Decoding Random Binary Linear Codes in $2^{n/20}$

How 1+1=0 Improves Information Set Decoding

A. Becker, A. Joux, A. May, **A. Meurer**

EUROCRYPT 2012, Cambridge

RUB

The Representation Technique [HGJ10]

How to find a needle N in a haystack H...

- Expand H into larger stack H'
- Expanding H' introduces r many representations $N_1, \ldots, N_r$
- Examine a 1/r – fraction of H' to find one N_i

The Representation Technique [HGJ10]

RUB

How to find a needle N in a haystack H...

- Expand H into larger stack H'
- Expanding H' introduces r many representations $N_1, \dots, N_r$
- Examine a 1/r – fraction of H' to find one N_i

Technicality: Find a way to examine a 1/r – fraction of H' without completely constructing it beforehand

The Representation Technique [HGJ10]

RUB

How to find a needle N in a haystack H...

- Expand H into larger stack H'
- Expanding H' introduces r many representations $N_1, \ldots, N_r$
- Examine a 1/r – fraction of H' to find one N_i

Technicality: Find a way to examine a 1/r – fraction of H' without completely constructing it beforehand

Has been used in [MMT11] to improve Information Set Decoding

The Representation Technique

RUB

Optimizing the Representation Technique [BCJ11]

- r = number of needles
- $|H'|$ = size of expanded haystack
- Ratio $|H'| / r$ determines efficiency

→ **Increase r while keeping $|H'|$ small**

The Representation Technique

Optimizing the Representation Technique [BCJ11]

- r = number of needles
- |H'| = size of expanded haystack
- Ratio |H'| / r determines efficiency

→ **Increase r while keeping |H'| small**

Can we use $1+1=0$ to increase r ?

Recap Binary Linear Codes

RUB

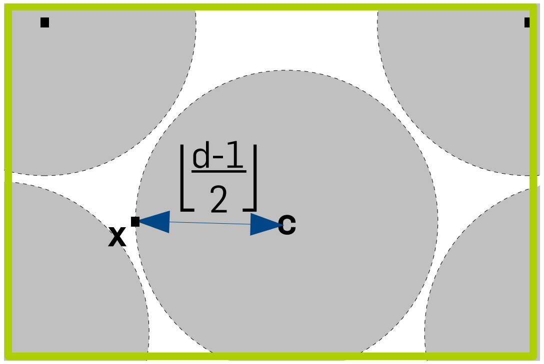
- C = random binary [n,k,d] code
- n = length / k = dimension / d = minimum distance

Bounded Distance Decoding (BDD)

- Given $\mathbf{x} = \mathbf{c}+\mathbf{e}$ with $\mathbf{c} \in C$

  and $w := wt(\mathbf{e}) = \left\lfloor \frac{d-1}{2} \right\rfloor$

- Find $\mathbf{e}$ and thus $\mathbf{c} = \mathbf{x}+\mathbf{e}$



Comparing Running Times

How to compare performance of decoding algorithms

- Running time $T(n,k,d)$
- Fixed code rate $R = k/n$
- For $n \rightarrow \infty$, k and d are related via Gilbert-Varshamov bound, thus

$$T(n,k,d) = T(n,k)$$

- Compare algorithms by complexity coefficient $F(k)$, i.e.

$$T(n,k) = 2^{F(k) \cdot n + o(n)}$$

Comparing Running Times

RUB

Minimize F(k)!

How to compare performanc… s

- Running time $T(n,k,d)$
- Fixed code rate $R = k/n$
- For $n \to \infty$, k and d are related vi… ert-Varshamov bound, thus

$$T(n,k,d) = T \ldots k)$$

- Compare algorithms by compl… xity coefficient $F(k)$, i.e.

