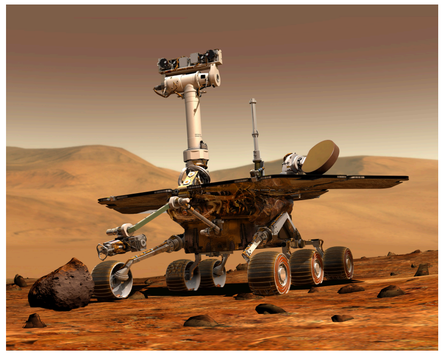
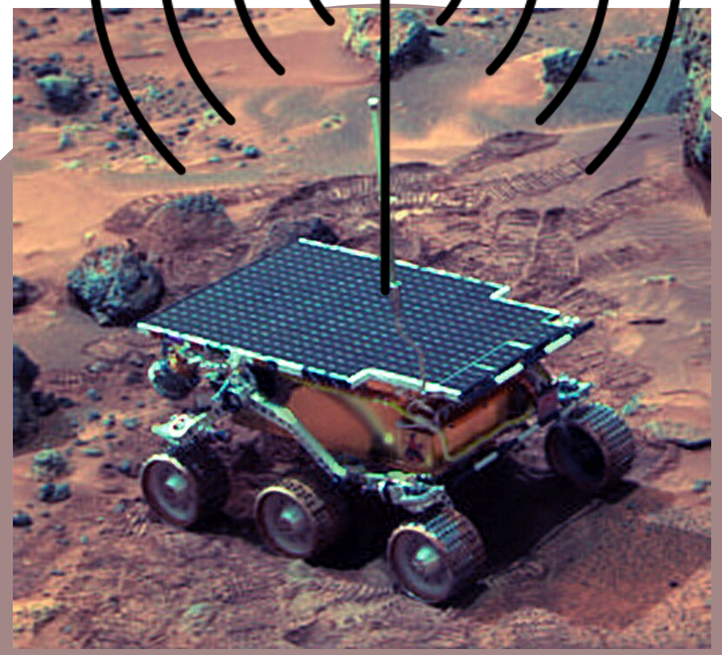
Signal recovery from very noisy codes

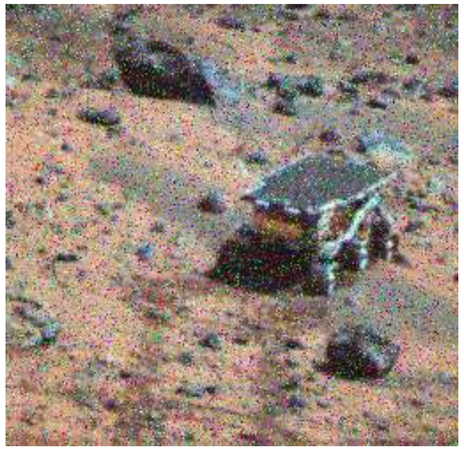


Anqi Li
University of Cambridge



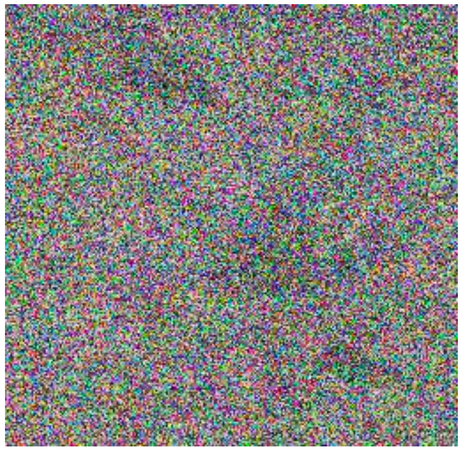
Sojourner (Mars rover)

