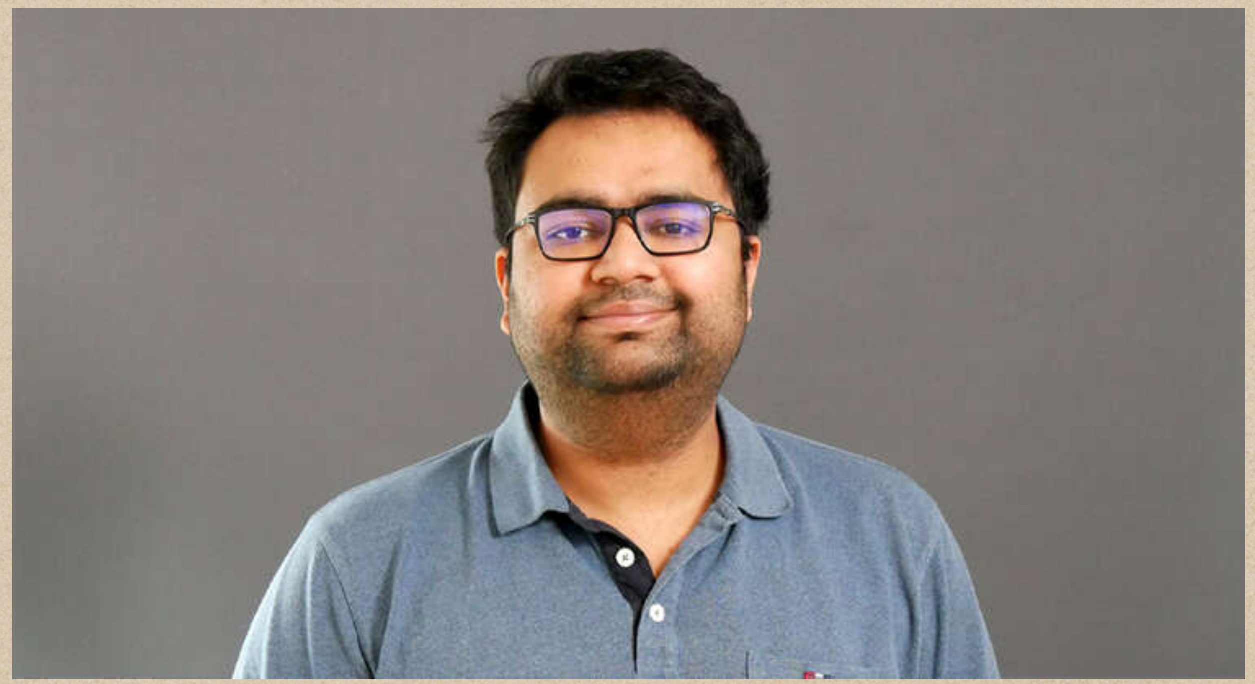
Local Correction Algorithms for Low-degree Polynomials

Amik Raj Behera (Uni. of Copenhagen)

with

Prashanth Amireddy
(Harvard)



Manaswi Paraashar
(Uni. of Copenhagen)

Srikanth Srinivasan
(Uni. of Copenhagen)

Madhu Sudan
(Harvard)

What is this talk about?

Local Correction Algorithms

for

Low-Degree Polynomials

What is this talk about ?

Local Correction Algorithms

for

Low-Degree Polynomials

↓

Good Codewords

What is this talk about?

Local **Correction** Algorithms

Recovering corrupted code words

for

Low - Degree Polynomials

Good Codewords

What is this talk about ?

Local

Super efficiently

Correction

Recovering corrupted code words

Algorithms

for

Low-Degree Polynomials

Good Codewords

What is this talk about?

Local

Super efficiently

Correction

Algorithms

Recovering corrupted code words

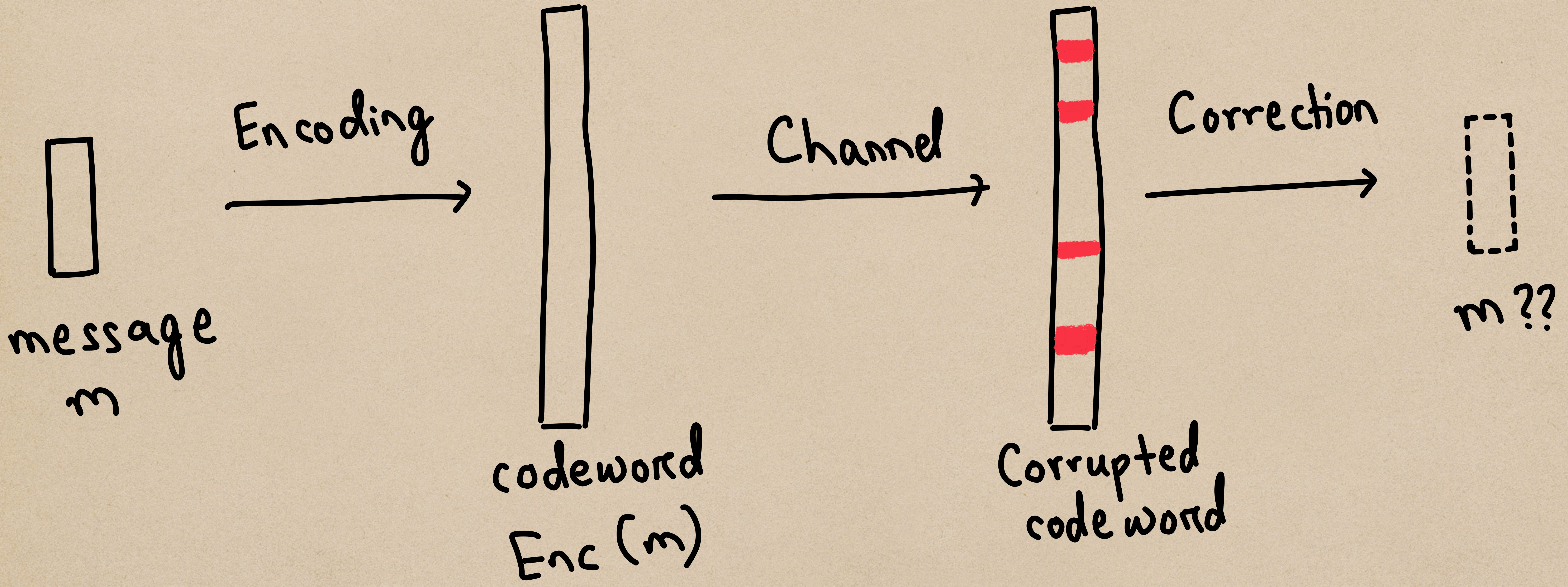
for

Low-Degree Polynomials

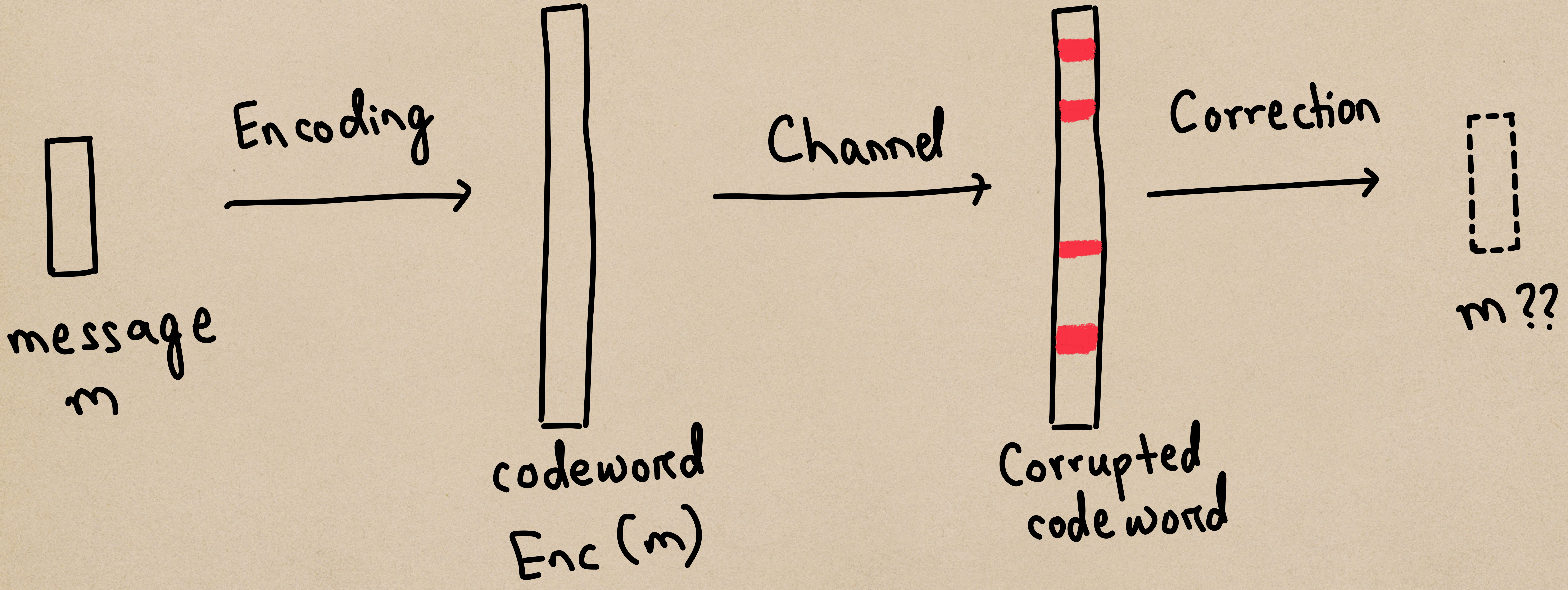
Good Codewords

What is Error Correction ?

What is Error Correction ?