$$T(n,k) = 2^{F(k) \cdot n + o(n)}$$

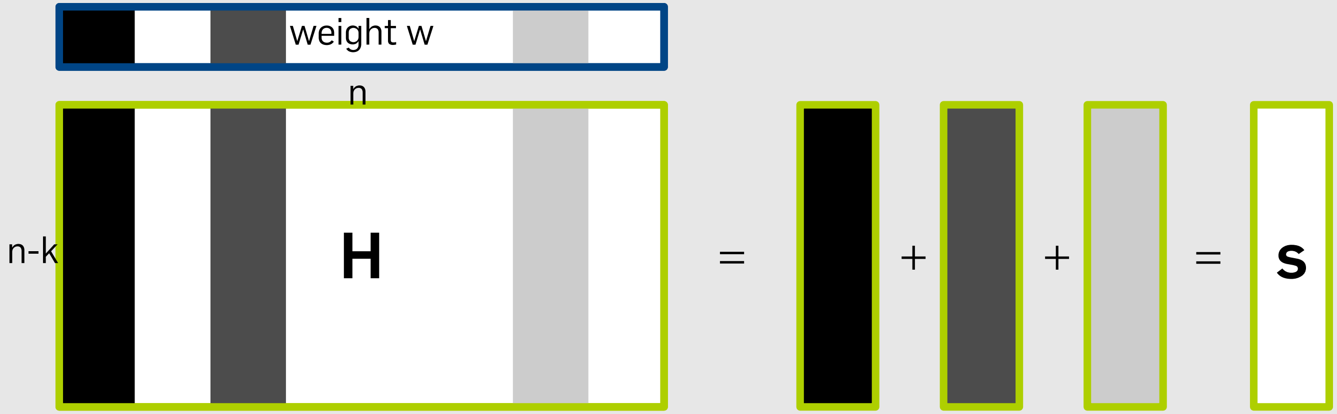
Syndrome Decoding

RUB

(BDD) Given $\mathbf{x} = \mathbf{c}+\mathbf{e}$ with $\mathbf{c} \in C$ and $wt(\mathbf{e})=w$, find $\mathbf{e}$!

- $\mathbf{H}$ = parity check matrix
- Consider syndrome $\mathbf{s} := s(\mathbf{x}) = \mathbf{H} \cdot \mathbf{x} = \mathbf{H} \cdot (\mathbf{c}+\mathbf{e}) = \mathbf{H} \cdot \mathbf{e}$

→ Find linear combination of w columns of **H** matching **s**

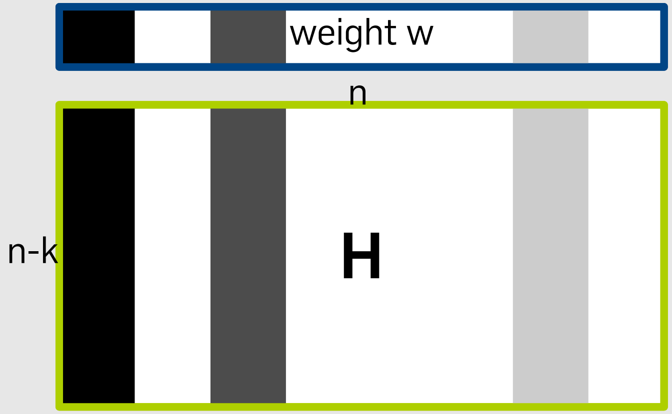


Syndrome Decoding

(BDD) Given $\mathbf{x} = \mathbf{c}+\mathbf{e}$ with $\mathbf{c} \in C$ and $wt(\mathbf{e})=w$, find $\mathbf{e}$!

- $\mathbf{H}$ = parity check matrix
- Consider syndrome $\mathbf{s} := s(\mathbf{x}) = \mathbf{H} \cdot \mathbf{x} = \mathbf{H} \cdot (\mathbf{c}+\mathbf{e}) = \mathbf{H} \cdot \mathbf{e}$

