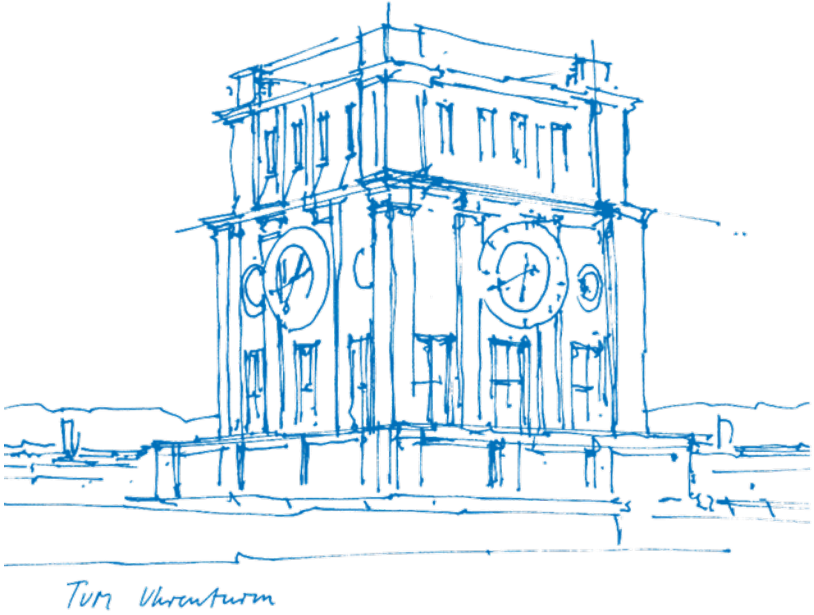
Decoding Insertions/Deletions via List Recovery

Anisha Banerjee[1], Roni Con[2], Antonia Wachter-Zeh[1] and Eitan Yaakobi[2]

[1] Technical University of Munich
Institute for Communications Engineering

[2]Technion – Israel Institute of Technology
Department of Computer Science

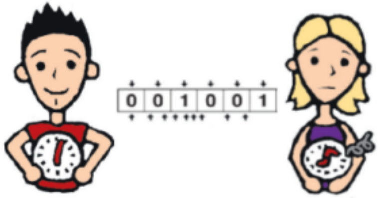
June 25, 2025



Outline

Motivation

- Insertions & deletions occur
  - due to improper synchronization

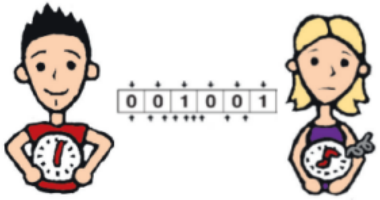


G. M. Church *et al.*, “Next-generation digital information storage in dna,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012

N. Goldman *et al.*, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013

Motivation

- Insertions & deletions occur
  - due to improper synchronization
  - in molecular storage paradigms, e.g. DNA data storage

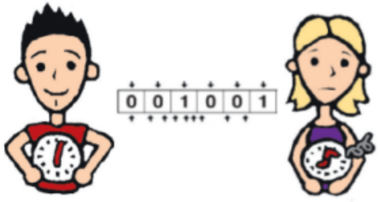


G. M. Church *et al.*, “Next-generation digital information storage in dna,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012

N. Goldman *et al.*, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013

Motivation

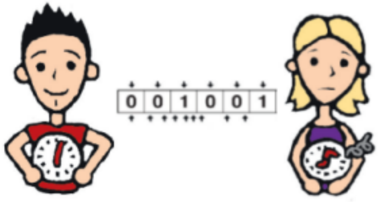
- Insertions & deletions occur
  - due to improper synchronization
  - in molecular storage paradigms, e.g. DNA data storage
- Hard to perform decoding efficiently.



G. M. Church *et al.*, “Next-generation digital information storage in dna,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012
N. Goldman *et al.*, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013

Motivation

- Insertions & deletions occur
  - due to improper synchronization
  - in molecular storage paradigms, e.g. DNA data storage
- Hard to perform decoding efficiently.
- *Trick:* reduce the insdel-decoding problem to *list recovery*!



G. M. Church *et al.*, “Next-generation digital information storage in dna,” *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012
N. Goldman *et al.*, “Towards practical, high-capacity, low-maintenance information storage in synthesized dna,” *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013

What is List Recovery?

- Consider $\mathcal{C} \subseteq \mathbb{F}_q^n$ that corrects t substitutions.
- Assume integers $t' > t$ and $L, \ell > 1$.

V. Guruswami, "Algorithmic results in list decoding," *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006

What is List Recovery?

- Consider $\mathcal{C} \subseteq \mathbb{F}_q^n$ that corrects t substitutions.
- Assume integers $t' > t$ and $L, \ell > 1$.
- Unique Decoding:
  Returns $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t$.

Decoder input

$$\boldsymbol{y} = \boxed{1 \mid 3 \mid \cdots\cdots \mid 0} \in \mathbb{F}_{q=4}^n$$

V. Guruswami, "Algorithmic results in list decoding," *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006

What is List Recovery?

- Consider $\mathcal{C} \subseteq \mathbb{F}_q^n$ that corrects t substitutions.
- Assume integers $t' > t$ and $L, \ell > 1$.
- Unique Decoding:
  Returns $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t$.
- List Decoding:
  Returns $\leq L$ codewords $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t'$.

Decoder input

$$\boldsymbol{y} = \boxed{1 \; 3 \; \cdots\cdots \; 0} \in \mathbb{F}_{q=4}^n$$

V. Guruswami, "Algorithmic results in list decoding," *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006

What is List Recovery?

