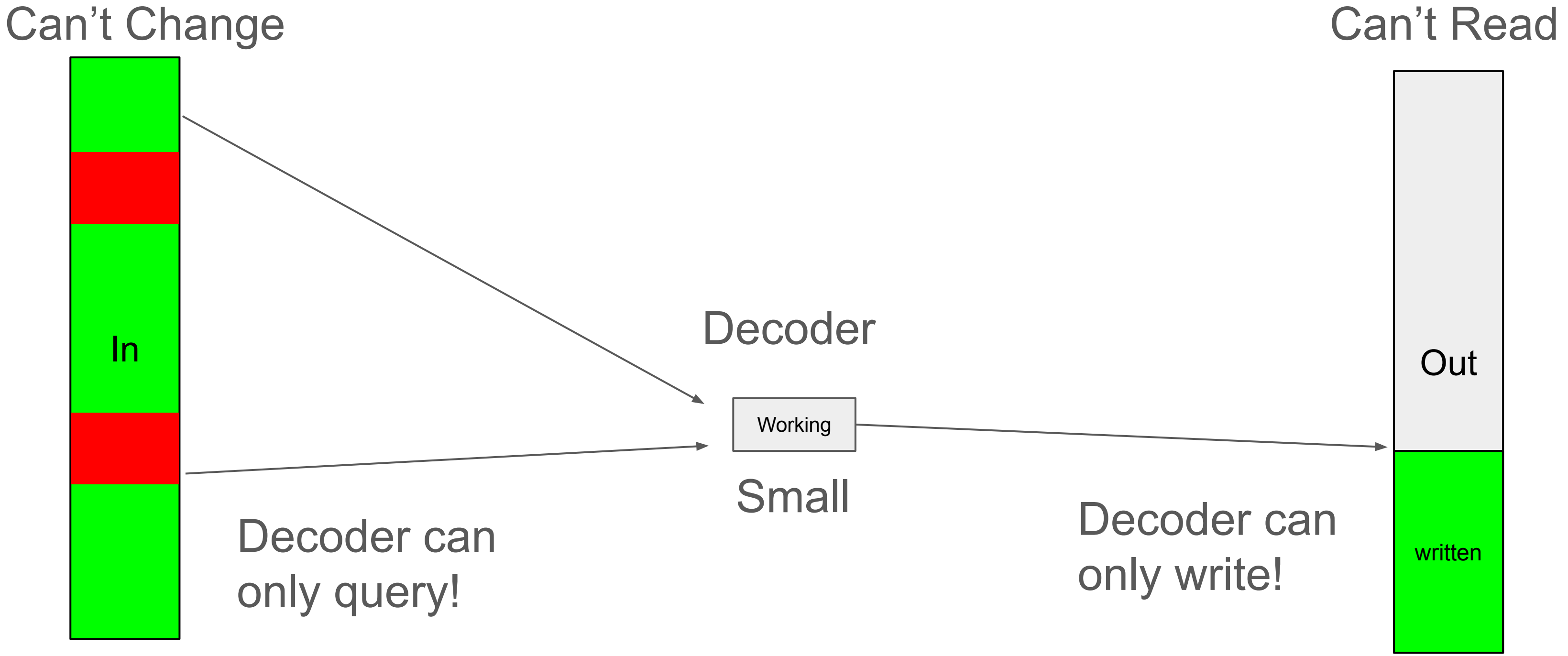
Want to know x.

Think of w as C(x) with noise.

Promised that only one codeword is in radius **δ**.

What is a low space decoder?

Input is read only. Output is write only. Have only $n^{o(1)}$ working space.



Can’t Change

Can’t Read

Decoder

Small

Decoder can only query!

Decoder can only write!

Local Decoding

Can we correct a single symbol only looking at a small number of symbols?