→ Find linear combination of w columns of **H** matching **s**



Brute-Force complexity

$$T(n,k,d) = \binom{n}{w}$$

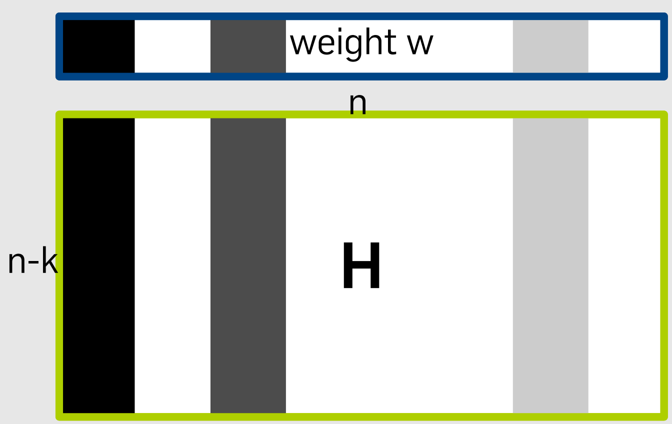
Syndrome Decoding

(BDD) Given $\mathbf{x} = \mathbf{c}+\mathbf{e}$ with $\mathbf{c} \in C$ and $\mathrm{wt}(\mathbf{e})=w$, find $\mathbf{e}$!

- $\mathbf{H}$ = parity check matrix

- Consider syndrome $\mathbf{s} := s(\mathbf{x}) = \mathbf{H}\cdot\mathbf{x} = \mathbf{H}\cdot(\mathbf{c}+\mathbf{e}) = \mathbf{H}\cdot\mathbf{e}$

→ Find linear combination of w columns of **H** matching **s**



$$F(k) \leq 0.3868$$

Some Basic Observations for BDD

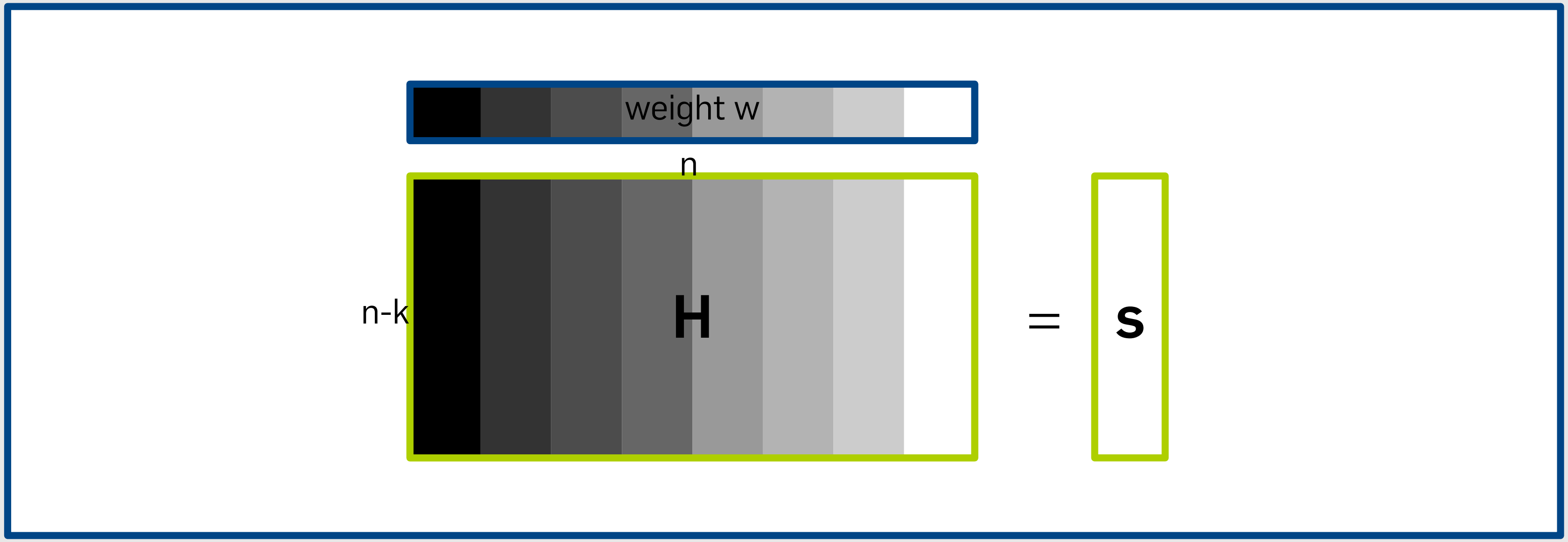
Allowed (linear algebra) transformations