- Consider $\mathcal{C} \subseteq \mathbb{F}_q^n$ that corrects t substitutions.
- Assume integers $t' > t$ and $L, \ell > 1$.
- Unique Decoding:
  Returns $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t$.
- List Decoding:
  Returns $\leq L$ codewords $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t'$.
- List Recovery:
  Returns $\leq L$ codewords $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \notin S_i\}| \leq t'$.

Decoder input

$$\boldsymbol{y} = \boxed{1 \mid 3 \mid \cdots\cdots \mid 0} \in \mathbb{F}_{q=4}^n$$

$$S = \boxed{\{1,0\} \mid \{2,3\} \mid \cdots\cdots \mid \{0,2\}} \in \binom{\mathbb{F}_{q=4}}{\ell=2}^n$$

V. Guruswami, “Algorithmic results in list decoding,” *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006

What is List Recovery?

- Consider $\mathcal{C} \subseteq \mathbb{F}_q^n$ that corrects t substitutions.
- Assume integers $t' > t$ and $L, \ell > 1$.
- Unique Decoding:
  Returns $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t$.
- List Decoding:
  Returns $\leq L$ codewords $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \neq y_i\}| \leq t'$.
- List Recovery:
  Returns $\leq L$ codewords $\hat{\boldsymbol{c}} \in \mathcal{C}$ s.t. $|\{i \in [n] : \hat{c}_i \notin S_i\}| \leq t'$.

Decoder input

$$\boldsymbol{y} = \boxed{1 \mid 3 \mid \cdots\cdots \mid 0} \in \mathbb{F}_{q=4}^n$$

$$S = \boxed{\{1,0\} \mid \{2,3\} \mid \cdots\cdots \mid \{0,2\}} \in \binom{\mathbb{F}_{q=4}}{\ell=2}^n$$

V. Guruswami, “Algorithmic results in list decoding,” *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006

List-Recoverable Codes

Definition (List-Decodable Codes)

For $\rho \in [0, 1]$ and integer L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, L)-*list-decodable* if for any $\boldsymbol{y} \in \mathbb{F}_q^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \neq y_i\}| \leq \rho n$.

List-Recoverable Codes

Definition (List-Decodable Codes)

For $\rho \in [0, 1]$ and integer L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, L)-*list-decodable* if for any $\boldsymbol{y} \in \mathbb{F}_q^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \neq y_i\}| \leq \rho n$.

Definition (List-Recoverable Codes)

For $\rho \in [0, 1]$ and integers ℓ, L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, ℓ, L)-*list-recoverable* if for any $S \in \binom{\mathbb{F}_q}{\ell}^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \notin S_i\}| \leq \rho n$.

List-Recoverable Codes

Definition (List-Decodable Codes)

For $\rho \in [0,1]$ and integer L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, L)-*list-decodable* if for any $\boldsymbol{y} \in \mathbb{F}_q^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \neq y_i\}| \leq \rho n$.

Definition (List-Recoverable Codes)

For $\rho \in [0,1]$ and integers ℓ, L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, ℓ, L)-*list-recoverable* if for any $S \in \binom{\mathbb{F}_q}{\ell}^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \notin S_i\}| \leq \rho n$.

- A (ρ, L)-list-decodable code $\equiv$ a $(\rho, \ell = 1, L)$-list-recoverable code.

List-Recoverable Codes

Definition (List-Decodable Codes)

For $\rho \in [0, 1]$ and integer L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, L)-*list-decodable* if for any $\boldsymbol{y} \in \mathbb{F}_q^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \neq y_i\}| \leq \rho n$.

Definition (List-Recoverable Codes)

For $\rho \in [0, 1]$ and integers ℓ, L, a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is (ρ, ℓ, L)-*list-recoverable* if for any $S \in \binom{\mathbb{F}_q}{\ell}^n$ there are $\leq L$ codewords $\boldsymbol{c} \in \mathcal{C}$ that satisfy $|\{i \in [n] : c_i \notin S_i\}| \leq \rho n$.

- A (ρ, L)-list-decodable code $\equiv$ a $(\rho, \ell = 1, L)$-list-recoverable code.
- Example: RS codes are efficiently list-recoverable by the Guruswami-Sudan (GS) decoder.

Prior Work

- Efficient insdel decoder with synchronization symbols [Haeupler and Shahrasbi '17]

Prior Work

- Efficient insdel decoder with synchronization symbols [Haeupler and Shahrasbi '17]
- Reed-Solomon codes for insdels
  - [Safavi-Naini and Wang '02], [Wang *et al.* '04], [Tonien and Safavi-Naini '07], [Duc *et al.* '21], [Liu and Tjuawinata '21], [Con *et al.* '23], [Liu '24], [Con *et al.* '24]

Prior Work

- Efficient insdel decoder with synchronization symbols [Haeupler and Shahrasbi '17]
- Reed-Solomon codes for insdels
  - [Safavi-Naini and Wang '02], [Wang *et al.* '04], [Tonien and Safavi-Naini '07], [Duc *et al.* '21], [Liu and Tjuawinata '21], [Con *et al.* '23], [Liu '24], [Con *et al.* '24]
- Only efficient non-trivial insdel decoder for $[n, k = 2]_q$ RS codes [Singhvi '24]
  - for $[n, k = 2]_q$ RS codes for insdels in [Con *et al.* '24]
  - corrects $n - 3$ deletions in linear time.

Overview of Results

- $d_{\mathrm{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Overview of Results

- $d_{\text{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Definition (List-Decodable-Insdel Codes)

For $\rho \in [0, 1]$ and integer $L \geq 1$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, L)-*list-decodable-insdel code* if for any $\boldsymbol{y} \in \mathbb{F}_q^m$ it holds that $|\{\boldsymbol{c} \in \mathcal{C} : d_{\text{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}| \leq L$.