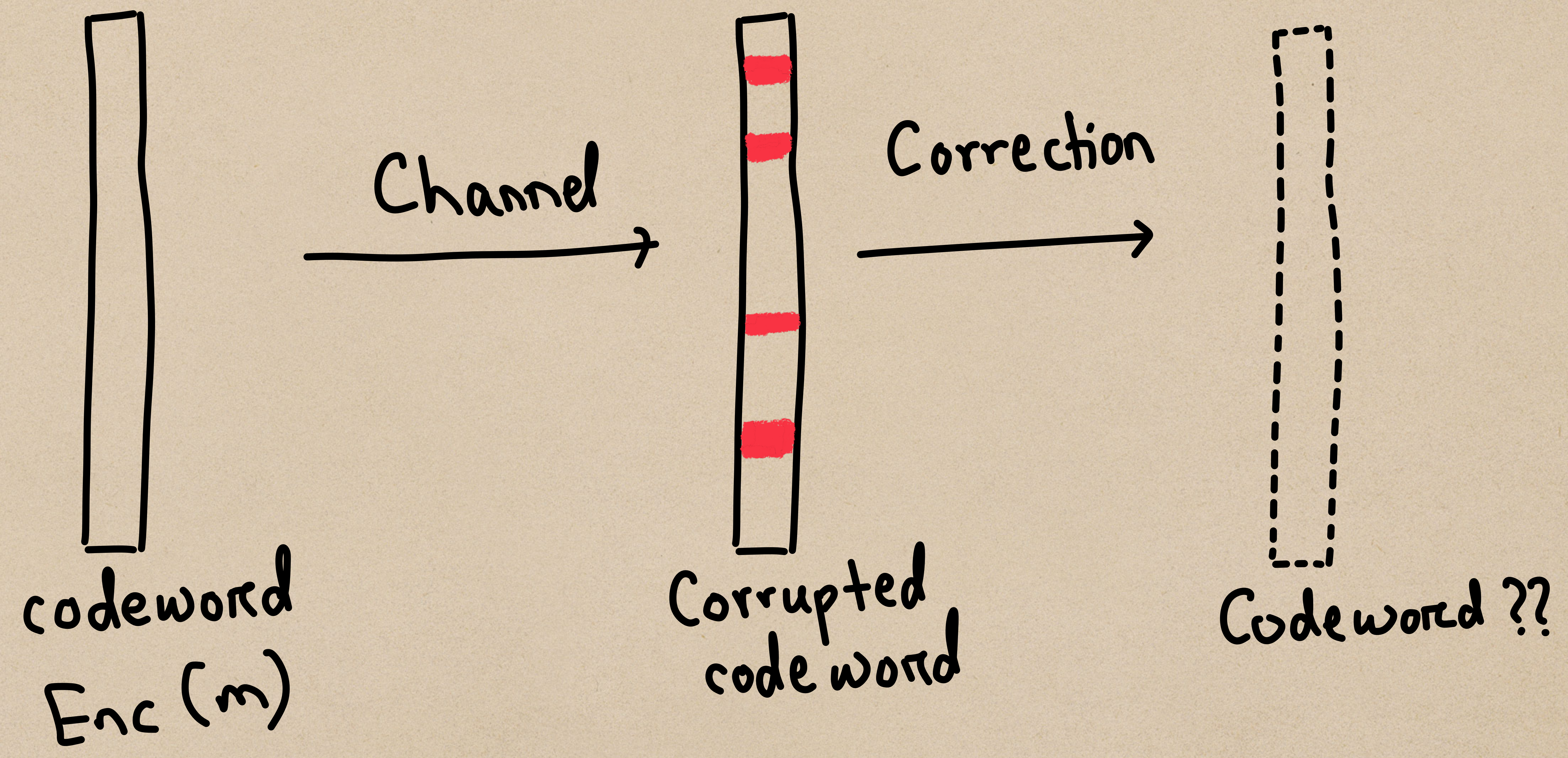


What is Error Correction ?



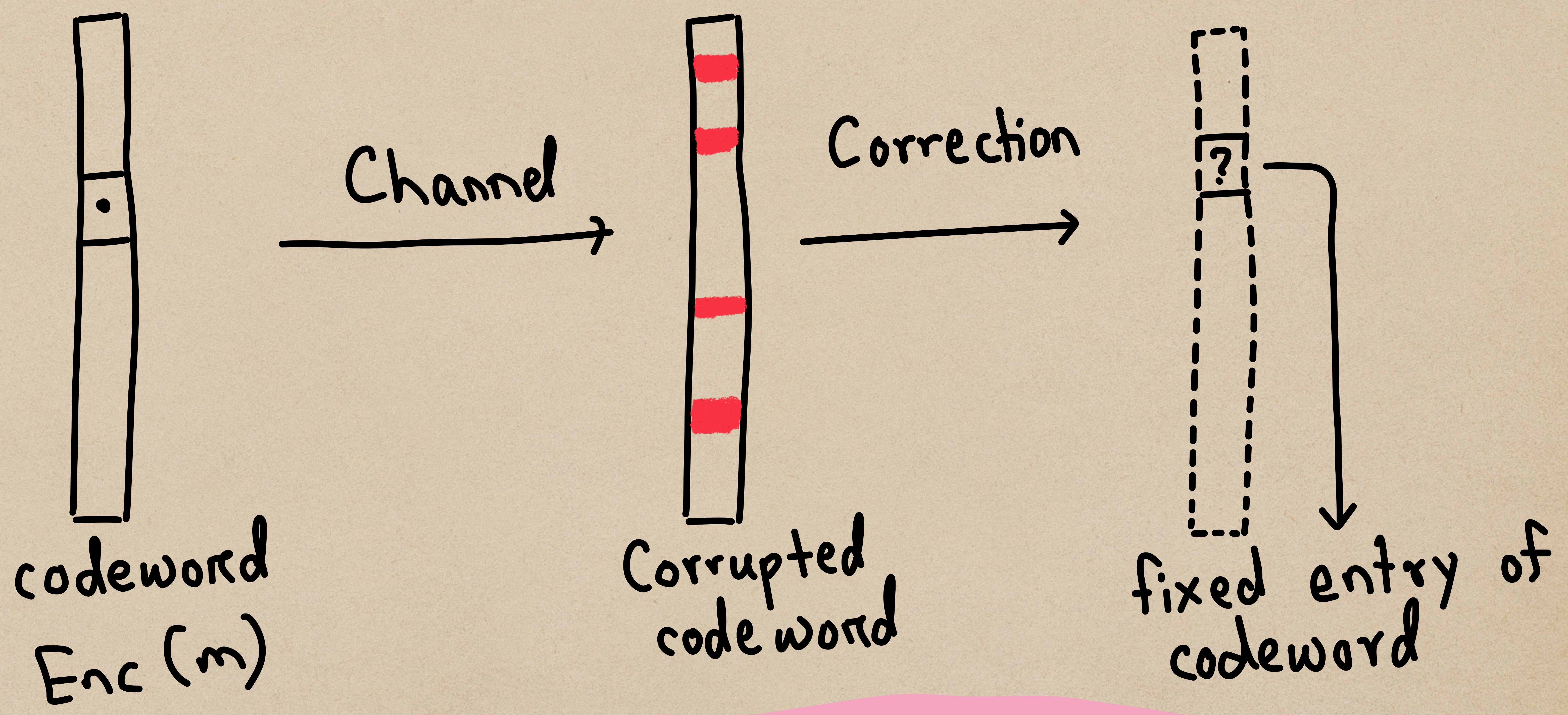
Error Correction : Given a corrupted codeword, Recover the original message

What is Error Correction ?