- Permuting the columns of $\mathbf{H}$ does not change the problem

Some Basic Observations for BDD

Allowed (linear algebra) transformations

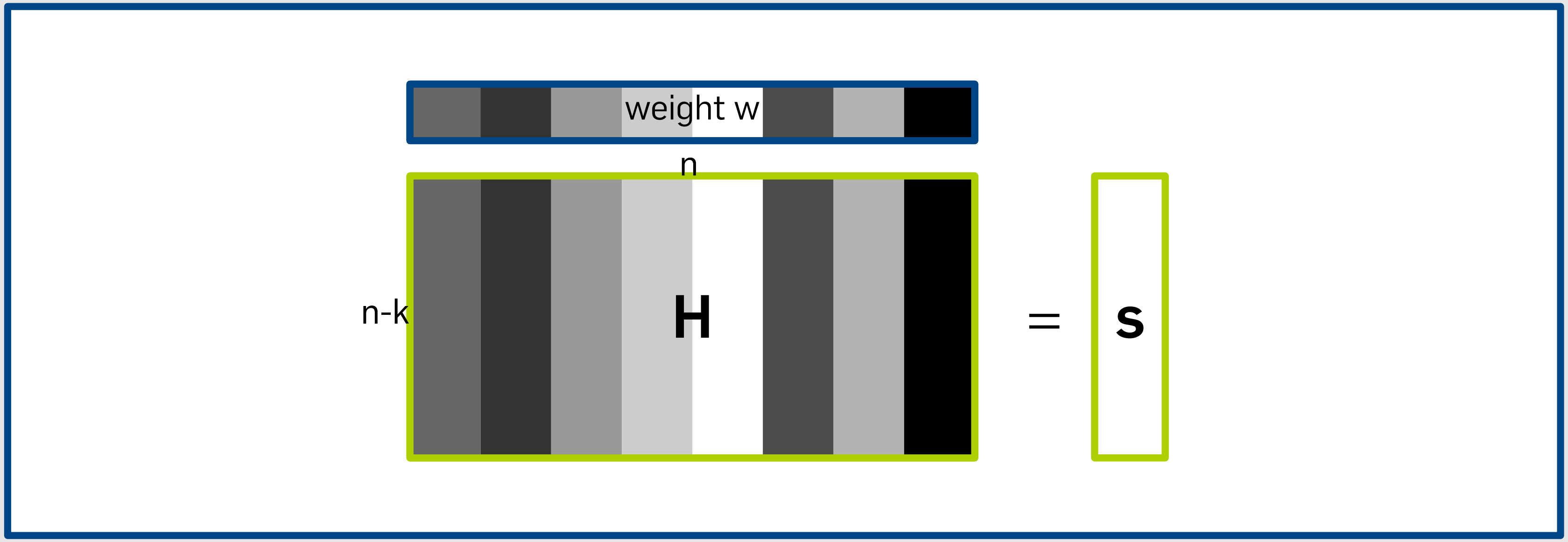
- Permuting the columns of **H** does not change the problem