Overview of Results

- $d_{\mathrm{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Definition (List-Decodable-Insdel Codes)

For $\rho \in [0, 1]$ and integer $L \geq 1$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, L)-*list-decodable-insdel code* if for any $\boldsymbol{y} \in \mathbb{F}_q^m$ it holds that $|\{\boldsymbol{c} \in \mathcal{C} : d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}| \leq L$.

Our Results

- Any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list-decodable-insdel code.

Overview of Results

- $d_{\text{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Definition (List-Decodable-Insdel Codes)

For $\rho \in [0, 1]$ and integer $L \geq 1$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, L)-*list-decodable-insdel code* if for any $\boldsymbol{y} \in \mathbb{F}_q^m$ it holds that $|\{\boldsymbol{c} \in \mathcal{C} : d_{\text{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}| \leq L$.

Our Results

- Any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list-decodable-insdel code.
  - First efficient insdel-list-decoder for $[n, k > 2]$ RS codes!

Overview of Results

- $d_{\text{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Definition (List-Decodable-Insdel Codes)

For $\rho \in [0, 1]$ and integer $L \geq 1$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, L)-*list-decodable-insdel code* if for any $\boldsymbol{y} \in \mathbb{F}_q^m$ it holds that $|\{\boldsymbol{c} \in \mathcal{C} : d_{\text{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}| \leq L$.

Our Results

- Any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list-decodable-insdel code.
  - First efficient insdel-list-decoder for $[n, k > 2]$ RS codes!
  - Rate-error tradeoffs for adversarial & probabilistic insdel channels

Overview of Results

- $d_{\mathrm{ed}}(\boldsymbol{x}, \boldsymbol{y}) \leftarrow$ edit distance between $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$

Definition (List-Decodable-Insdel Codes)

For $\rho \in [0, 1]$ and integer $L \geq 1$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, L)-*list-decodable-insdel code* if for any $\boldsymbol{y} \in \mathbb{F}_q^m$ it holds that $|\{\boldsymbol{c} \in \mathcal{C} : d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}| \leq L$.

Our Results

- Any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list-decodable-insdel code.
  - First efficient insdel-list-decoder for $[n, k > 2]$ RS codes!
  - Rate-error tradeoffs for adversarial & probabilistic insdel channels
- Adapted the Koetter-Vardy algorithm for probabilistic insdel channel [Davey and MacKay '01]
  - also for jointly decoding multiple received sequences

Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

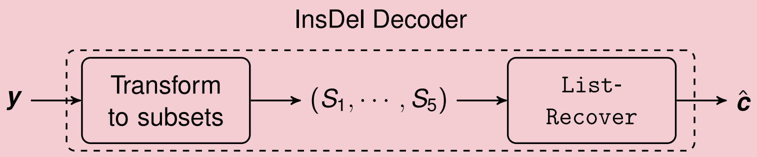
$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$

Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$



How to assign S_i?

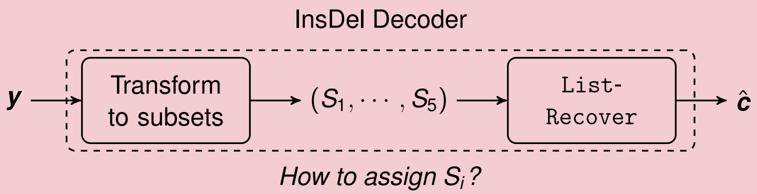
Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$

$c_1 \in \{0, 2\}$



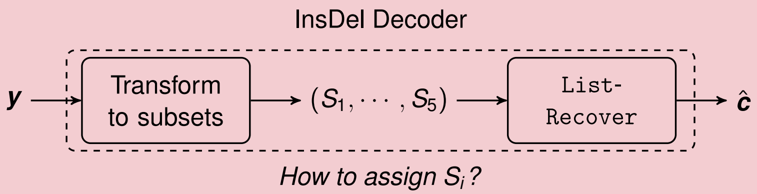
Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$

$$c_1 \in \{0, 2\}$$
$$c_2 \in \{2, 1\}$$



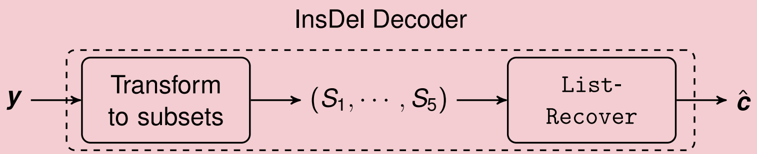
Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$

$$\begin{aligned} c_1 &\in \{0, 2\} \\ c_2 &\in \{2, 1\} \\ &\vdots \\ c_5 &\in \{0, 2\} \end{aligned}$$



How to assign S_i?

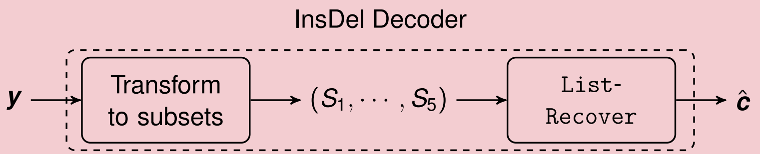
Correct One Insertion via List Recovery

Example

Consider a $(\rho, \ell = 2, L)$-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_{q=3}^{n=5}$.

$$\boldsymbol{c} \in \mathcal{C} \xrightarrow{\text{1 insertion}} \boldsymbol{y} = (0, 2, 1, 1, 0, 2)$$

$$c_1 \in \{0, 2\} \to S_1$$
$$c_2 \in \{2, 1\} \to S_2$$
$$\vdots$$
$$c_5 \in \{0, 2\} \to S_5$$



How to assign S_i?