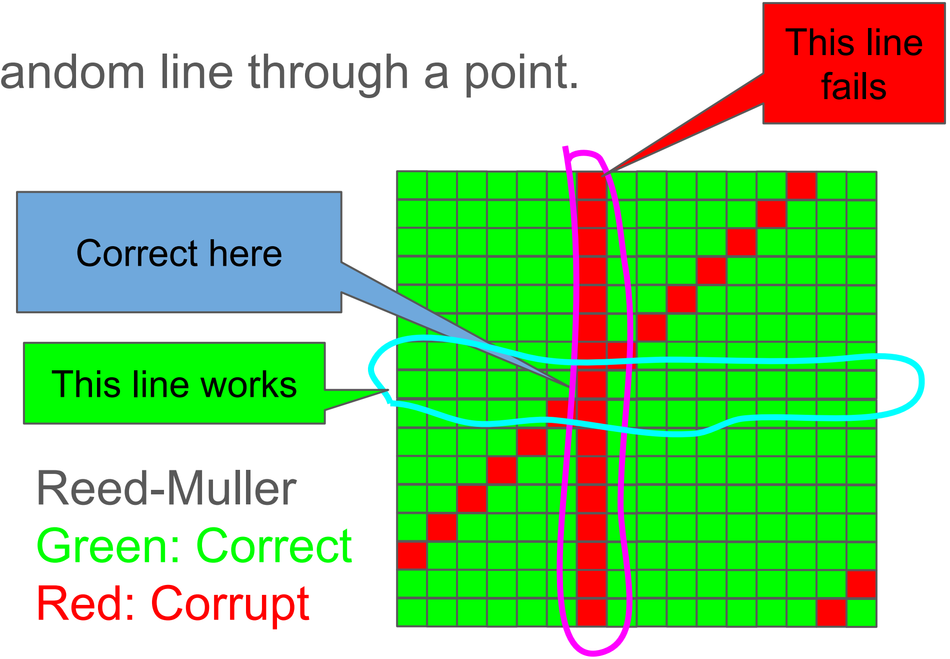
Randomly? Yes.

For Reed Muller, take a random line through a point.

If it rarely hits corruption, local correction succeeds.

Deterministically? No.

Always checks the same few symbols, adversary corrupts those.



This line fails

Correct here

This line works

Reed-Muller

Green: Correct

Red: Corrupt

Locally Correctable Codes (LCC)

Definition:

An LCC is a code $C: \Sigma^k \rightarrow \Sigma^n$ with a randomized algorithm D such that:

For any $w \in \Sigma^n$, $x \in \Sigma^k$ where $\Pr_i[C(x)_i \neq w_i] \leq \boldsymbol{\delta}'$ and any $i \in [n]$ we have

$$\Pr[D(w, i) \neq C(x)_i] \leq ⅓$$

D is q query if it only makes q queries to w. **δ**’ is called the correcting radius.

Most codes we consider are systematic (the message is contained as plain text within the codeword), so local correctors are decoders.

Randomized Decoder Continued

Reed-Muller codes are locally correctable!

By repeating O(log(n)) times, the error probability drops below 1/n.

So it is unlikely any symbol is decoded incorrectly.

This gives us a time and space efficient **randomized** decoder.

Local correction cannot be **deterministic** (and *always* correct)!

This decoder is **non-adaptive** (queries do not depend on the input).

First Result: Efficient (Non-Uniform) Deterministic Decoders

Theorem (Deterministic Decoder for LCC)**:**

Good, typical* $q = n^{\alpha}$ query LCCs have **non-uniform**, *deterministic* decoders running in space $O(q \log(n)^2)$ and time $n^{1+O(\sqrt{\alpha})}$.

*(typical means systematic, non-adaptive, and with perfect completeness).

For $q = n^{o(1)}$: space is $n^{o(1)}$ and time $n^{1+o(1)}$.

A prior result by Gronemeier proved non-adaptive decoders for good codes required time T and space S such that $ST = \Omega(n^2)$.