Some Basic Observations for BDD

Allowed (linear algebra) transformations

- Permuting the columns of **H** does not change the problem



Some Basic Observations for BDD

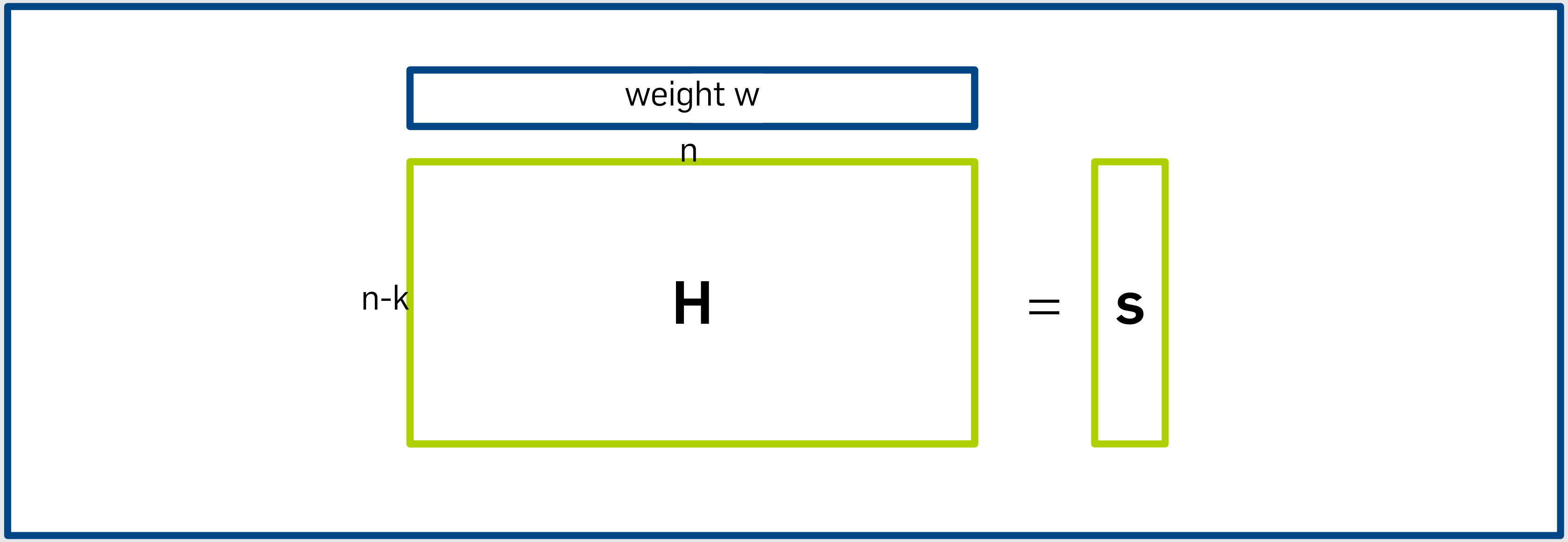
Allowed (linear algebra) transformations

- Permuting the columns of **H** does not change the problem
- Elementary row operations on **H** do not change the problem

Some Basic Observations for BDD

Allowed (linear algebra) transformations

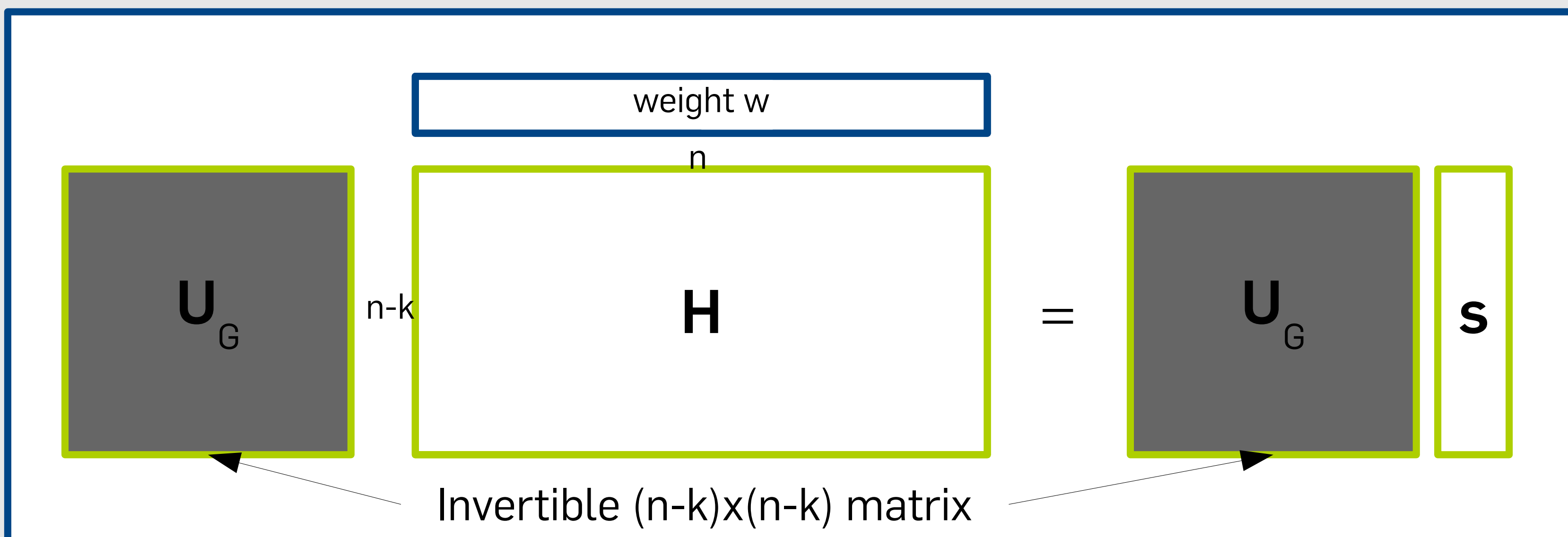
- Permuting the columns of **H** does not change the problem
- Elementary row operations on **H** do not change the problem