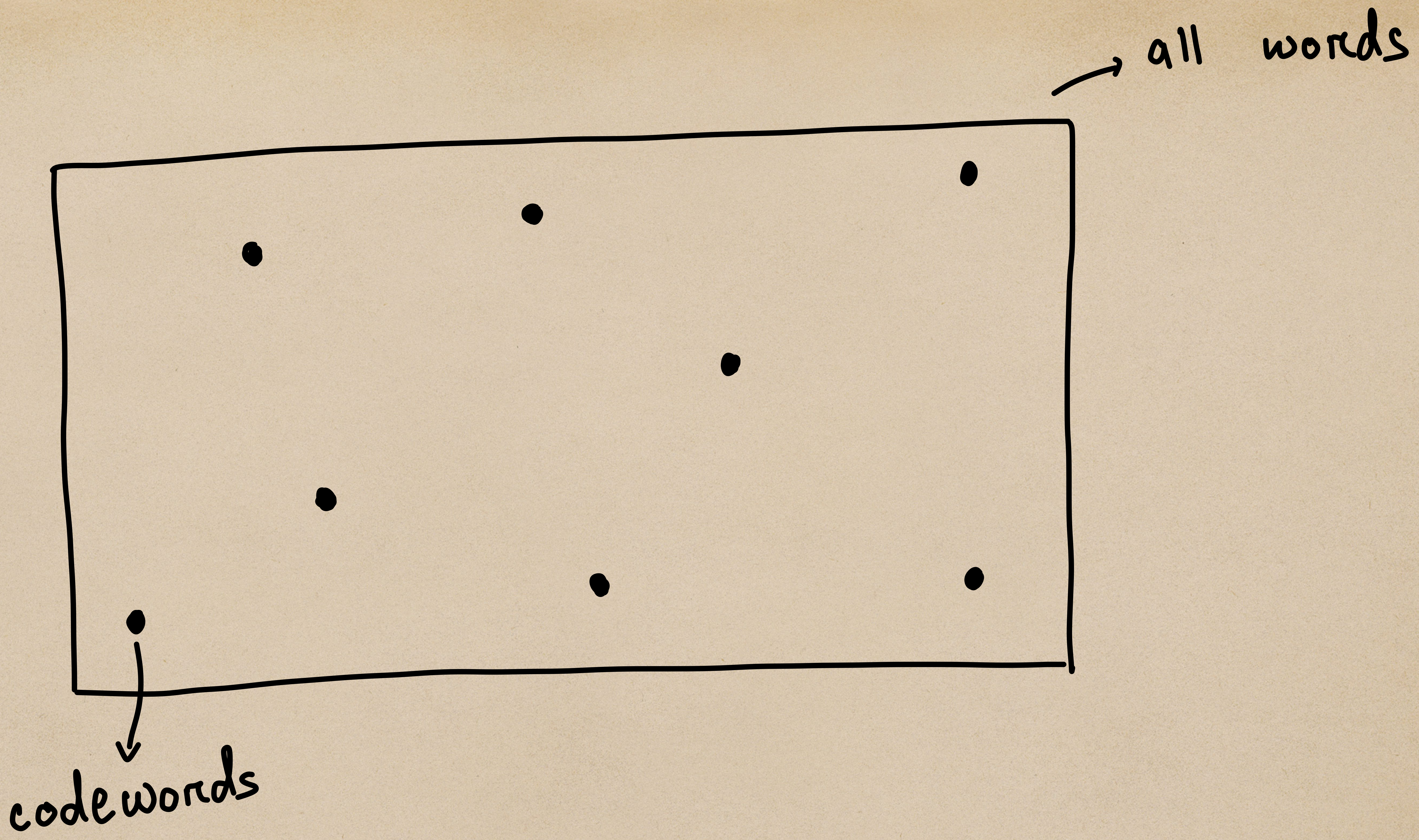


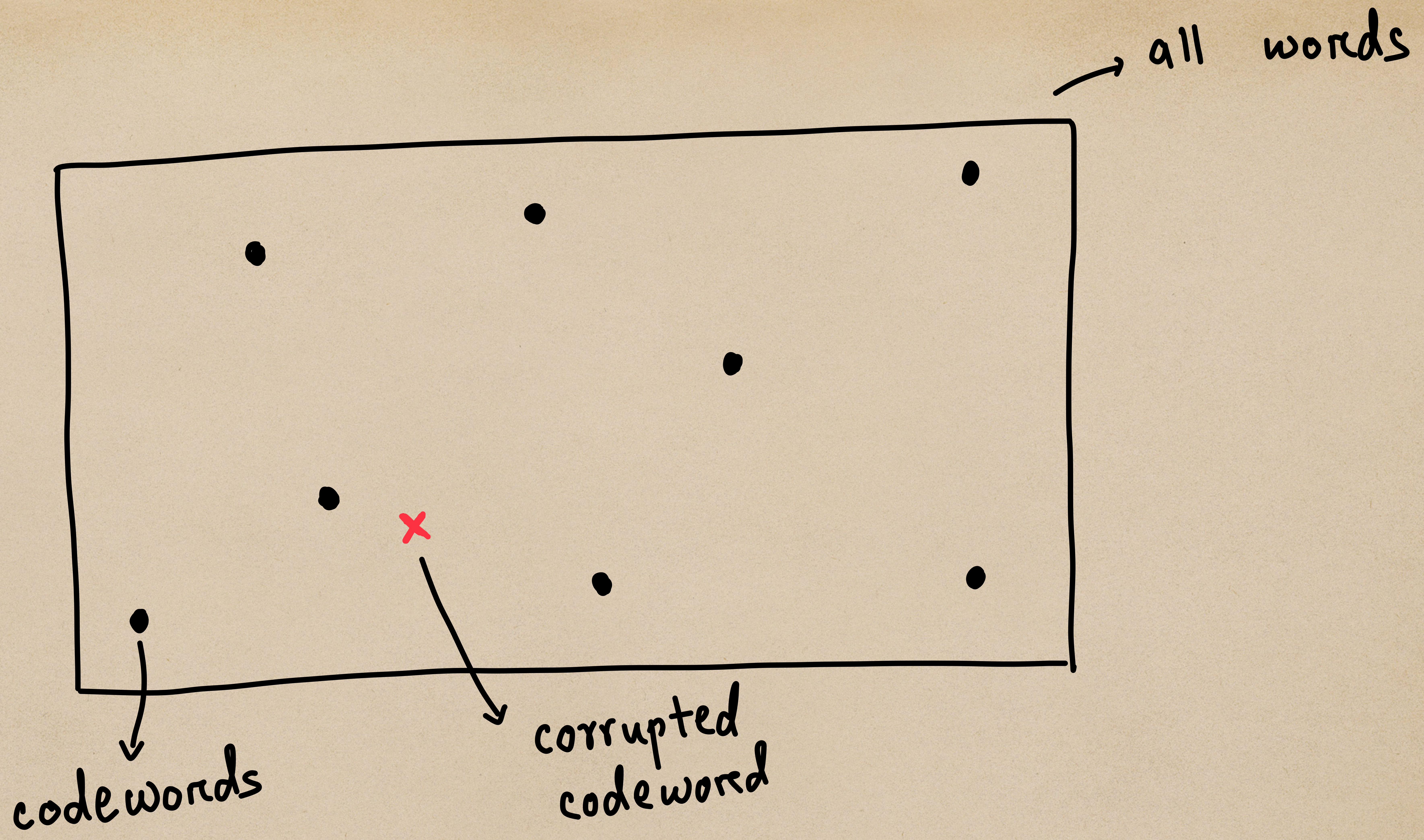
Error Correction : Given a corrupted codeword, Recover the original codeword

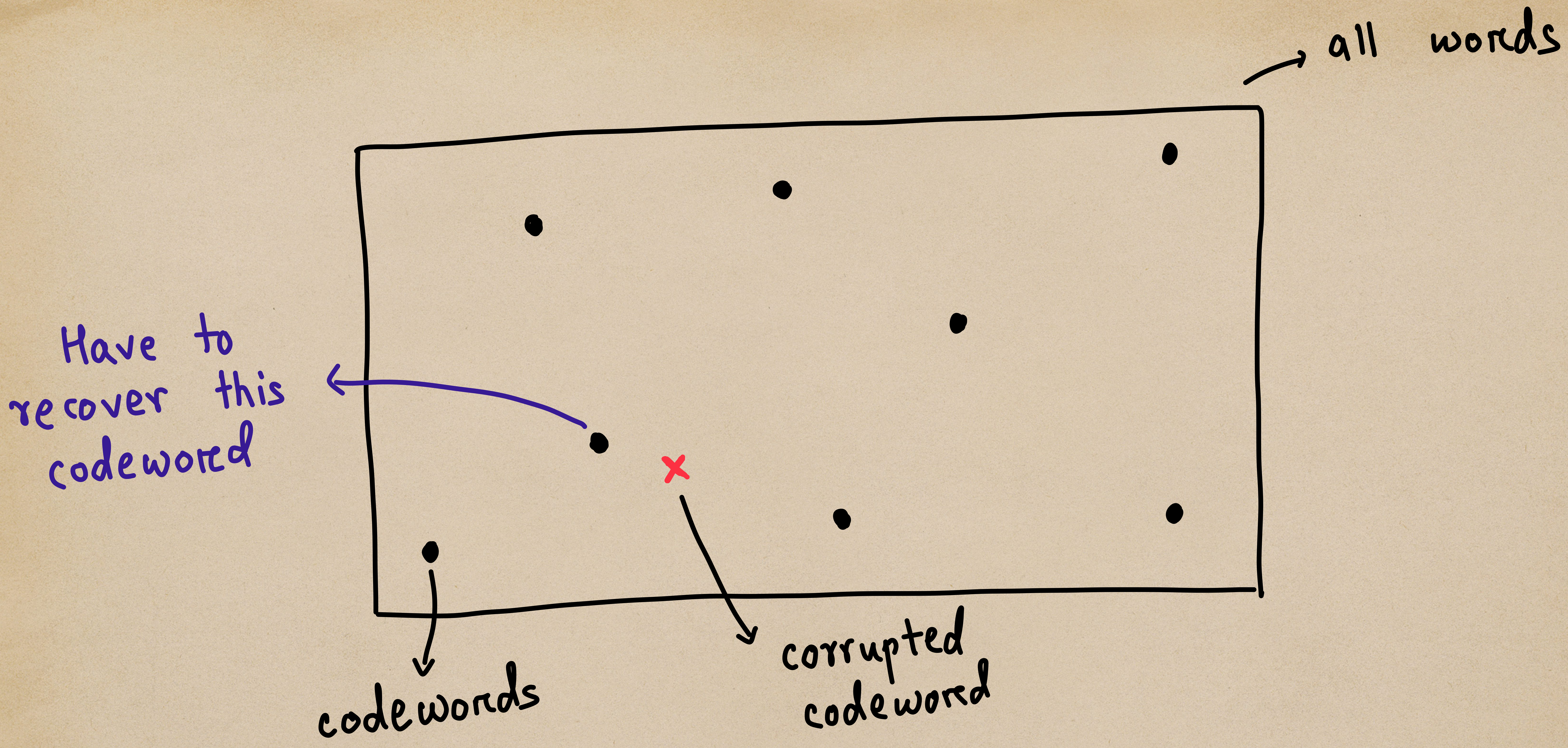
What is Local Correction ?

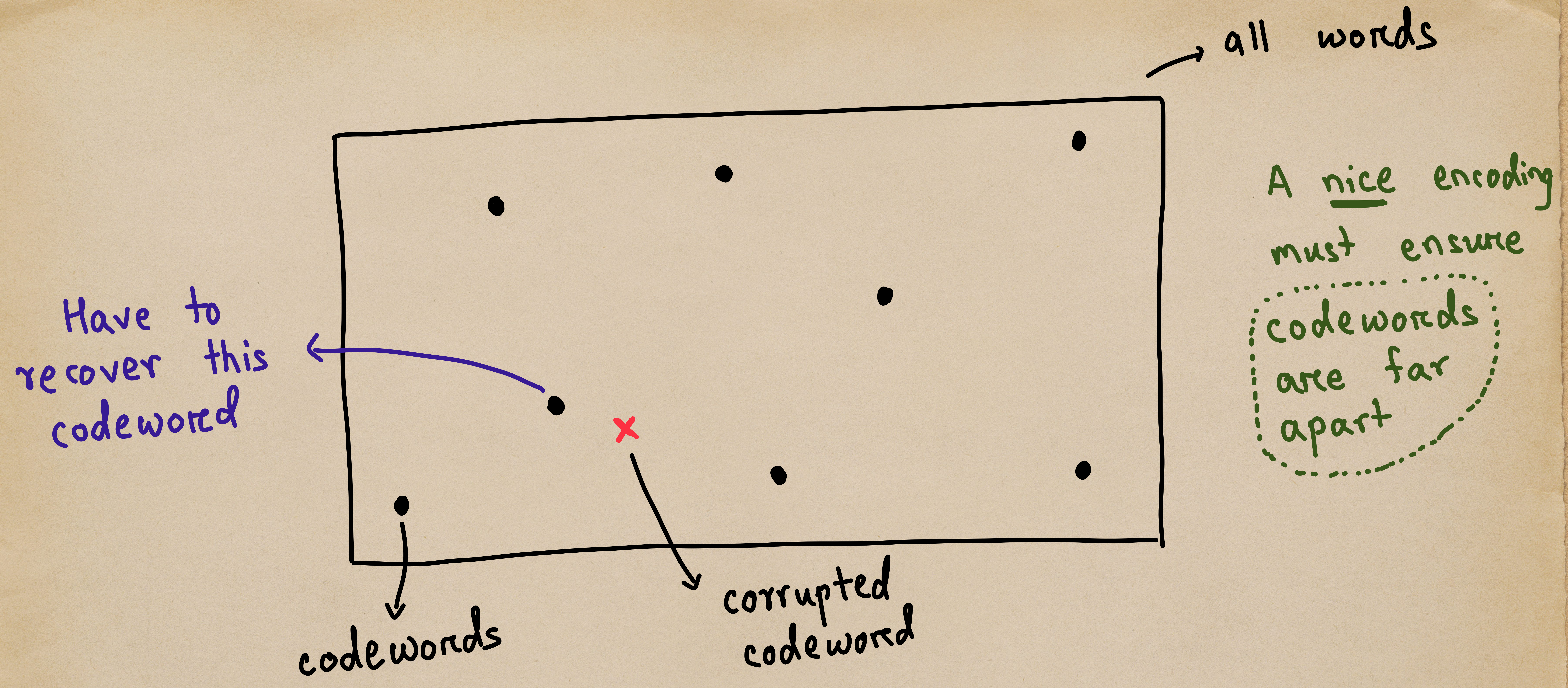


Local Correction : Given a corrupted codeword, Recover a fixed entry of the original codeword all words


all words
codewords


all words
codewords
corrupted codeword


all words
Have to recover this codeword
codewords
corrupted codeword


all words
A nice encoding must ensure codewords are far apart
Have to recover this codeword
codewords
corrupted codeword

Low-degree Polynomials as Error Correcting Codes

$P(x_1, x_2, \ldots, x_n) : \{0,1\}^n \longrightarrow \mathbb{R}$ — deg 1 poly.