A flawed approach

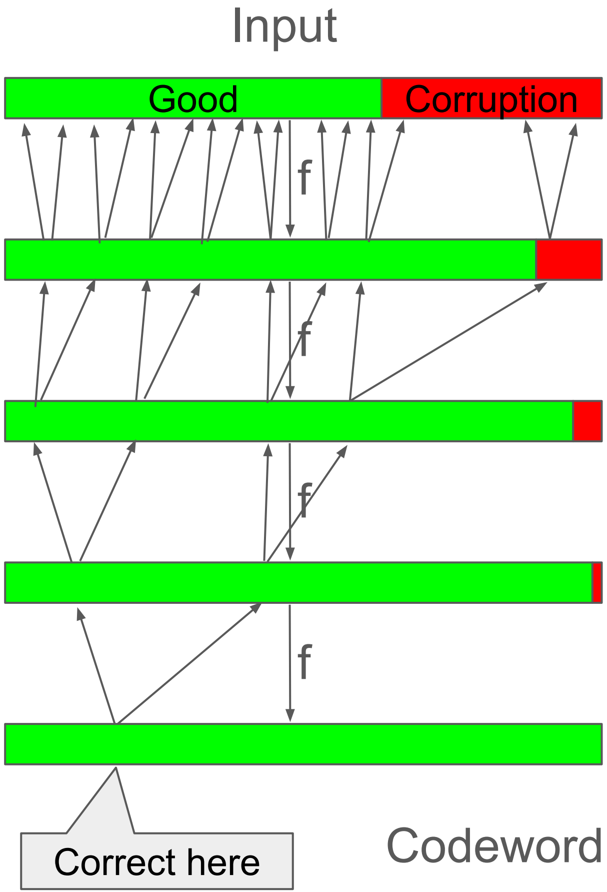
Find a (single) q query f such that

$$\Pr_{i,j}[C(x)_i \neq f_j(w)_i] \leq \eta \Pr_i[C(x)_i \neq w_i]$$

f reduces the fraction of corruptions by η.

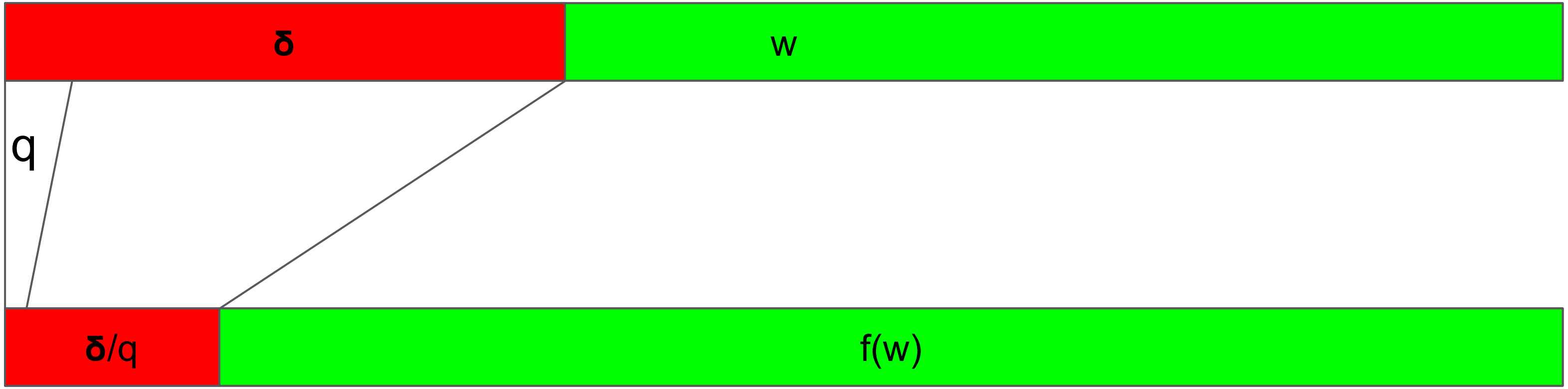
Tradeoff between q and η.

Need both q and η small.



How good can a single, deterministic f be?

If f makes q queries, and we can corrupt **δ** fraction of symbols.