Some Basic Observations for BDD

Allowed (linear algebra) transformations

- Permuting the columns of **H** does not change the problem
- Elementary row operations on **H** do not change the problem



Randomized quasi-systematic form

RUB

- Work on randomly column-permuted version of **H**
- Transform **H** into quasi-systematic form

$$\mathbf{H} = \begin{array}{cc|cl} & k+l & n-k-l & \\ \hline \left[\right. & \mathbf{q}_1, \ldots, \mathbf{q}_{k+l} & \mathbf{0} & \left.\right] \; \} \; l \text{ rows} \\ \hline \left[\right. & \mathbf{Q}' & \mathbf{I}_{n-k-l} & \left.\right] \end{array}$$

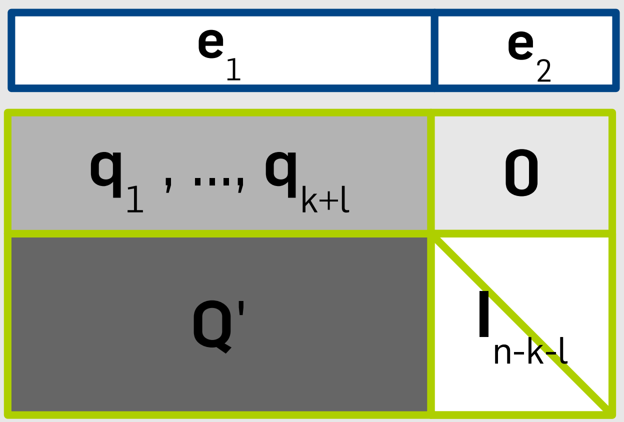
First used in generalized ISD framework of [FS09]

Information Set Decoding

''Reducing the brute-force search space by linear algebra.''

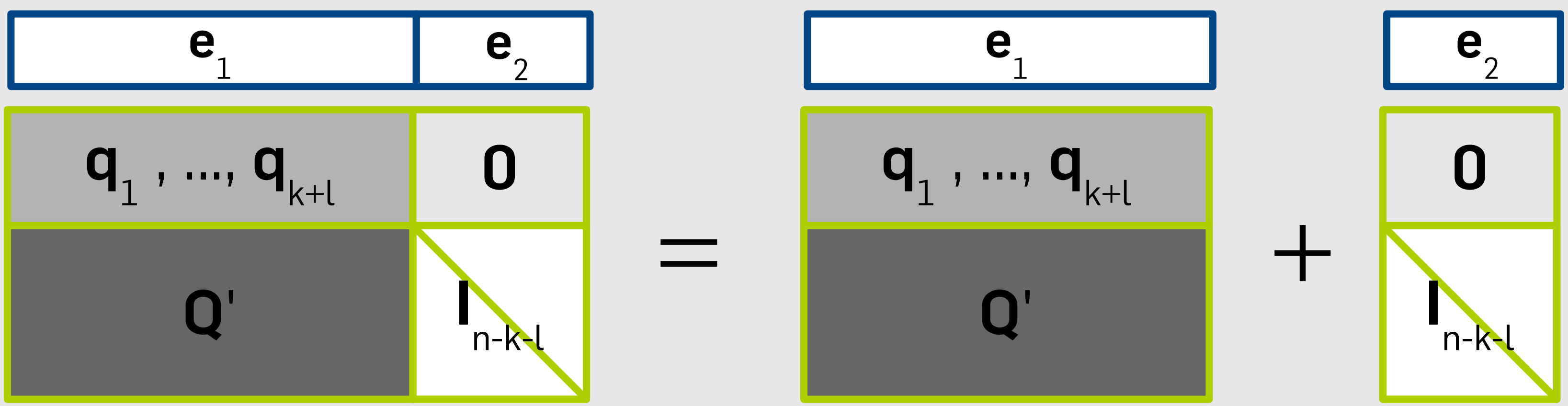
The ISD Principle

- Structure of **H** allows to divide $\mathbf{e} = \left[\ \overset{k+l}{\mathbf{e}_1}\ \middle|\ \overset{n-k-l}{\mathbf{e}_2}\ \right]$