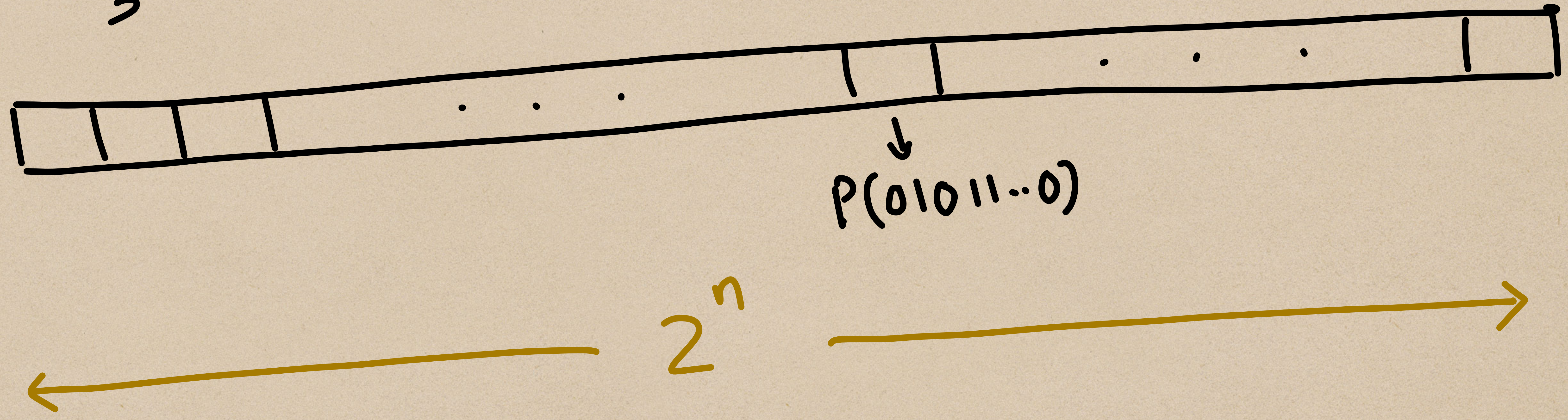
E.g. $4x_1 - \frac{2}{3}x_2 + 5x_{10} - 99$

Low-degree Polynomials as Error Correcting Codes

$P(x_1, x_2, \ldots, x_n) : \{0,1\}^n \longrightarrow \mathbb{R}$ — deg 1 poly.

E.g. $4x_1 - \frac{2}{3}x_2 + 5x_{10} - 99$

$P \equiv$

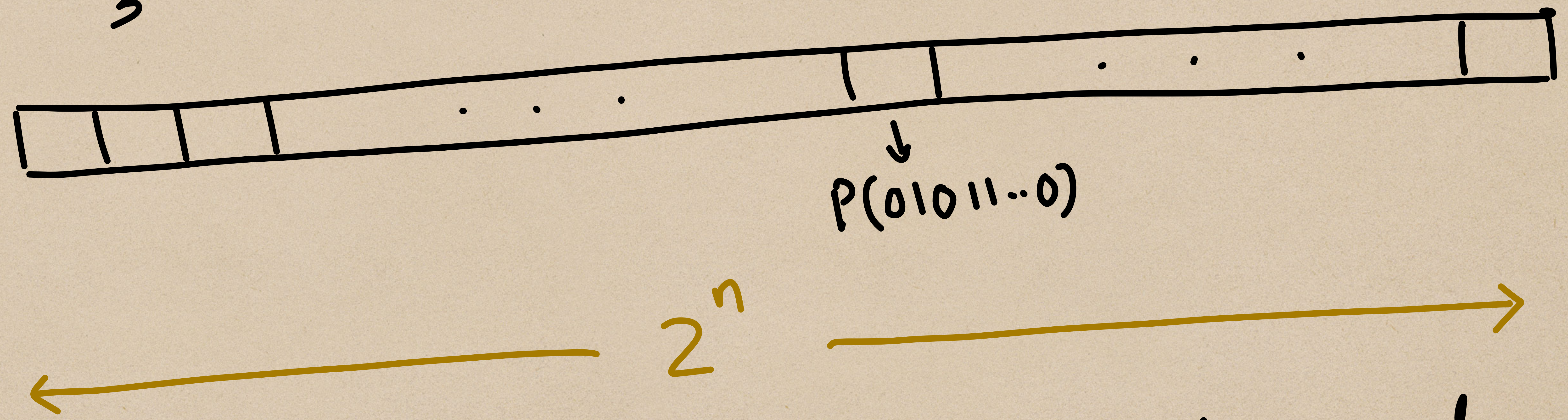


Low-degree Polynomials as Error Correcting Codes

$P(x_1, x_2, \ldots, x_n) : \{0,1\}^n \longrightarrow \mathbb{R}$ — deg 1 poly.

E.g. $4x_1 - \frac{2}{3}x_2 + 5x_{10} - 99$

$P \equiv$



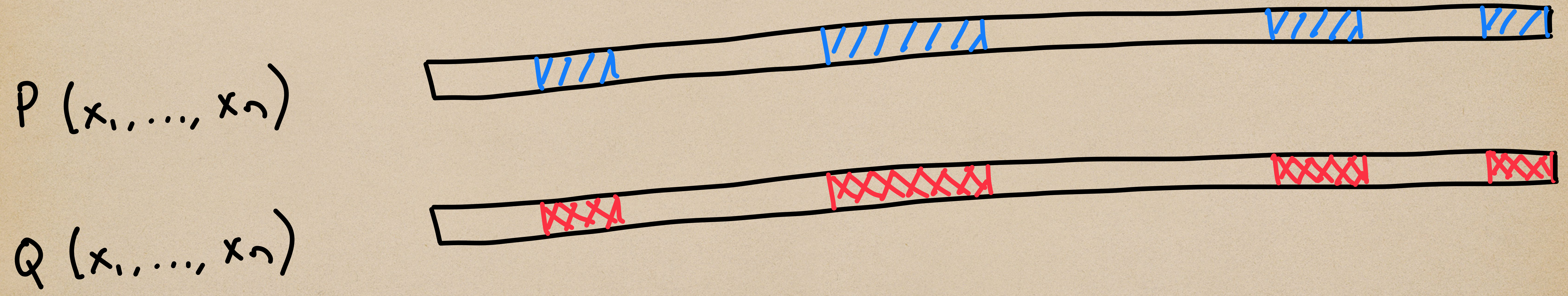
Codewords : Evaluation table of deg-1 polynomials over $\{0,1\}^n$

Low-degree Polynomials are Nice

Fact: Two distinct degree-1 polynomials on $\{0,1\}^n$ disagree on $\geq \frac{1}{2}$-fraction of $\{0,1\}^n$.

Low-degree Polynomials are Nice

Fact: Two distinct degree-1 polynomials on $\{0,1\}^n$ disagree on $\geq \frac{1}{2}$-fraction of $\{0,1\}^n$.

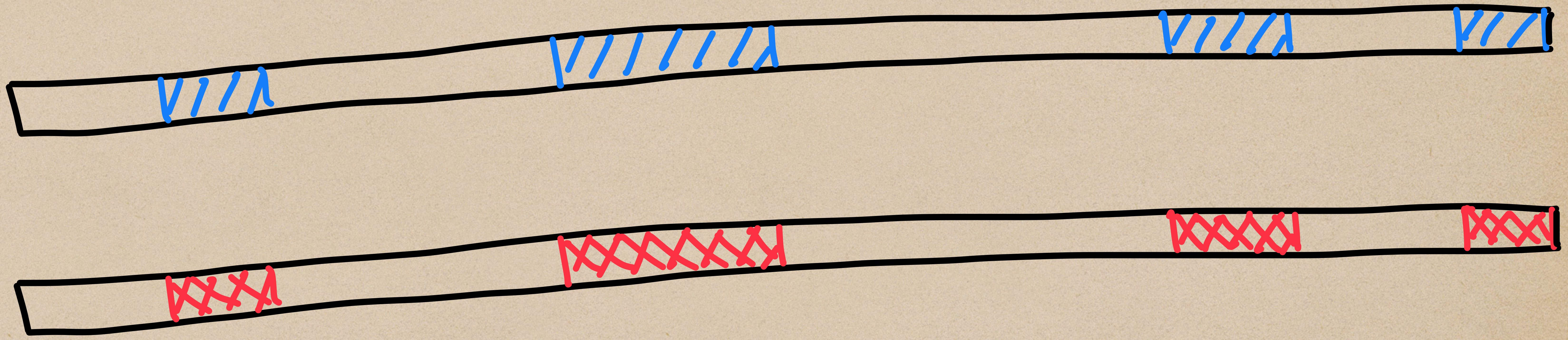


Low-degree Polynomials are Nice

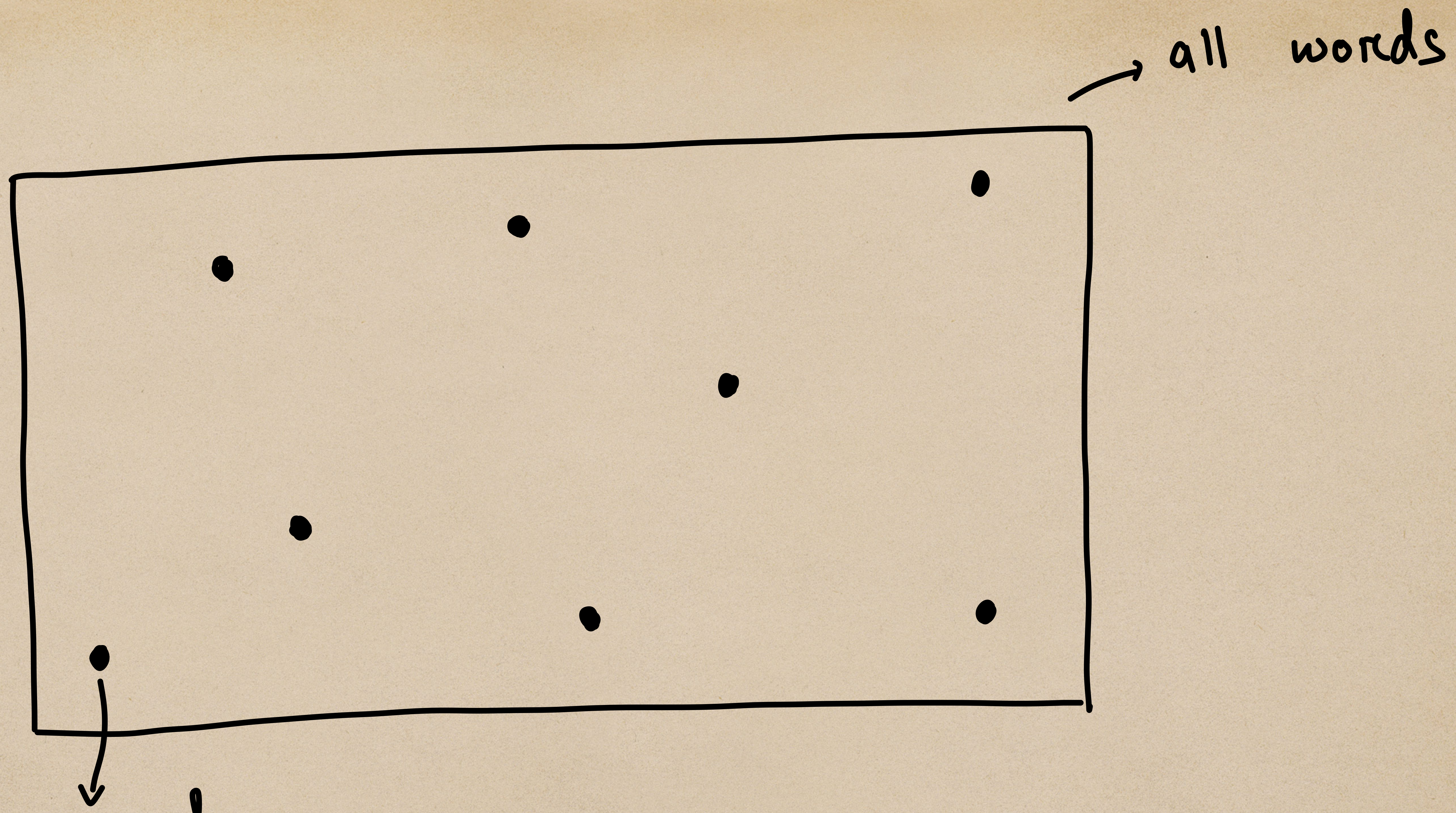
Fact: Two distinct degree-1 polynomials on $\{0,1\}^n$ disagree on $\geq \frac{1}{2}$-fraction of $\{0,1\}^n$.