Decoding Algorithm

For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ
Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

Decoding Algorithm

For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ

Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

for $i \in [n]$ **do**

$\quad S_i \leftarrow \{y_{\max\{1, i-\lfloor \ell/2 \rfloor\}}, \ldots, y_{\min\{m, i+\lfloor \ell/2 \rfloor\}}\}$

Decoding Algorithm

For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ

Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

for $i \in [n]$ **do**

  $S_i \leftarrow \{y_{\max\{1, i-\lfloor \ell/2 \rfloor\}}, \ldots, y_{\min\{m, i+\lfloor \ell/2 \rfloor\}}\}$

Execute `List-Recover` with the lists $S_1, \ldots, S_n$ to get $\mathcal{L} \subseteq \mathcal{C}$.

Decoding Algorithm

For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

---

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ

Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

1 **for** $i \in [n]$ **do**

2 $\quad S_i \leftarrow \{y_{\max\{1, i - \lfloor \ell/2 \rfloor\}}, \ldots, y_{\min\{m, i + \lfloor \ell/2 \rfloor\}}\}$

3 Execute `List-Recover` with the lists $S_1, \ldots, S_n$ to get $\mathcal{L} \subseteq \mathcal{C}$.

4 return $\{\boldsymbol{c} \in \mathcal{L} | d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}$.

Decoding Adversarial Insdels

Theorem

Let $\rho \in [0, 1]$ and set $\ell = 2\rho n + 1$. Assume that $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, ℓ, L)-list-recoverable code with an algorithm `List-Recover` that runs in time T.

Decoding Adversarial Insdels

Theorem

Let $\rho \in [0,1]$ and set $\ell = 2\rho n + 1$. Assume that $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, ℓ, L)-list-recoverable code with an algorithm `List-Recover` that runs in time T.

1. Then, running alg. `Decode` on input $\boldsymbol{y} \in \mathbb{F}_q^m$ returns a list $\mathcal{L}' \subseteq \mathcal{C}$ such that
   - for every $\boldsymbol{c} \in \mathcal{L}'$, $d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n$, and
   - $|\mathcal{L}'| \leq L$.

Decoding Adversarial Insdels

Theorem

Let $\rho \in [0,1]$ and set $\ell = 2\rho n + 1$. Assume that $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, ℓ, L)-list-recoverable code with an algorithm `List-Recover` that runs in time T.

1. Then, running alg. `Decode` on input $\boldsymbol{y} \in \mathbb{F}_q^m$ returns a list $\mathcal{L}' \subseteq \mathcal{C}$ such that
   - for every $\boldsymbol{c} \in \mathcal{L}'$, $d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n$, and
   - $|\mathcal{L}'| \leq L$.
2. The running time of Alg. `Decode` is $O(T + L \cdot n^2)$.

Decoding Adversarial Insdels

Theorem

Let $\rho \in [0,1]$ and set $\ell = 2\rho n + 1$. Assume that $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a (ρ, ℓ, L)-list-recoverable code with an algorithm `List-Recover` that runs in time T.

1. Then, running alg. `Decode` on input $\boldsymbol{y} \in \mathbb{F}_q^m$ returns a list $\mathcal{L}' \subseteq \mathcal{C}$ such that
   - for every $\boldsymbol{c} \in \mathcal{L}'$, $d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n$, and
   - $|\mathcal{L}'| \leq L$.
2. The running time of Alg. `Decode` is $O(T + L \cdot n^2)$.
3. Moreover, if $\rho n \leq \lfloor \frac{d_{\mathrm{ed}}(\mathcal{C})-1}{2} \rfloor$, then $|\mathcal{L}'| \leq 1$.

Reed-Solomon Codes

- Efficiently list-recoverable!

Definition

Let $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ be distinct. An $[n, k]_q$ *Reed-Solomon (RS) code* is defined as

$$\mathcal{RS}(n, k)_q = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}.$$

Reed-Solomon Codes

- Efficiently list-recoverable!

Definition

Let $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ be distinct. An $[n,k]_q$ *Reed-Solomon (RS) code* is defined as

$$\mathcal{RS}(n,k)_q = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}.$$

Corollary (Guruswami-Sudan Decoder)

Let $\varepsilon > 0$ and $\mathcal{C}$ be an $[n,k]_q$ RS code that corrects from t insdels where

$$t \leq n - \sqrt{(1+\varepsilon) \cdot kn \cdot (2t+1)}\,. \tag{1}$$

Then, $\mathcal{C}$ has a deterministic unique-decoding algorithm that corrects t insdels in time $O(n^3\varepsilon^{-6})$.

Reed-Solomon Codes

- Efficiently list-recoverable!

Definition

Let $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ be distinct. An $[n, k]_q$ *Reed-Solomon (RS) code* is defined as

$$\mathcal{RS}(n, k)_q = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}.$$

Corollary (Guruswami-Sudan Decoder)

Let $\varepsilon > 0$ and $\mathcal{C}$ be an $[n, k]_q$ RS code that corrects from t insdels where

$$t \leq n - \sqrt{(1+\varepsilon) \cdot kn \cdot (2t+1)}\ . \tag{1}$$

Then, $\mathcal{C}$ has a deterministic unique-decoding algorithm that corrects t insdels in time $O(n^3\varepsilon^{-6})$.

- ***Note***: (1) requires $k \cdot t = O(n)$.

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_{\mathrm{d}} \in (0, 1)$.

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_\mathrm{d} \in (0, 1)$.
- How to assign the subsets S_i for $\texttt{List-Recover}$?

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_{\mathrm{d}} \in (0, 1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_\mathrm{d} \in (0,1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?
  - *Hint*: i symbols transmitted $\rightarrow (1-P_\mathrm{d})i$ symbols received on average.