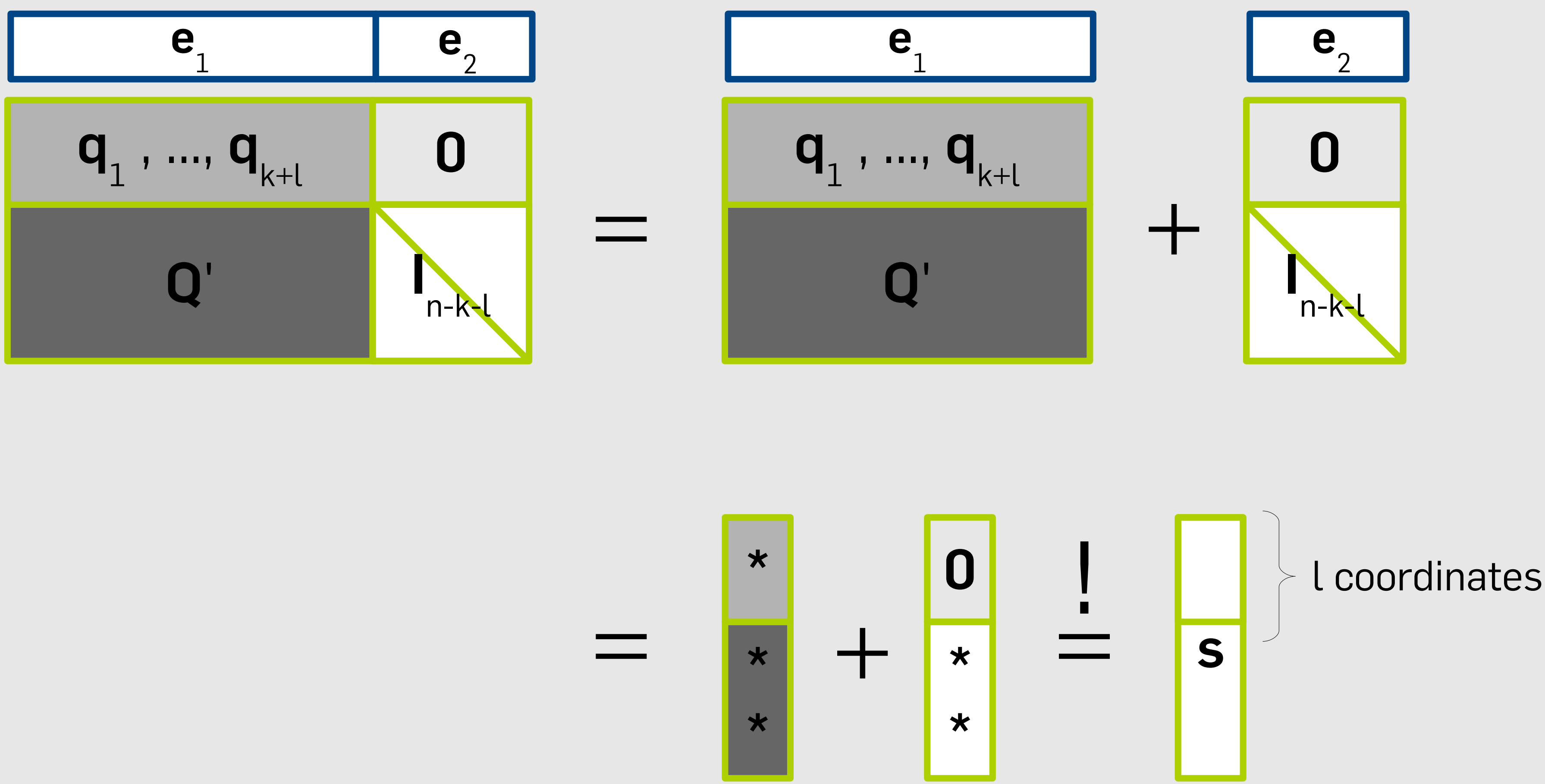
The ISD Principle

- Structure of **H** allows to divide $\mathbf{e} = \begin{array}{|c|c|}\hline \mathbf{e}_1 & \mathbf{e}_2 \\ \hline \end{array}$