$P(x_1, \dots, x_n)$

$Q(x_1, \dots, x_n)$

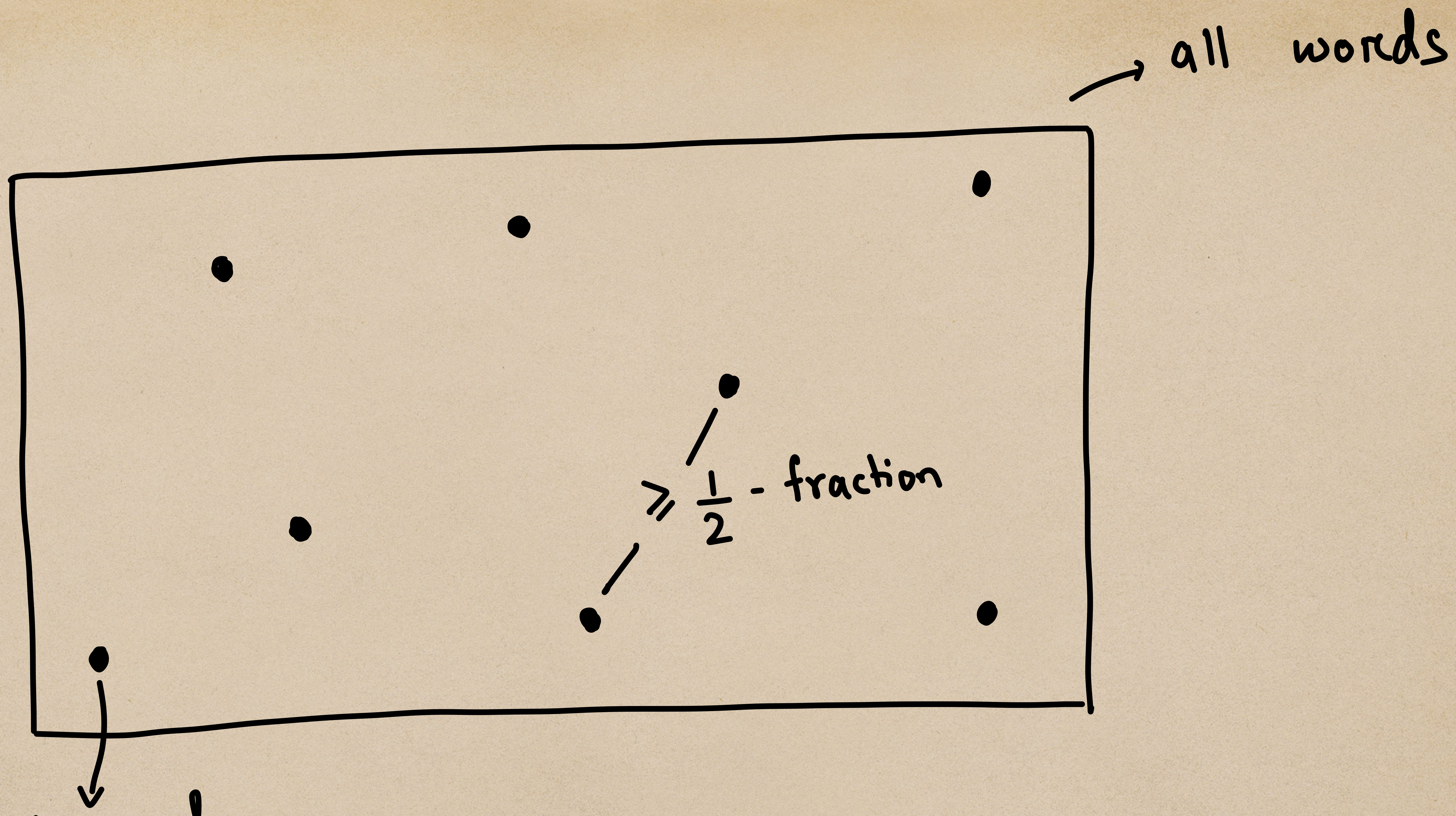


"If they disagree on a single point, they disagree on a lot of points."

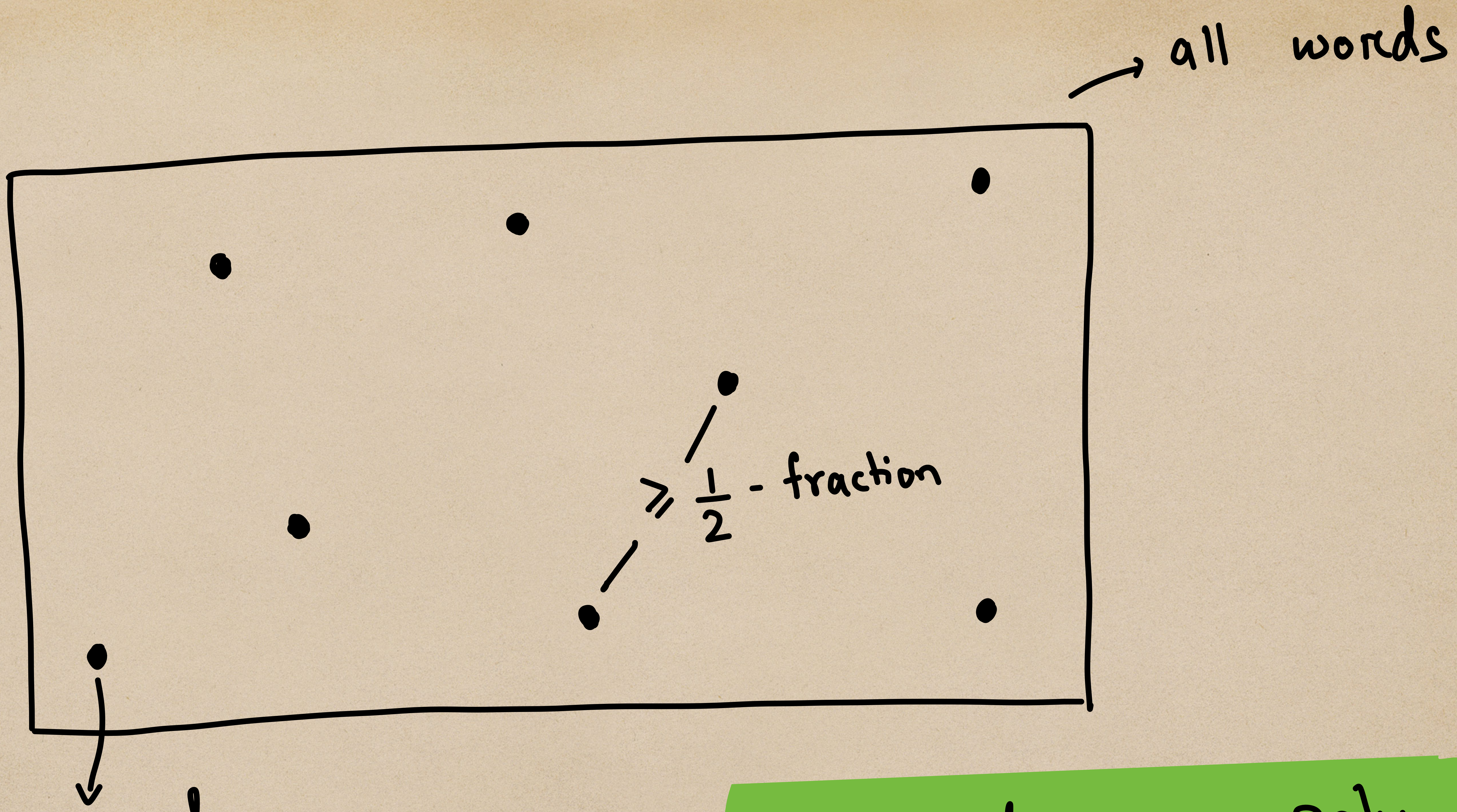


codewords

= Evaluation of deg. 1 poly. on $\{0,1\}^n$



codewords
= Evaluation of deg. 1
poly. on $\{0,1\}^n$



codewords
= Evaluation of deg. 1
poly. on $\{0,1\}^n$

Low-degree poly. are far apart.

Low-degree are Quite Useful

— Randomized Algorithms, e.g. matching

— Probabilistically Checkable Proofs (PCPs)

— Combinatorics, e.g. Kakeya's conjecture, Polynomial Method

⋮

Correction Algorithm for Low-degree Polynomials

Input $\quad f : \{0,1\}^n \longrightarrow \mathbb{R}$

$f \equiv$ 

Correction Algorithm for Low-degree Polynomials

Input $f : \{0,1\}^n \to \mathbb{R}$

Corruption at unknown places

$f \equiv$

$p \equiv$

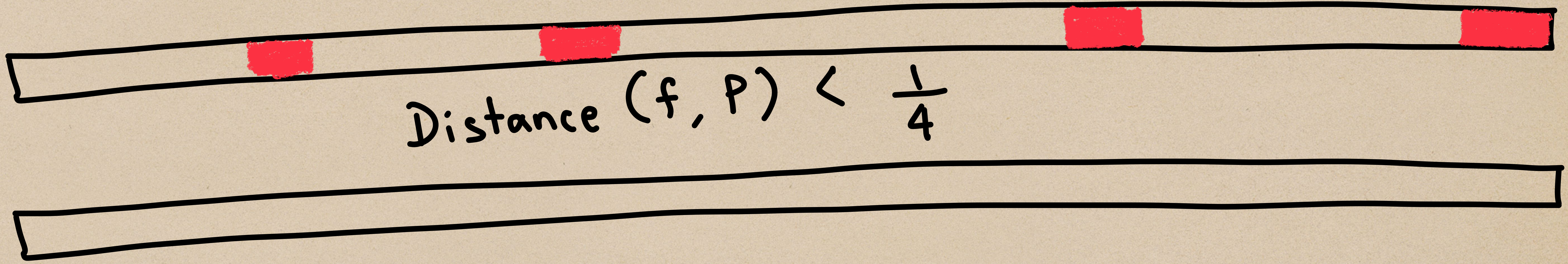
p → degree-1 poly.