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_\mathrm{d} \in (0,1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?
  - *Hint*: i symbols transmitted $\rightarrow (1 - P_\mathrm{d})i$ symbols received on average.
- For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_{\mathrm{d}} \in (0,1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?
  - *Hint*: i symbols transmitted $\rightarrow (1 - P_{\mathrm{d}})i$ symbols received on average.
- For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ, n
Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_\mathrm{d} \in (0,1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?
  - *Hint*: i symbols transmitted $\rightarrow (1-P_\mathrm{d})i$ symbols received on average.
- For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ, n
Output: Codewords $\boldsymbol{c} \in \mathcal{C}$

```
for i ∈ [n] do
    S_i ← {y_{max{1,(1−P_d)i−ℓ/2}}, ..., y_{min{m,(1−P_d)i+ℓ/2}}}
```

$$\textbf{for } i \in [n] \textbf{ do}$$
$$S_i \leftarrow \{y_{\max\{1,(1-P_\mathrm{d})i-\ell/2\}}, \ldots, y_{\min\{m,(1-P_\mathrm{d})i+\ell/2\}}\}$$

Probablistic Deletion Channel

- Deletes each transmitted symbol independently with probability $P_{\mathrm{d}} \in (0,1)$.
- How to assign the subsets S_i for `List-Recover`?
  - I.e., if $\boldsymbol{c}$ is transmitted, in which size-ℓ window of $\boldsymbol{y}$ is c_i most likely to appear?
  - *Hint*: i symbols transmitted $\rightarrow$ $(1-P_{\mathrm{d}})i$ symbols received on average.
- For a (ρ, ℓ, L)-list-recoverable code $\mathcal{C} \subseteq \mathbb{F}_q^n$,

Algorithm: `Decode`

Input: $(y_1, \ldots, y_m)$, ℓ, n
Output: Codewords $\boldsymbol{c} \in \mathcal{C}$
1 **for** $i \in [n]$ **do**
2 $\quad S_i \leftarrow \{y_{\max\{1,(1-P_{\mathrm{d}})i-\ell/2}, \ldots, y_{\min\{m,(1-P_{\mathrm{d}})i+\ell/2}\}$
3 Execute `List-Recover` with the lists $S_1, \ldots, S_n$ to get $\mathcal{L} \subseteq \mathcal{C}$.
4 return $\{\boldsymbol{c} \in \mathcal{L} | d_{\mathrm{ed}}(\boldsymbol{c}, \boldsymbol{y}) \leq \rho n\}$.

Probabilistic Deletions

Theorem

Let $P_{\mathrm{d}} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_{\mathrm{d}} + \varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.

Probabilistic Deletions

Theorem

Let $P_{\mathrm{d}} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_{\mathrm{d}}+\varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.
Say the transmission of $\boldsymbol{c} \in \mathcal{C}$ over a channel with deletion probability P_{d} produces $\boldsymbol{y} \in \mathbb{F}_q^m$.

Probabilistic Deletions

Theorem

Let $P_\mathrm{d} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_\mathrm{d} + \varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.
Say the transmission of $\boldsymbol{c} \in \mathcal{C}$ over a channel with deletion probability P_d produces $\boldsymbol{y} \in \mathbb{F}_q^m$.

1. Then, with probability $\exp(-\Omega(n^{0.002}))$, alg. `Decode` produces a list $\mathcal{L}$ such that $\boldsymbol{c} \in \mathcal{L}$.

Probabilistic Deletions

Theorem

Let $P_{\mathrm{d}} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_{\mathrm{d}} + \varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.
Say the transmission of $\boldsymbol{c} \in \mathcal{C}$ over a channel with deletion probability P_{d} produces $\boldsymbol{y} \in \mathbb{F}_q^m$.

1. Then, with probability $\exp(-\Omega(n^{0.002}))$, alg. `Decode` produces a list $\mathcal{L}$ such that $\boldsymbol{c} \in \mathcal{L}$.
2. Moreover, if $(P_{\mathrm{d}} + \varepsilon)n \leq \lfloor \frac{d_{\mathrm{ed}}(\mathcal{C})-1}{2} \rfloor$, then $|\mathcal{L}| = 1$.

Probabilistic Deletions

Theorem

Let $P_{\mathrm{d}} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_{\mathrm{d}}+\varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.
Say the transmission of $\boldsymbol{c} \in \mathcal{C}$ over a channel with deletion probability P_{d} produces $\boldsymbol{y} \in \mathbb{F}_q^m$.

1. Then, with probability $\exp(-\Omega(n^{0.002}))$, alg. `Decode` produces a list $\mathcal{L}$ such that $\boldsymbol{c} \in \mathcal{L}$.
2. Moreover, if $(P_{\mathrm{d}}+\varepsilon)n \leq \lfloor \frac{d_{\mathrm{ed}}(\mathcal{C})-1}{2} \rfloor$, then $|\mathcal{L}| = 1$.

- Unique, efficient decoding (with high probability) for any $P_{\mathrm{d}} \in (0,1)$, needs $k = O(n^{1/2-0.001})$.
  - Better rate-error tradeoff than the adversarial setting: $k \cdot t = O(n)$

Probabilistic Deletions

Theorem

Let $P_{\mathrm{d}} \in (0,1)$, $\varepsilon > 0$ and $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a $(P_{\mathrm{d}} + \varepsilon, n^{1/2+0.001}, L)$-list-recoverable code with an efficient list recoverable algorithm `List-recover`.
Say the transmission of $\boldsymbol{c} \in \mathcal{C}$ over a channel with deletion probability P_{d} produces $\boldsymbol{y} \in \mathbb{F}_q^m$.