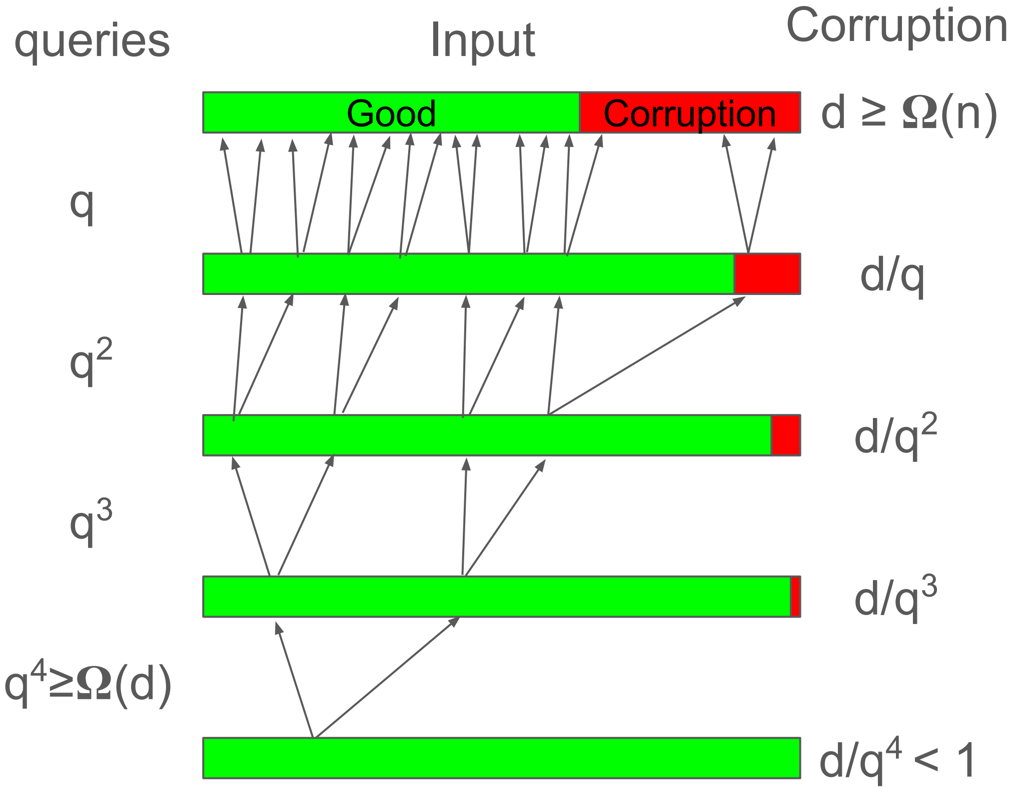
For deterministic f, only get η = 1/q

Why doesn’t it work?

Deterministic, q query f only reduces **δ** corruptions to **δ**/q.

To get zero errors, we would need **δ**n queries (per symbol)!

This gives a total time of **δ**n^2.



Fix: More than one function

What about m different q query functions $f_1, \ldots, f_m$?

Now the O(**δ**/q) failures are distributed among m functions.

Less than O(**δ** / m) on average.

Don't have to pay for m in the recursion, so can make m $\gg$ q!

Definition:

$f_1, \ldots, f_m$ is a below **δ**, factor η improving set if for any w and C(x) with $\Pr_i[C(x)_i \neq w_i] \leq$ **δ** we have

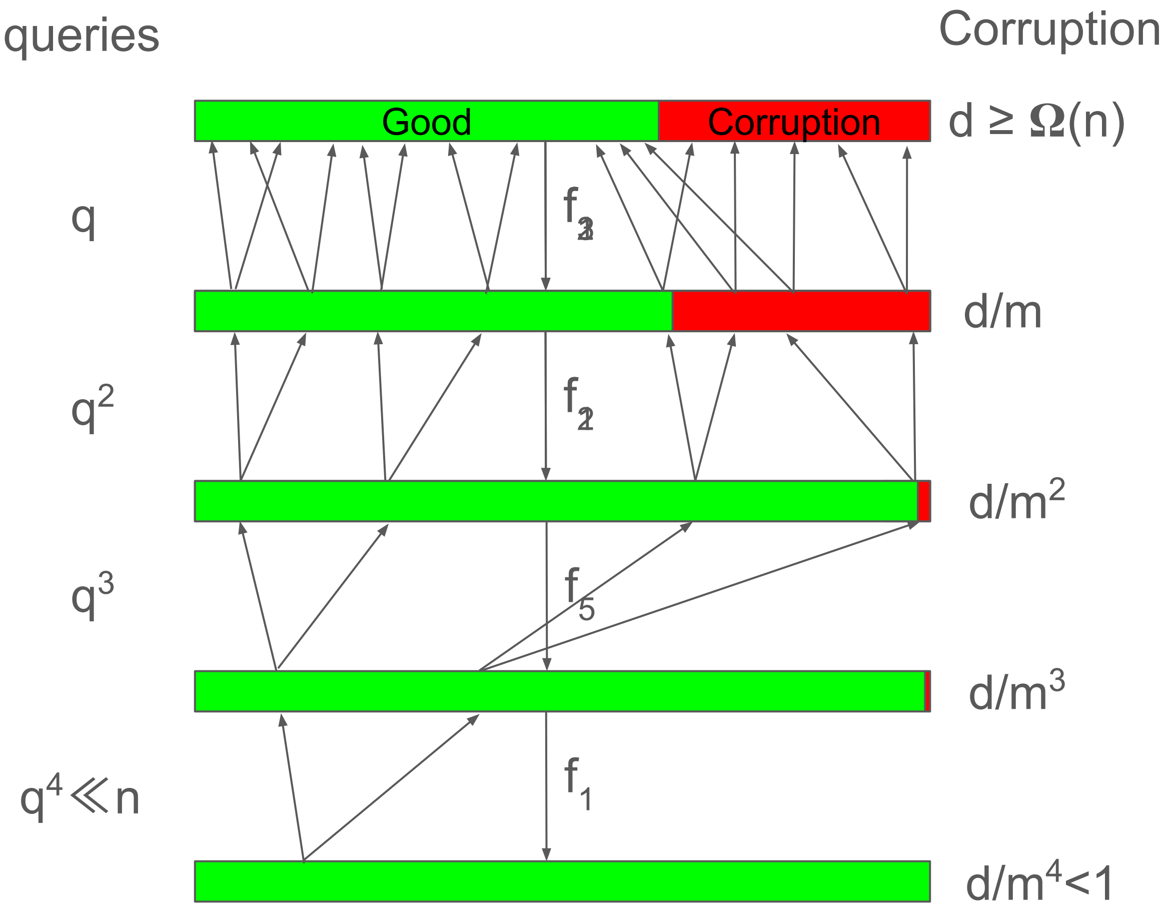
$$\Pr_{i,j}[C(x)_i \neq f_j(w)_i] \leq \eta \Pr_i[C(x)_i \neq w_i]$$