Correction Algorithm for Low-degree Polynomials

Input $f : \{0,1\}^n \longrightarrow \mathbb{R}$

Corruption at unknown places

$f \equiv$



$P \equiv$

$\downarrow$

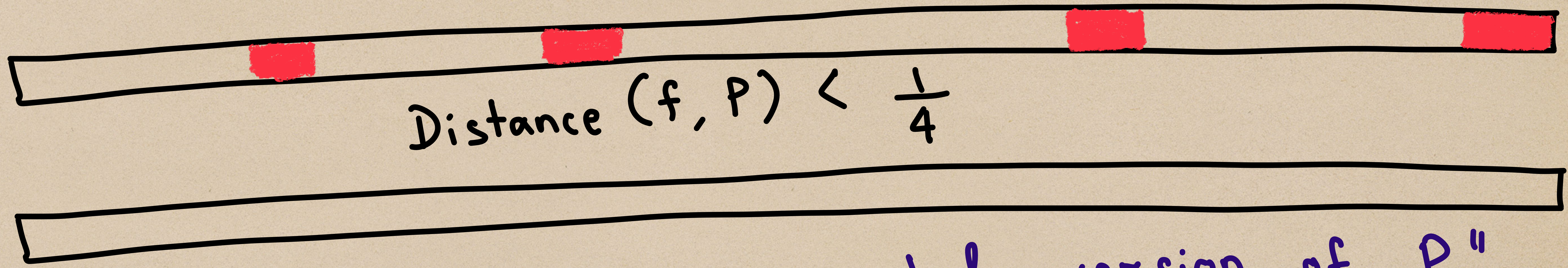
degree-1 poly.

Correction Algorithm for Low-degree Polynomials

Input $f : \{0,1\}^n \longrightarrow \mathbb{R}$

Corruption at unknown places

$f \equiv$



$\text{Distance}(f, P) < \frac{1}{4}$

$P \equiv$

"f is a corrupted version of P"

degree-1 poly.

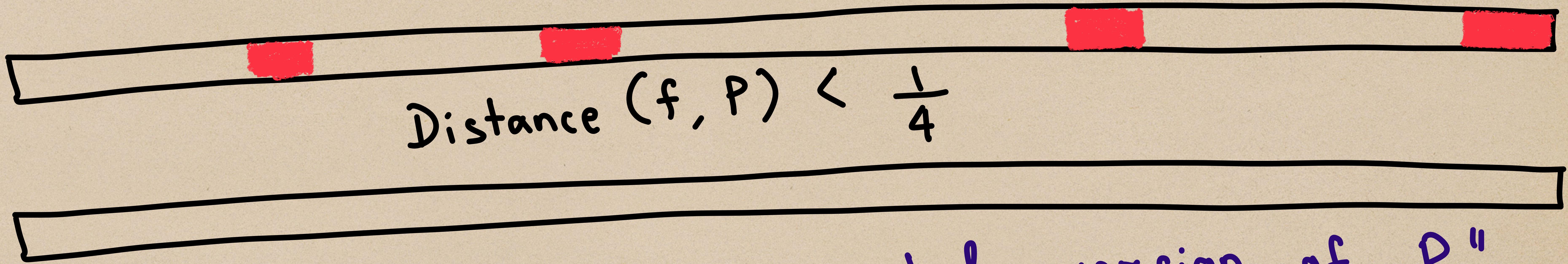
Correction Algorithm for Low-degree Polynomials

Input $f: \{0,1\}^n \to \mathbb{R}$

Corruption at unknown places

$f \equiv$

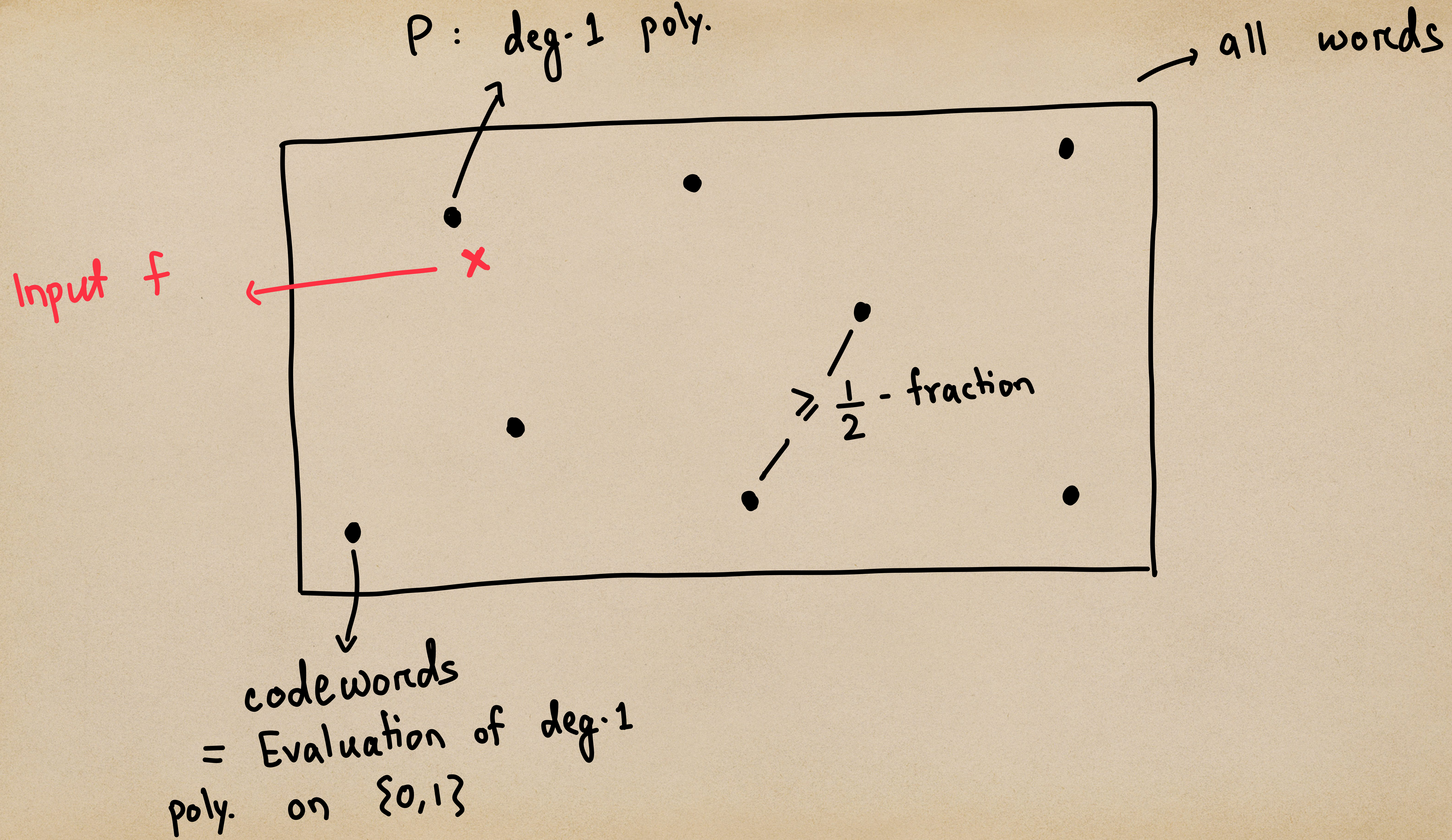
$P \equiv$



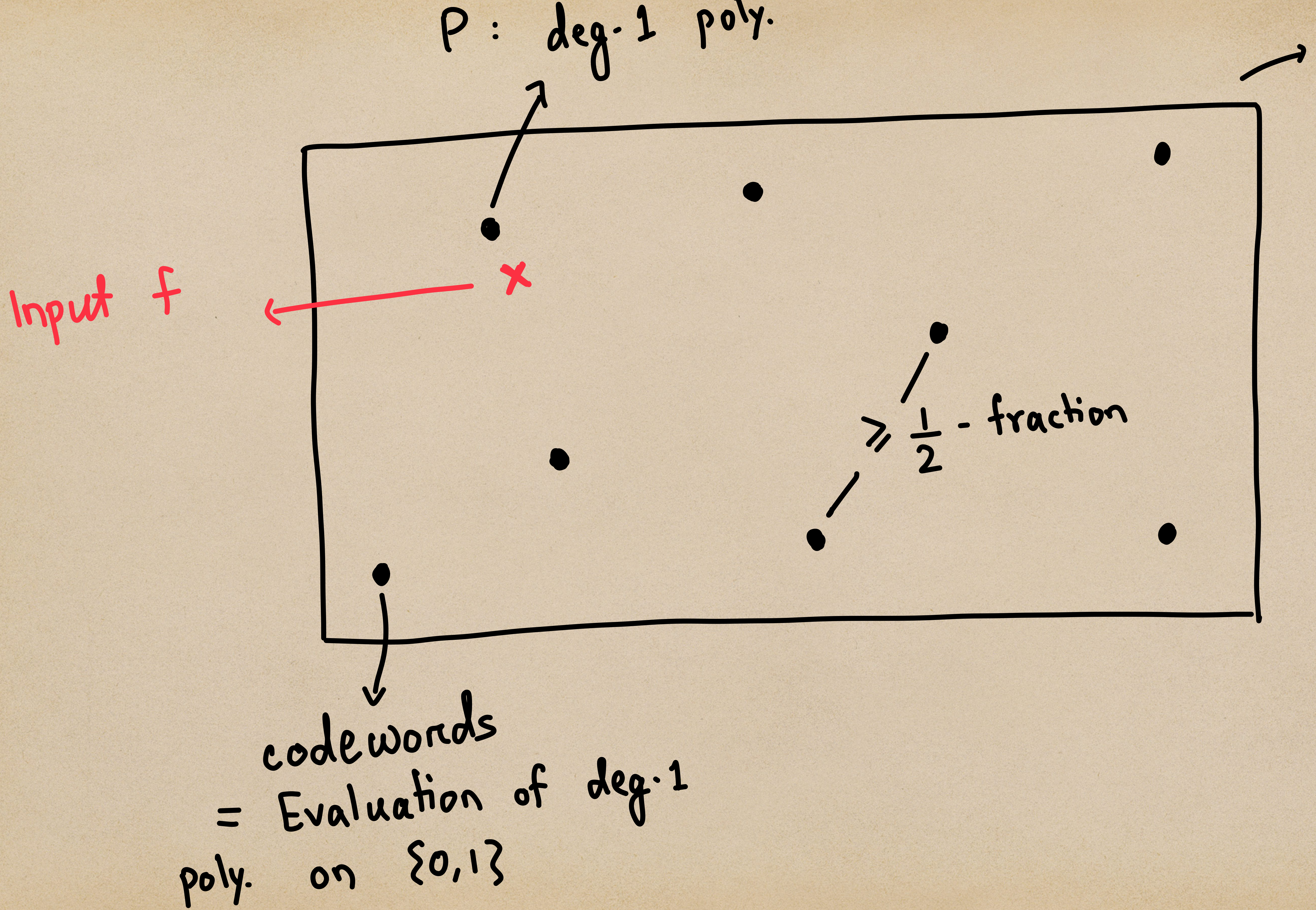
"f is a corrupted version of P"

P → degree-1 poly.