1. Then, with probability $\exp(-\Omega(n^{0.002}))$, alg. `Decode` produces a list $\mathcal{L}$ such that $\boldsymbol{c} \in \mathcal{L}$.
2. Moreover, if $(P_{\mathrm{d}} + \varepsilon)n \leq \lfloor \frac{d_{\mathrm{ed}}(\mathcal{C})-1}{2} \rfloor$, then $|\mathcal{L}| = 1$.

- Unique, efficient decoding (with high probability) for any $P_{\mathrm{d}} \in (0,1)$, needs $k = O(n^{1/2-0.001})$.
  - Better rate-error tradeoff than the adversarial setting: $k \cdot t = O(n)$
  - Similar result for a probabilistic insdel channel! [Davey and MacKay '01]

Introduction

Decoding via List Recovery

- Adversarial Insdels
- Probabilistic Insdels

Adapting Koetter-Vardy Algorithm

Conclusion

Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy ’03]

Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0), \ldots, f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,

Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy ’03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\ldots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\ldots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.

Koetter-Vardy Algorithm

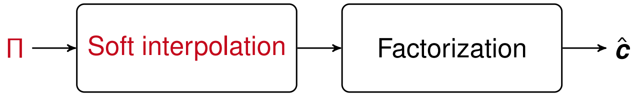
- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\dots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\dots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.
- *Reliability matrix* $\Pi \in (0,1)^{q\times n}$ incorporates soft information on the transmitted symbols.

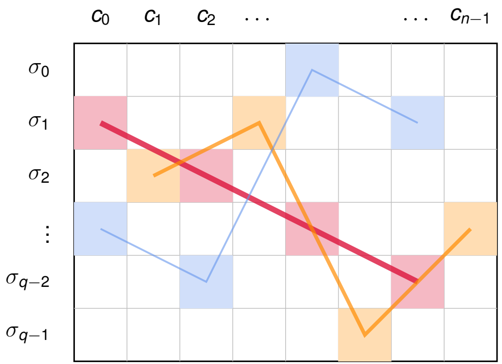
Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\ldots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\ldots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.
- *Reliability matrix* $\Pi \in (0,1)^{q\times n}$ incorporates soft information on the transmitted symbols.
  - If $\mathbb{F}_q = \{\sigma_0,\ldots,\sigma_{q-1}\}$, then for received vector $\boldsymbol{y} \in \mathbb{F}_q^m$, $\pi_{i,j} = \Pr[c_j = \sigma_i | \boldsymbol{y}]$.

Koetter-Vardy Algorithm

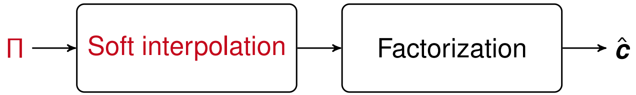
- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\dots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\dots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.
- *Reliability matrix* $\Pi \in (0,1)^{q\times n}$ incorporates soft information on the transmitted symbols.
  - If $\mathbb{F}_q = \{\sigma_0,\dots,\sigma_{q-1}\}$, then for received vector $\boldsymbol{y} \in \mathbb{F}_q^m$, $\pi_{i,j} = \Pr[c_j = \sigma_i | \boldsymbol{y}]$.

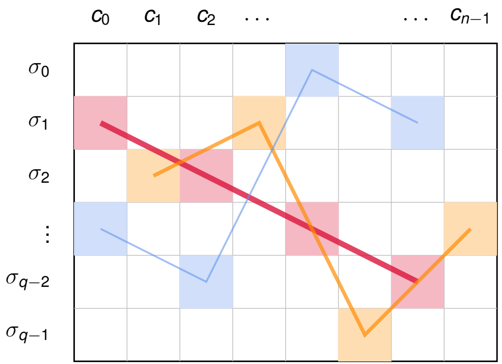




Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\dots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\dots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.
- *Reliability matrix* $\Pi \in (0,1)^{q\times n}$ incorporates soft information on the transmitted symbols.
  - If $\mathbb{F}_q = \{\sigma_0,\dots,\sigma_{q-1}\}$, then for received vector $\boldsymbol{y} \in \mathbb{F}_q^m$, $\pi_{i,j} = \Pr[c_j = \sigma_i | \boldsymbol{y}]$.

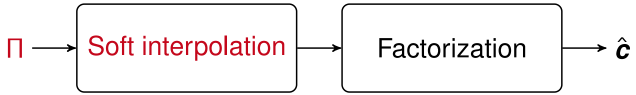


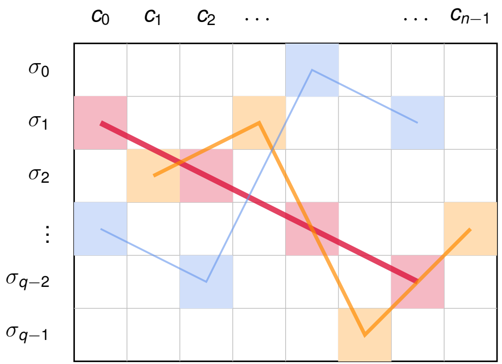


- We compute Π for the Davey-MacKay channel given $\boldsymbol{y}_1,\dots,\boldsymbol{y}_M$!

Koetter-Vardy Algorithm

- Soft-decision decoder for RS codes [Koetter and Vardy '03]
  - Recall: $\mathcal{RS}(n,k)_q = \{(f(\alpha_0),\dots,f(\alpha_{n-1})) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}$,
  - Decoding task: Find best f that fits $(f(\alpha_0),\dots,f(\alpha_{n-1}))$ best to $\boldsymbol{y}$.
- *Reliability matrix* $\Pi \in (0,1)^{q\times n}$ incorporates soft information on the transmitted symbols.
  - If $\mathbb{F}_q = \{\sigma_0,\dots,\sigma_{q-1}\}$, then for received vector $\boldsymbol{y} \in \mathbb{F}_q^m$, $\pi_{i,j} = \Pr[c_j = \sigma_i|\boldsymbol{y}]$.