Example

Try all f_1, .., f_m.

Keep going until error is zero.

Still recursive, but fewer levels if $q \ll m$.



Can We Find such an Improving Set?

Yes, (for typical LCC).

For any q query LCC and η, there is a q query improving set with size

$$m = O(\log(n)^2/\eta).$$

If $q = n^{o(1)}$, then setting η appropriately gives space $n^{o(1)}$ time:

$$n^{1+o(1)}.$$

Uniform Decoding for Reed-Muller

Second Result: Efficient **Uniform** Decoders

Theorem (Deterministic, Uniform Decoders)**:**

There is a good code with a **uniform**, deterministic decoder running in space $n^{o(1)}$ and time $n^{1+o(1)}$.

The code is based on Lifted Reed-Solomon codes.

Also applies to the special case of Reed-Muller codes.

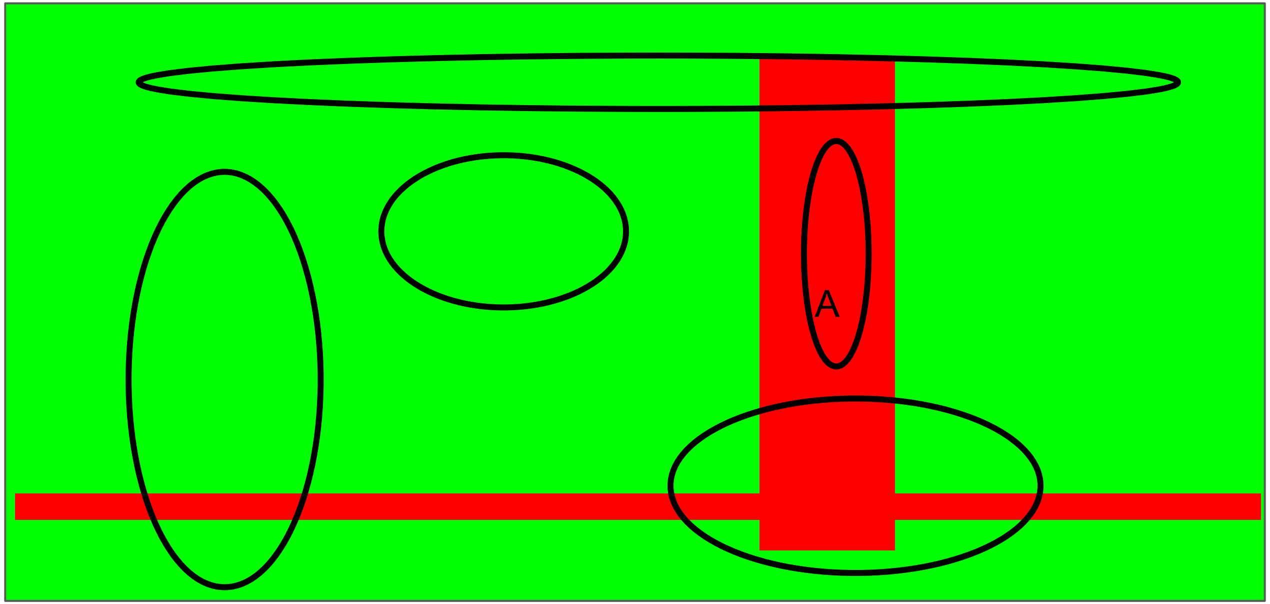
Samplers

Family of subsets of N, $\mathscr{S}$.