0 %

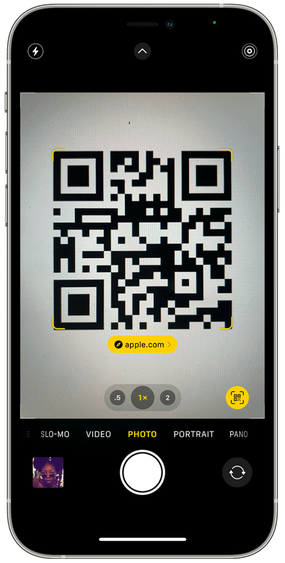


5 %

20 %

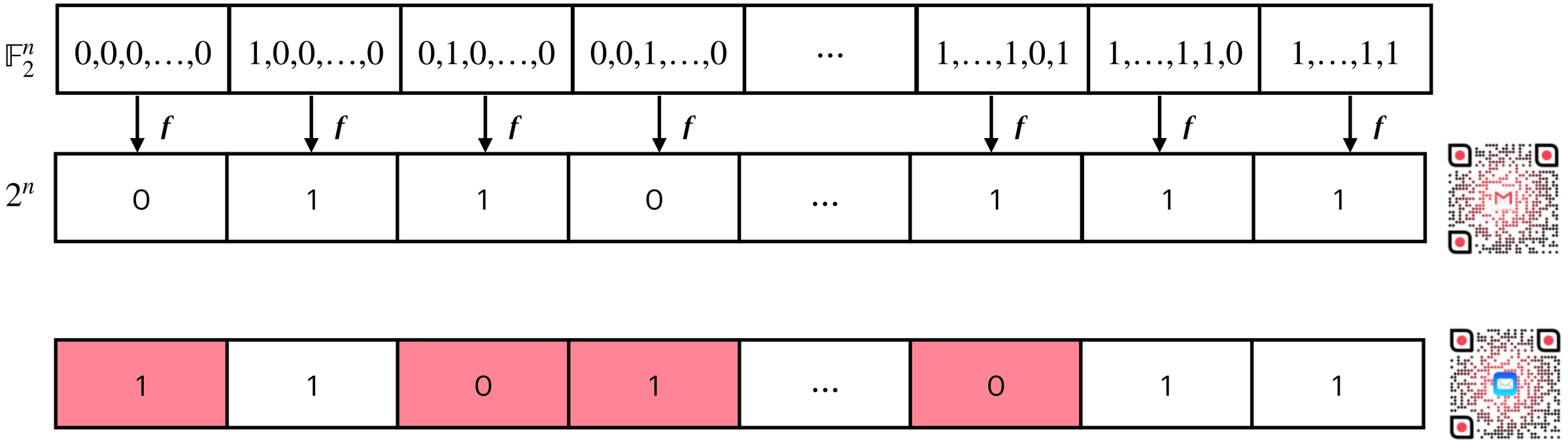


40 %


apple.com
.5
1×
2
SLO-MO
VIDEO
PHOTO
PORTRAIT
PANO

Error-correcting Reed-Muller codes

encode polynomial $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$ in terms of its outputs



Reed-Muller codes

Uncorrupted

0	1	1	0	…	1	1	1

Corrupted

1	1	0	1	…	0	1	1

- If 1% of the bits are corrupted, can we efficiently recover the original bits?
- If 70% of the bits are corrupted, can we ~~efficiently~~ recover the original bits?
- If 49% of the bits are corrupted, can we efficiently recover the original bits?

ish

List decoding

When there's lots of noise

- **Input**: $(50 - \varepsilon)$ % corrupted output of degree d polynomial
- **Given**: query access to the corrupted 2^n bit string
- **Goal**: Output a list of candidate polynomials (with high probability) using polynomial in n number of queries

List decoding

When there's lots of noise

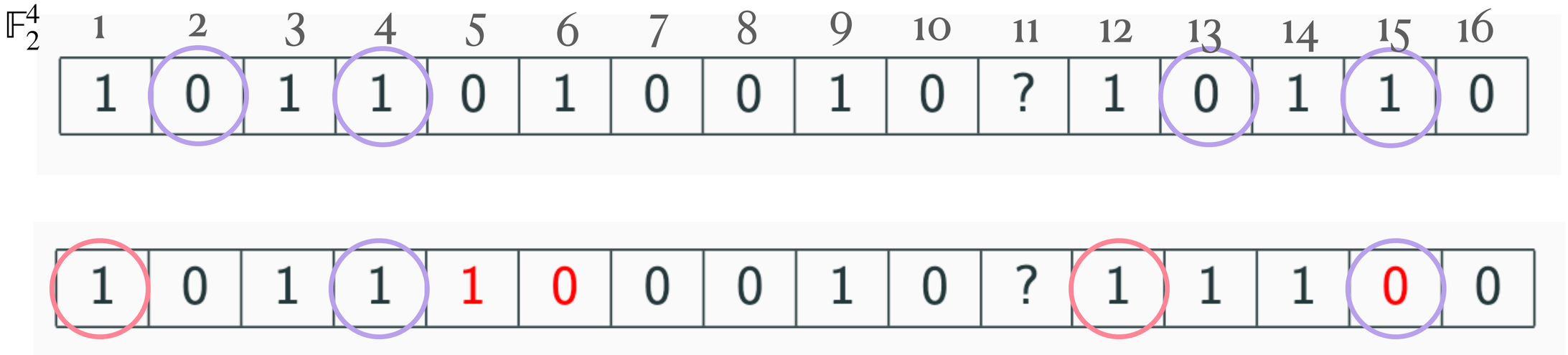
- **Input**: $(50 - \varepsilon)$ % corrupted output of degree d polynomial
- **Goal**: Output a list of possible starting polynomials efficiently

History

- $d = 1$: Goldreich-Levin (1989); applications in *cryptography, learning theory, ...*
- $d = 2$: Quadratic Goldreich-Levin (2011 - Tulsiani, Wolf)
- $d = 3$: **Cubic Goldreich-Levin** (2023 - Kim, L., Tidor)

Scenario with less noise

- **Input**: 20 % corrupted output of linear function
- **Given:** query access to the corrupted 2^n bit string
- **Goal:** Output the actual value of the i-th bit with high probability efficiently



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
g	1	0	1	1	1	0	0	0	1	0	?	1	1	1	0	0

- $g(11) = g(11+b) - g(b)$ for $\geq 60\,\%$ of $b \in \{1,\ldots,16\}$
- When there is 49% noise, this no longer decodes uniquely
  Differencing picks out linear functions ⇒ (discrete) Fourier analysis
- Large Fourier coefficients = candidate linear functions
- Parseval's theorem ⇒ there are few such candidate functions
- For polynomials, "successive differencing" ⇒ higher order Fourier analysis

Codes from polynomials are very resistant to corruption by noise.

Overview of other projects

extremal & additive combinatorics ∩ theory CS

discrete Fourier analysis ∩ probabilistic methods