- We compute Π for the Davey-MacKay channel given $\boldsymbol{y}_1,\dots,\boldsymbol{y}_M$!
  - *Joint decoding*: complexity grows linearly in M.

Davey-MacKay Channel

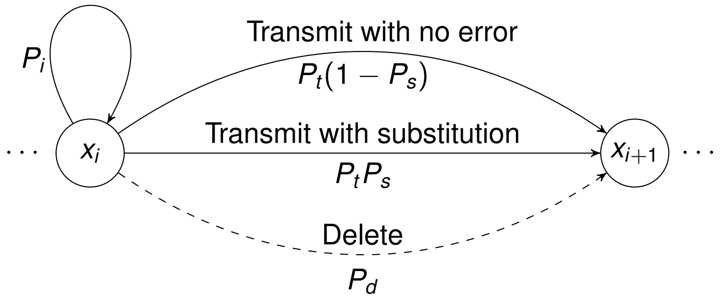
- Induces random insertions, deletions and substitutions [Davey and MacKay ’01]

Davey-MacKay Channel

- Induces random insertions, deletions and substitutions [Davey and MacKay '01]
- Let channel input be $\boldsymbol{x} = (x_1, \ldots, x_T)$.
  - When x_i awaits transmission, four events possible.

Davey-MacKay Channel

- Induces random insertions, deletions and substitutions [Davey and MacKay '01]
- Let channel input be $\boldsymbol{x} = (x_1, \ldots, x_T)$.
  - When x_i awaits transmission, four events possible.

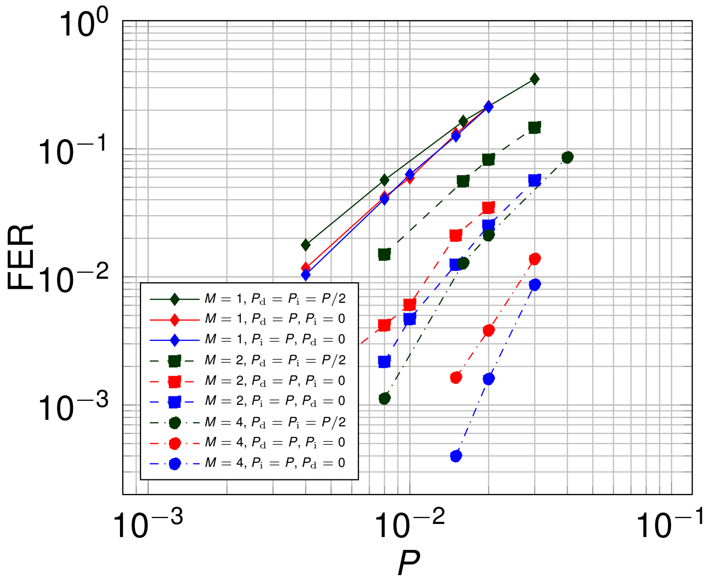


Simulation Results

- Primitive $\mathcal{RS}(100, 33)$ over $\mathbb{F}_{101}$
  - But eval. points randomly permuted. [Beelen *et al.* '25]
- List size set to 5.

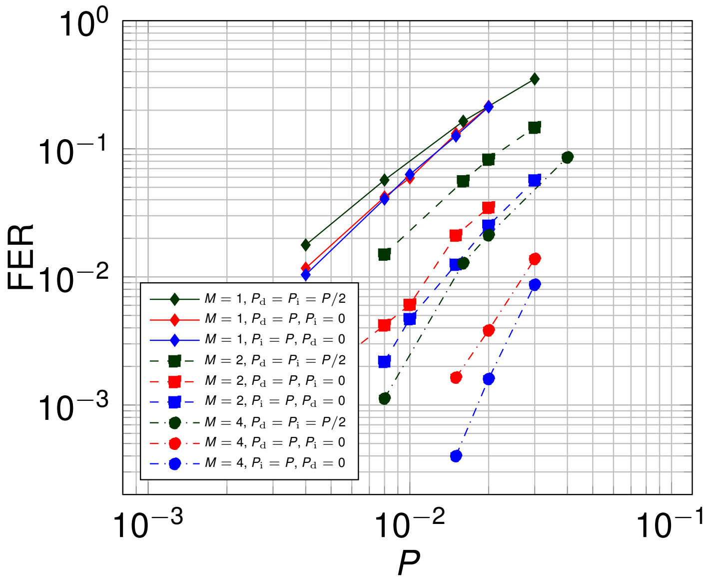
Simulation Results

- Primitive $\mathcal{RS}(100, 33)$ over $\mathbb{F}_{101}$
  - But eval. points randomly permuted. [Beelen *et al.* '25]
- List size set to 5.