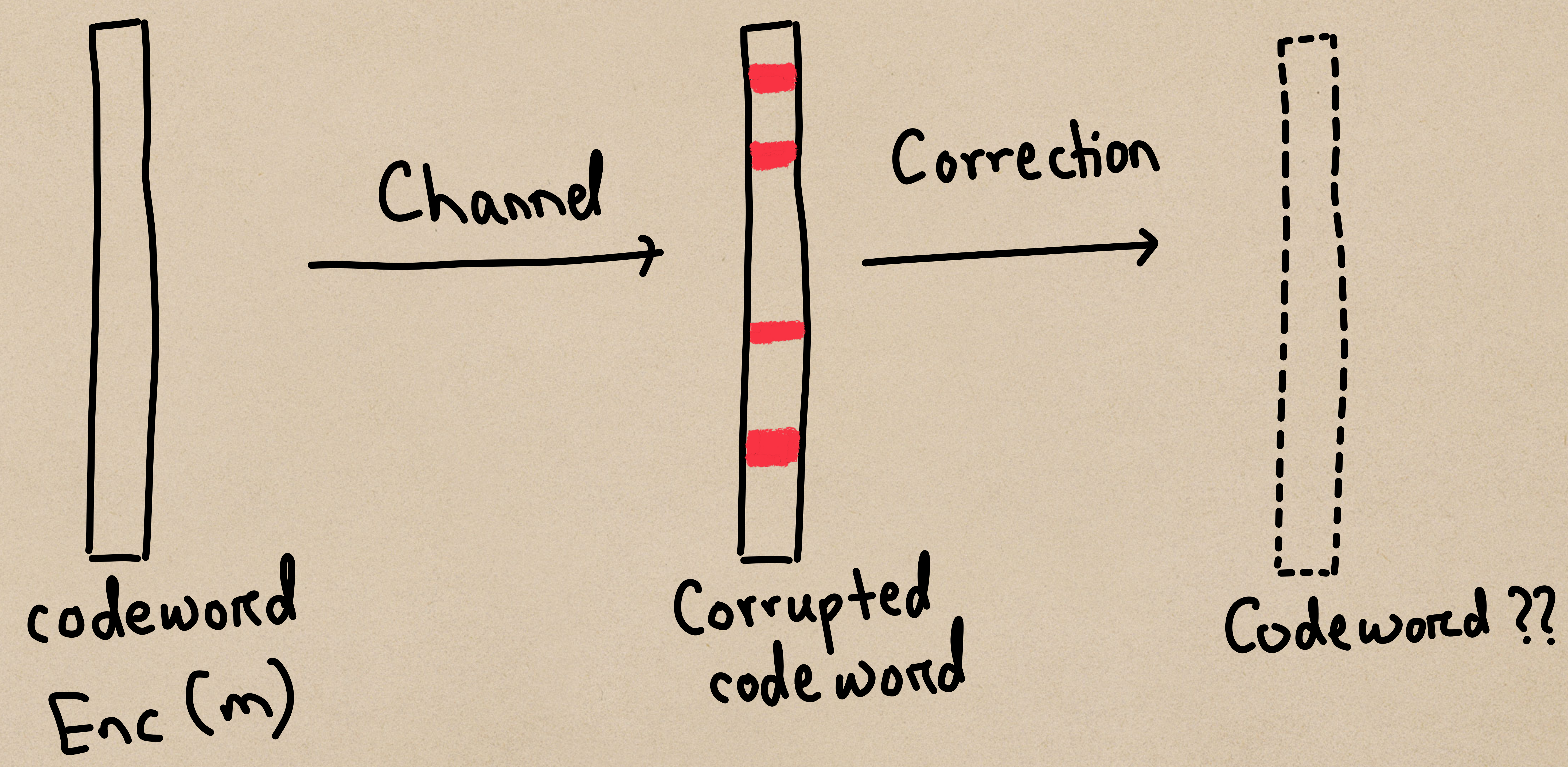
For input $f(x_1, \ldots, x_n)$, deg-1 poly. $P(x_1, \ldots, x_n)$ is unique bcoz of distance of deg-1 polynomials.

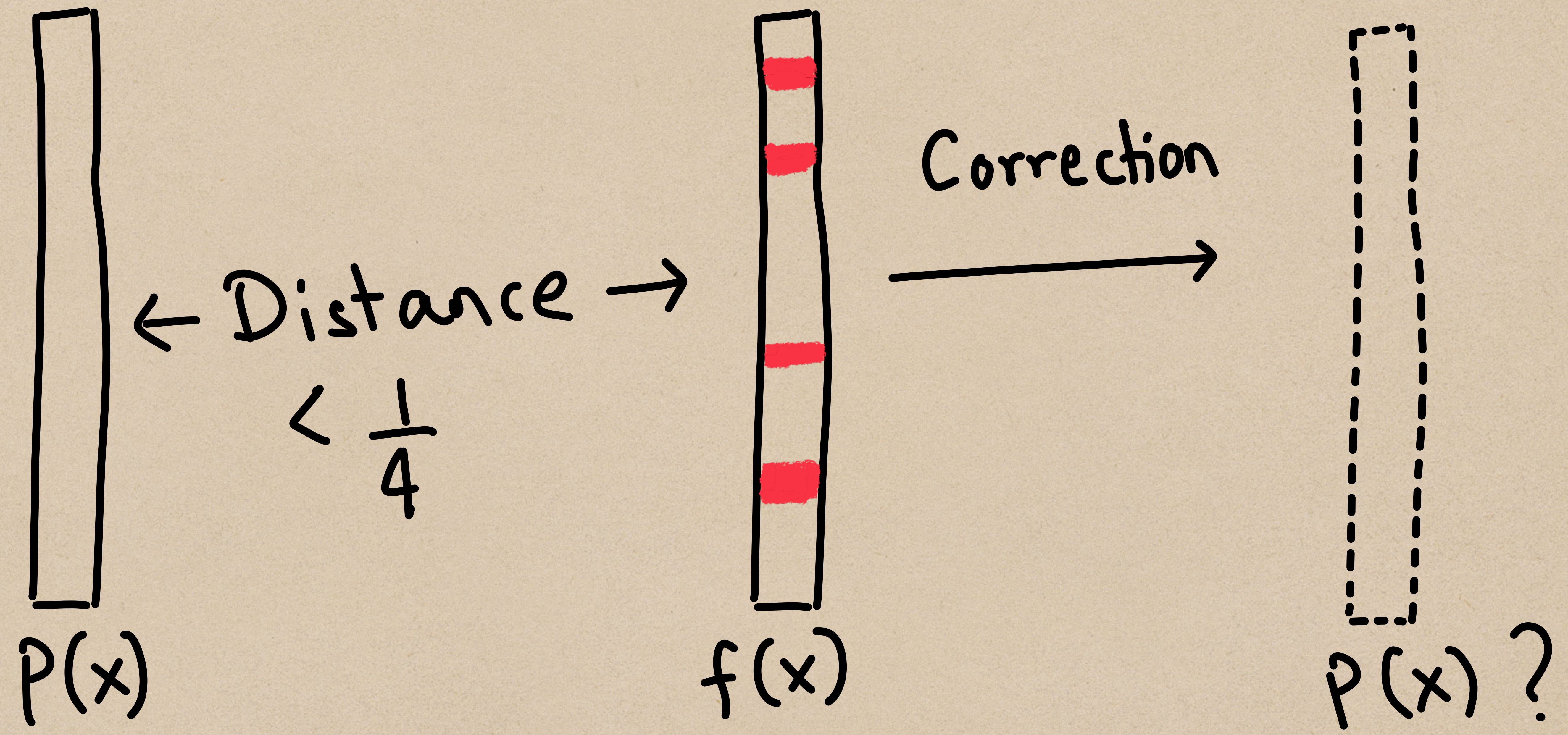


P : deg. 1 poly.
all words
Input f
$\geq \frac{1}{2}$ - fraction
codewords
= Evaluation of deg. 1
poly. on {0,1}



Given access to $f(x)$, how efficiently can we recover $P(x)$?

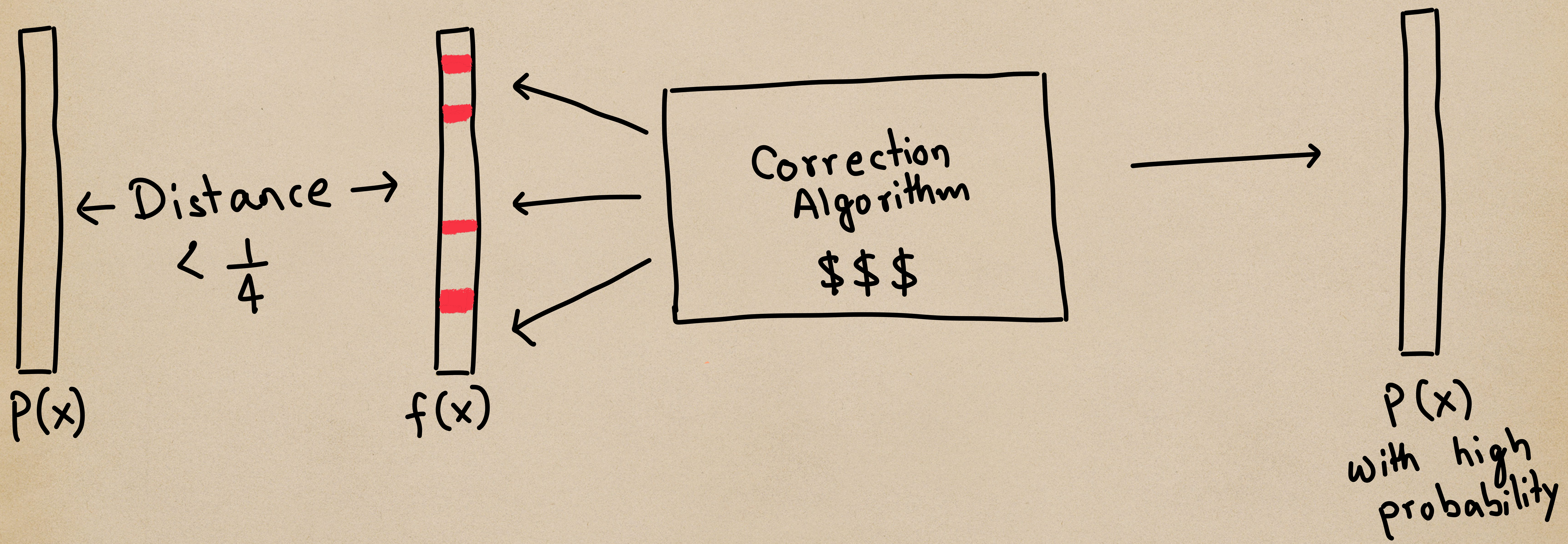

Channel
Correction
codeword
Enc (m)
Corrupted
code word
Codeword ??