We say $\mathscr{S}$ is a sampler if:

For all sets A (let $\mu = {}^{|A|}/_{|N|}$).

The probability S ∈ $\mathscr{S}$ oversamples A is low.



Definition: We say $\mathscr{S} = (S_1, S_2, \ldots S_k)$ where each $S_i \subseteq N$ is a sampler for N if for some accuracy error $\varepsilon > 0$ and strong confidence error δ, for all $A \subseteq N$, and $\mu = {}^{|A|}/_{|N|}$ we have

$$\Pr_i\left[{}^{|S_i \cap A|}/_{|S_i|} \geq \mu + \varepsilon\right] \leq \delta\mu.$$

Curve Samplers

Need samplers with special structure to allow decoding.

- Lines (Line samplers)
- Subspaces (Space Samplers).
- Curves (Curve Samplers).

Prior curve samplers by Ta-Shma and Umans (and later by Guo) exist, but they:

- Had too many samplers, more than n^4, while we need $n^{1+o(1)}$.
- Only proved a weaker notion of confidence error.

Third Result: New Curve Samplers

Theorem (Curve Sampler)**:**

For appropriate degree t, dimension d, and any $\varepsilon > 0$ there is a degree t-curve sampler for $\mathbb{F}^d$ of size n $|\mathbb{F}|^{\text{poly}(t)}$ with accuracy error ε and strong confidence error:

$$\delta = O_t\left({}^{1}/(\varepsilon|\mathbb{F}|)^{\square(t)}\right)$$

The number of samples is close to $n = |\mathbb{F}|^d$ when $t \ll d$.

The number of queries is $q = |\mathbb{F}|$.

For large t, confidence error is less than 1/q.

Open Problems

1. Find a single code that is encodable and decodable in space $n^{o(1)}$, time $n^{1+o(1)}$.

~~2. Extend to list decoding.~~

   a. Already did this in a follow up work *for Reed-Muller* codes.

      i. Constants are huge, 10^{20}. Give better constants.

   b. Doesn't have constant rate when achieving space $n^{o(1)}$, time $n^{1+o(1)}$.

3. Give time/space efficient uniform decoders for more codes (like multiplicity codes).

   a. Our uniform decoders are only for lifted Reed-Solomon.

   b. Our technique for multiplicity codes only gives non-uniform decoders.

4. Find a deterministic decoder running in space polylog(n) and time npolylog(n).

Any Questions?

Extra Technical Details

Getting Better Rate with Fewer Queries.

Reed-Muller gives bad rate for few queries.

Use a closely related code called Lifted Reed-Solomon.

  Low degree only when restricted to lines.

  For high degree and low characteristic, more general than Reed-Muller.

Lifted Reed-Solomon with high rate (and few queries) has low distance.

Use similar distance amplification technique as Kopparty, Saraf, and Yekhanin.

Why not Multiplicity codes?

Multiplicity codes need pseudorandom lines, like ours.

But multiplicity needs lines in directions that spans the space.

Otherwise some directional derivatives cannot be recovered.

Our first sampling step restricts us to a subspace:

Can’t get derivatives outside that subspace.

Similar sampler *may* work, but the samples need more structure than just being lines (a single line is not sufficient for correcting even a single symbol).

However, our techniques give a non-uniform decoder.

Selecting f_j from Improving Set

Ideally for every i compare $C(x)_i$ to $f_j(w)_i$ to see how good f_j is.

Don’t have access to C(x), but

In expectation a random $f_k(w)$ is close to C(x).

Choose the f_j such that $f_j(w)$ agrees with the most $f_k(w)$ at the most indexes.

Runtime of Algorithm

Selecting a function takes $O(q\ m^2\ n)$ queries to the level before it.

A query to level L takes q^L queries.