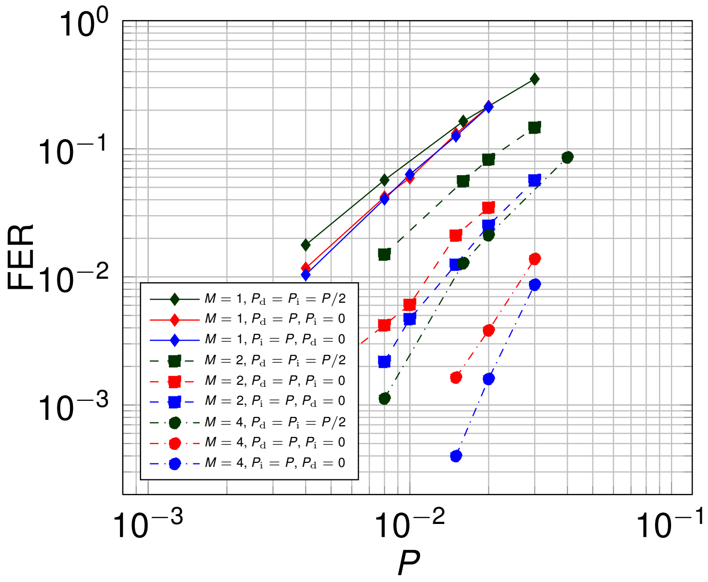
Simulation Results

- Primitive $\mathcal{RS}(100, 33)$ over $\mathbb{F}_{101}$
  - But eval. points randomly permuted. [Beelen *et al.* '25]
- List size set to 5.
- Observations:
  - Deletions harder to correct than insertions.



Simulation Results

- Primitive $\mathcal{RS}(100, 33)$ over $\mathbb{F}_{101}$
  - But eval. points randomly permuted. [Beelen *et al.* '25]
- List size set to 5.
- Observations:
  - Deletions harder to correct than insertions.
  - As M increases, FER decreases.



Conclusion

Results

- Showed that any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list decodable-insdel code.
  - First efficient insdel decoder for $[n, k]$ RS codes for $k > 2$!
  - Better rate-error tradeoff for the probabilistic model.
- Adapted the Koetter-Vardy algorithm for insdels for $M \geq 1$ received sequences

Conclusion

Results

- Showed that any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list decodable-insdel code.
  - First efficient insdel decoder for $[n, k]$ RS codes for $k > 2$!
  - Better rate-error tradeoff for the probabilistic model.
- Adapted the Koetter-Vardy algorithm for insdels for $M \geq 1$ received sequences

Future work

- Alternant codes, folded RS codes..

Conclusion

Results

- Showed that any $(\rho, 2\rho n + 1, L)$-list-recoverable code $\mathcal{C}$ is a (ρ, L)-list decodable-insdel code.
  - First efficient insdel decoder for $[n, k]$ RS codes for $k > 2$!
  - Better rate-error tradeoff for the probabilistic model.
- Adapted the Koetter-Vardy algorithm for insdels for $M \geq 1$ received sequences

Future work

- Alternant codes, folded RS codes..

Thank you!

DNA DATA APPLICATIONS
European Innovation Council
Funded by the European Union

References I

[1] G. M. Church, Y. Gao, and S. Kosuri, "Next-generation digital information storage in dna," *Science*, vol. 337, no. 6102, pp. 1628–1628, Sep. 2012.

[2] N. Goldman *et al.*, "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," *Nature*, vol. 494, no. 7435, pp. 77–80, Feb. 2013.

[3] V. Guruswami, "Algorithmic results in list decoding," *Foundations and Trends® in Theoretical Computer Science*, vol. 2, no. 2, pp. 107–195, 2006.

[4] B. Haeupler and A. Shahrasbi, "Synchronization strings: Codes for insertions and deletions approaching the Singleton bound," in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, ACM, 2017, pp. 33–46.

[5] R. Safavi-Naini and Y. Wang, "Traitor tracing for shortened and corrupted fingerprints," in *ACM workshop on Digital Rights Management*, Springer, 2002, pp. 81–100.

[6] Y. Wang, L. McAven, and R. Safavi-Naini, "Deletion correcting using generalized Reed-Solomon codes," in *Coding, Cryptography and Combinatorics*, Springer, 2004, pp. 345–358.

References II

[7] D. Tonien and R. Safavi-Naini, "Construction of deletion correcting codes using generalized Reed–Solomon codes and their subcodes," *Designs, Codes and Cryptography*, vol. 42, no. 2, pp. 227–237, 2007.

[8] T. D. Duc, S. Liu, I. Tjuawinata, and C. Xing, "Explicit constructions of two-dimensional Reed-Solomon codes in high insertion and deletion noise regime," *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2808–2820, 2021.

[9] S. Liu and I. Tjuawinata, "On 2-dimensional insertion-deletion Reed-Solomon codes with optimal asymptotic error-correcting capability," *Finite Fields and Their Applications*, vol. 73, p. 101 841, 2021.

[10] R. Con, A. Shpilka, and I. Tamo, "Reed–Solomon codes against adversarial insertions and deletions," *IEEE Transactions on Information Theory*, 2023.

[11] J. Liu, "Optimal RS codes and GRS codes against adversarial insertions and deletions and optimal constructions," *IEEE Transactions on Information Theory*, 2024.

[12] R. Con, Z. Guo, R. Li, and Z. Zhang, "Random reed-solomon codes achieve the half-singleton bound for insertions and deletions over linear-sized alphabets," *arXiv preprint arXiv:2407.07299*, 2024.

[13] S. Singhvi, "Optimally decoding two-dimensional reed-solomon codes up to the half-singleton bound," *arXiv preprint arXiv:2412.20771*, 2024.

References III

[14] R. Con, A. Shpilka, and I. Tamo, "Optimal two-dimensional reed–solomon codes correcting insertions and deletions," *IEEE Transactions on Information Theory*, 2024.

[15] M. Davey and D. MacKay, "Reliable communication over channels with insertions, deletions, and substitutions," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 687–698, Feb. 2001.

[16] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of reed-solomon codes," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[17] P. Beelen, R. Con, A. Gruica, M. Montanucci, and E. Yaakobi, *Reed-solomon codes against insertions and deletions: Full-length and rate-$1/2$ codes*, Jan. 2025. arXiv: 2501.11371 [cs].