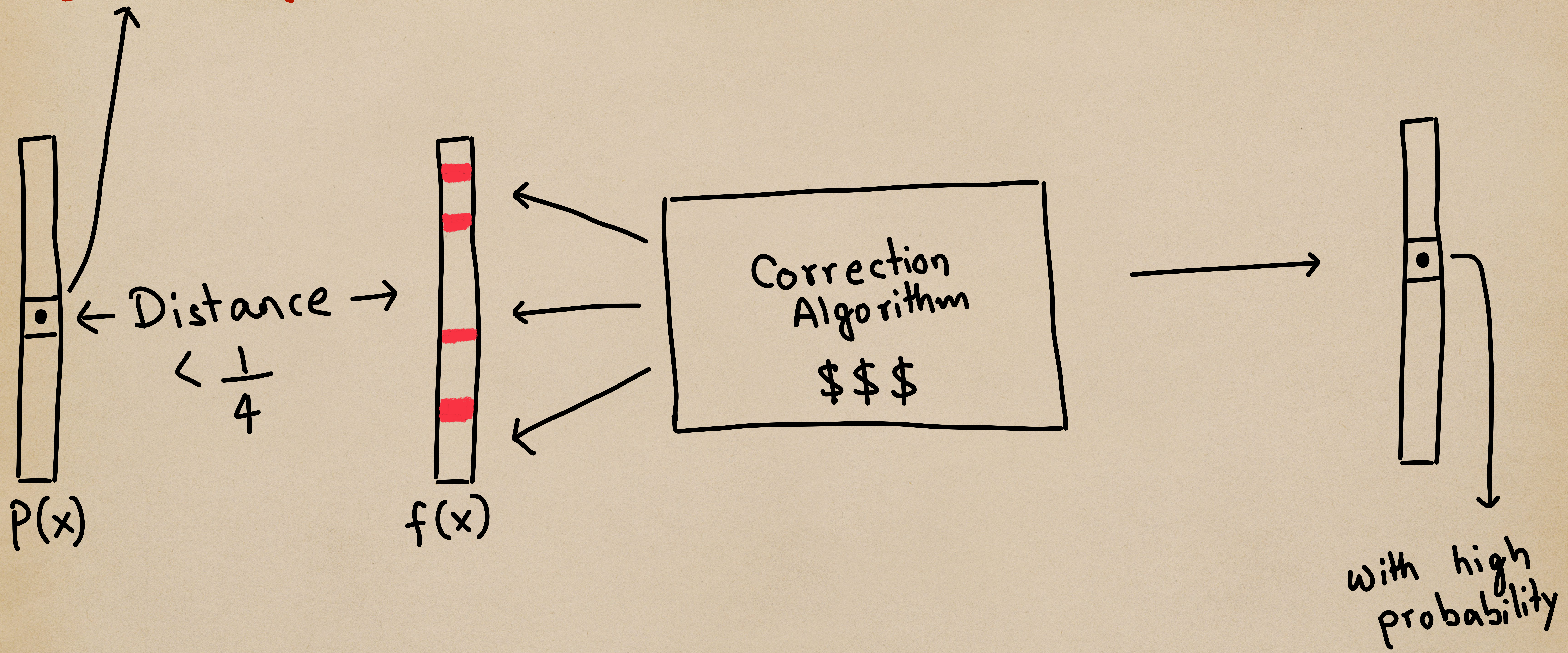
Distance
$< \frac{1}{4}$
Correction
$P(x)$
$f(x)$
$P(x)$?

Correction for Low-degree Polynomials

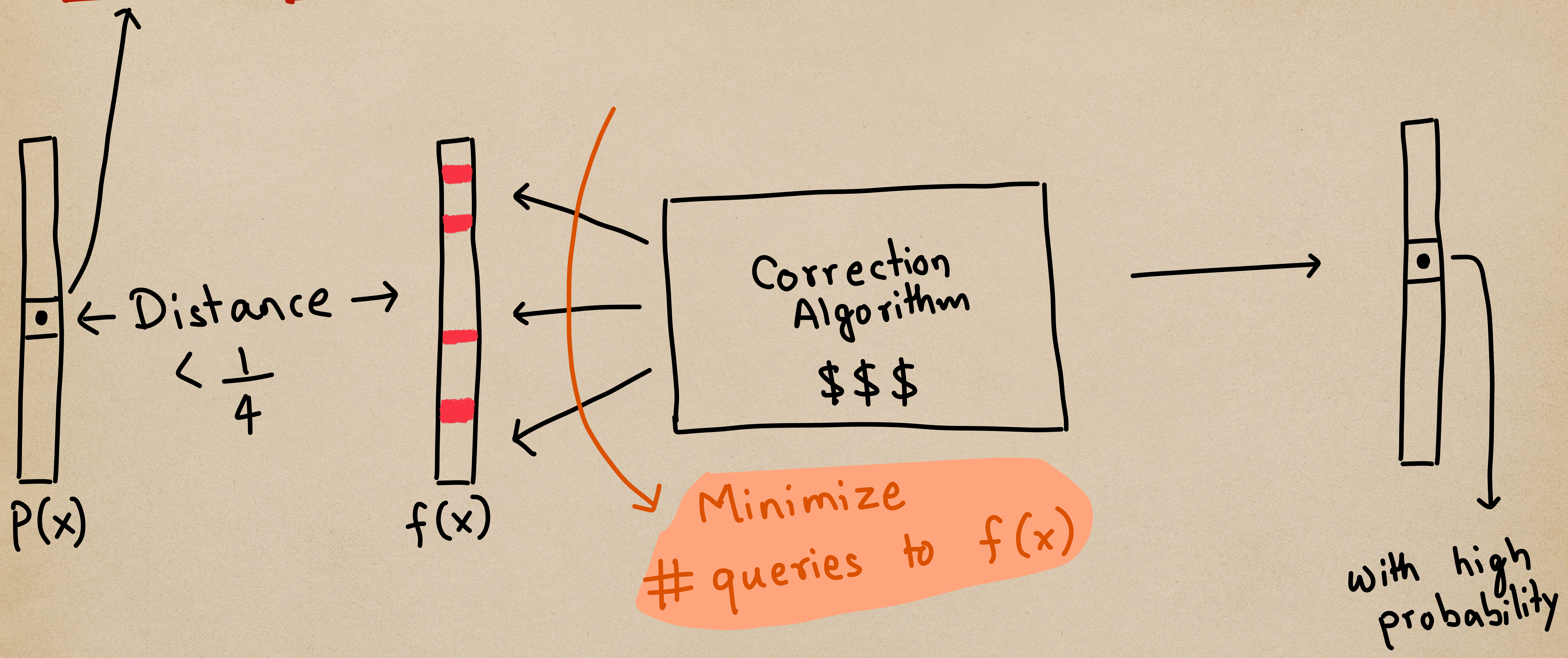
Correction for Low-degree Polynomials



Local Correction for Low-degree Polynomials



Local Correction for Low-degree Polynomials



Prior Work

- [Goldreich - Levin '89] : Linear functions $\mathbb{F}_2^n \longrightarrow \mathbb{F}_2$

Prior Work

- [Goldreich - Levin '89] : Linear functions $\mathbb{F}_2^n \to \mathbb{F}_2$
- [Beaver- Feigenbaum '90], [GLRSW '91], [Gemell- Sudan '92], [Sudan- Trevisan- Vadhan '01], [Gopalan- Klivans - Zuckerman '08], [Bhowmick - Lovett '18], ...

  Low - degree polynomials $\mathbb{F}_p^n \to \mathbb{F}_p$

Prior Work

- [Goldreich - Levin '89] : Linear functions $\mathbb{F}_2^n \to \mathbb{F}_2$
- [Beaver- Feigenbaum '90], [GLRSW '91], [Gemell- Sudan '92], [Sudan- Trevisan- Vadhan '01], [Gopalan- Klivans - Zuckerman '08], [Bhowmick - Lovett '18], ...

  Low - degree polynomials $\mathbb{F}_p^n \to \mathbb{F}_p$

- Useful in
  - Cryptography (One - way functions)
  - Pseudorandomness (Hardness Amplification)

Our Setting

- [Kim - Kopparty '17] : Correction for low-degree polynomials in poly(n) queries

Our Setting

- [Kim-Kopparty '17]: Correction for low-degree polynomials in poly(n) queries

- [Bafna-Srinivasan-Sudan '20]: Local correction for low-degree polynomials

$$P(x) : \{0,1\}^n \to \mathbb{F}_q$$

(constant characteristic)

in $O(1)$ queries.

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ $\qquad$ Distance $(f, P) < \frac{1}{4}$

(degree 1) $\qquad P(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ Distance $(f, P) < \frac{1}{4}$

(degree 1) $P(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

ABPSS'25: Degree-1 polynomials can be locally corrected using nearly $\tilde{O}(\lg n)$ queries.

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ Distance $(f, P) < \frac{1}{4}$

(degree 1) $p(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

ABPSS'25: Degree-1 polynomials can be locally corrected using nearly $\tilde{O}(\lg n)$ queries.

- Nearly optimal because of a lower bound by [Bafna, Srinivasan, and Sudan' 20].

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ Distance$(f, P) < \frac{1}{2^{d+1}}$

(degree d) $P(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

ABPSS'25: Degree-d polynomials can be locally corrected using nearly $\tilde{O}(\lg n)^d$ queries.

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ Distance $(f, P) < \frac{1}{2^{d+1}}$

(degree d) $P(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

ABPSS'25: Degree-d polynomials can be locally corrected using nearly $\tilde{O}(\lg n)^d$ queries.

We also have combinatorial list decoding bound and sublinear query local list correction algorithm.

Our Results

$f(x) : \{0,1\}^n \longrightarrow \mathbb{R}$ Distance $(f, P) < \frac{1}{2^{d+1}}$

(degree d) $p(x) : \{0,1\}^n \longrightarrow \mathbb{R}$

ABPSS'25: Degree-d polynomials can be locally corrected using nearly $\tilde{O}(\lg n)^d$ queries.

We also have combinatorial list decoding bound and sublinear query local list correction algorithm.

THANK YOU !!

Questions?
Comments?