Final decoder only requires space q L and time

$$O(n\ m^2 q^L).$$

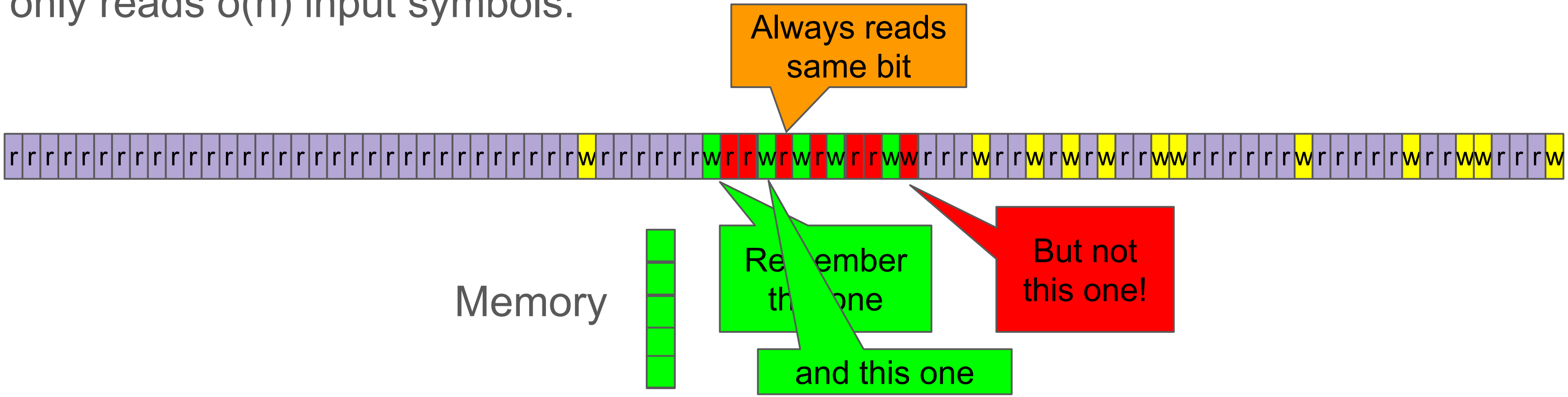
As long as $\eta \ll 1/q$, q^L will be small (if $\eta = q^{-a}$ then $q^L = n^{1/a}$).

Non Adaptive, Deterministic Decoders Fail

Gronemeier proved that a non-adaptive decoder deterministic decoder fails.

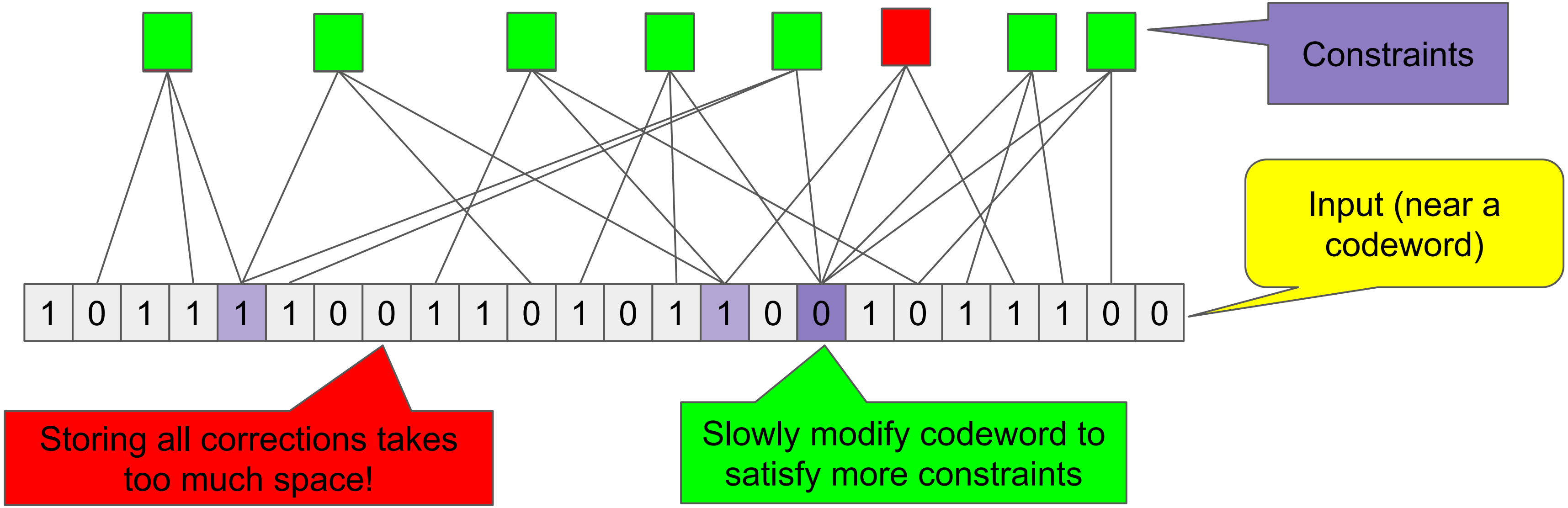
Non-adaptive means when and where the decoder reads and when it writes are independent of the input.

Idea: for space S, wait for an interval where the decoder outputs S+1 symbols and only reads o(n) input symbols.


Always reads same bit
r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r r w r r r r r r w r r w r w r w r r w w r r r w r r w r w r w r r w w r r r r r r w r r r r r w r r w w r r r w
Memory
Remember this one
and this one
But not this one!

Decoding Expander Codes in Linear Time

Standard approaches to linear time decoding often require storing partially corrected codeword in memory and making iterative corrections.


Constraints
Input (near a codeword)
1 0 1 1 1 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0
Storing all corrections takes too much space!
Slowly modify codeword to satisfy more constraints

Sampler Construction

First sample a subspace.

Epsilon biased sets in extension field. Gives a line sampler, which is a subspace sampler over original field.

Sub-sample with curves.

Since subspace is small, use all low degree curves as a sampler.

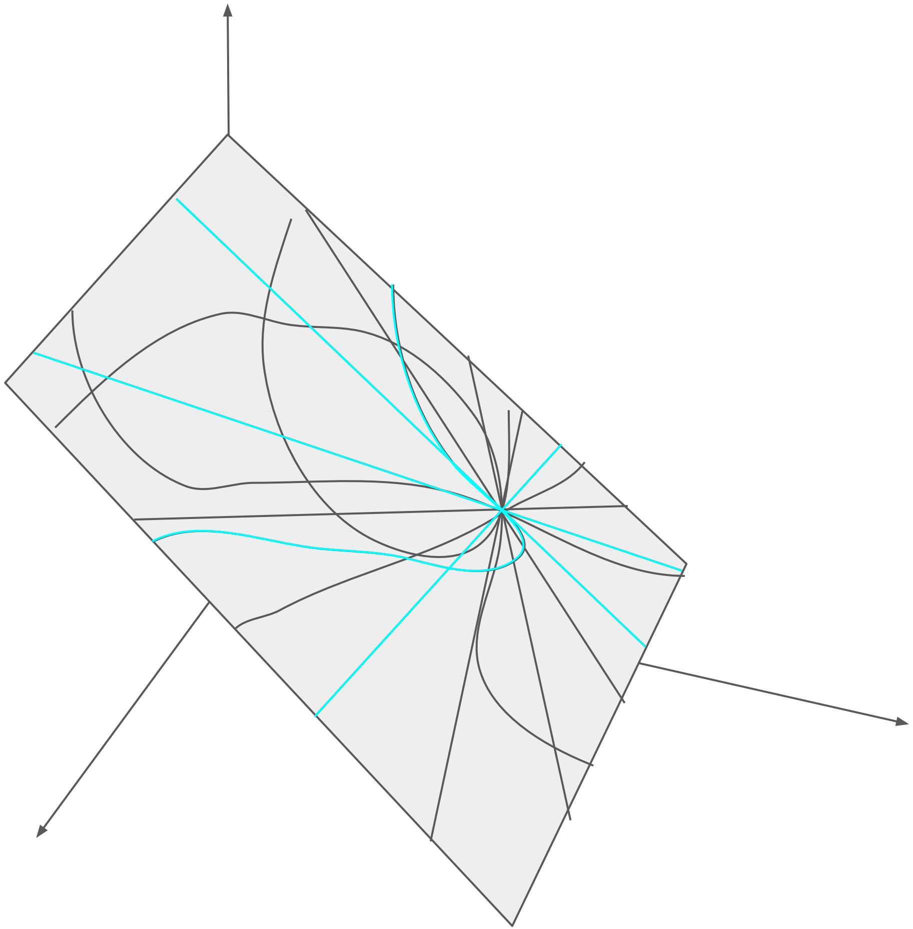
Choose **one** curve.

Do low degree correction on that curve.

Or a sample a few lines through a point in that subspace.

Choose **a few** lines.

Correct on each line and take a majority.



Making Our Sampler

Constructed subspace an epsilon biased set for extension field.

Gives line samplers for extension field.

Lines in extension field are subspaces in base fields.

To sample the lines through a point, we use the lines that intersect a random, low degree curve.