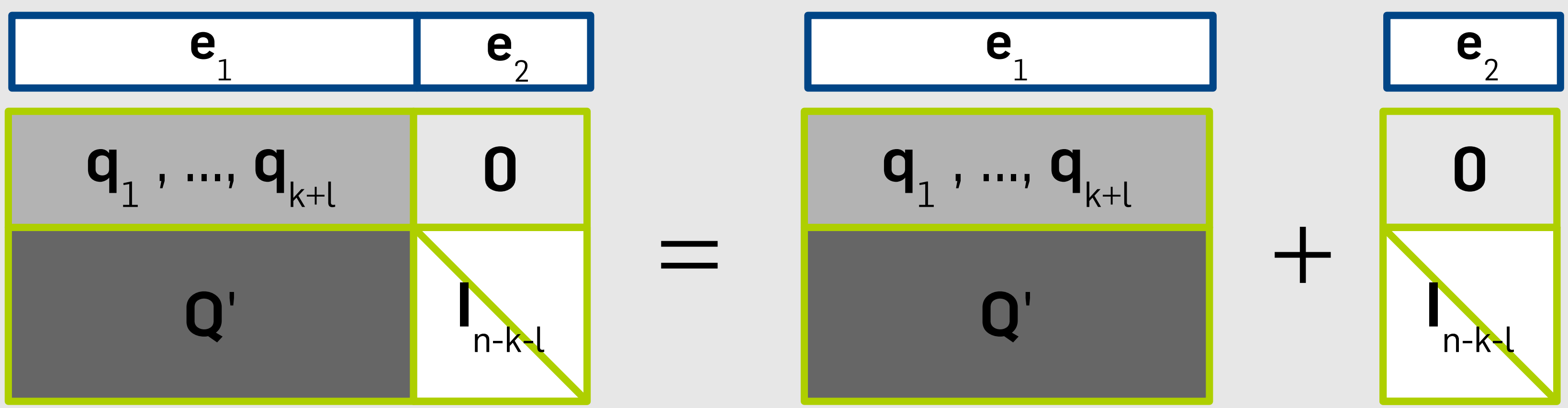
The ISD Principle

- Structure of **H** allows to divide $\mathbf{e} = \begin{array}{|c|c|}\hline \mathbf{e}_1 & \mathbf{e}_2 \\ \hline\end{array}$

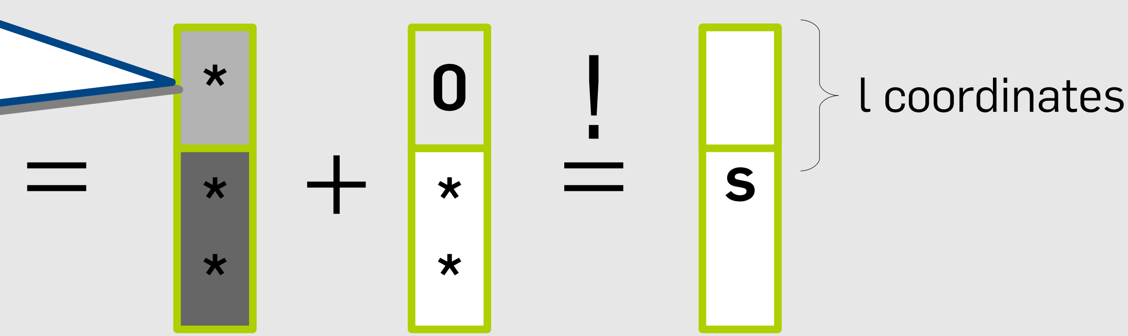


The ISD Principle

- Structure of **H** allows to divide **e** = $\boxed{\mathbf{e}_1 \mid \mathbf{e}_2}$

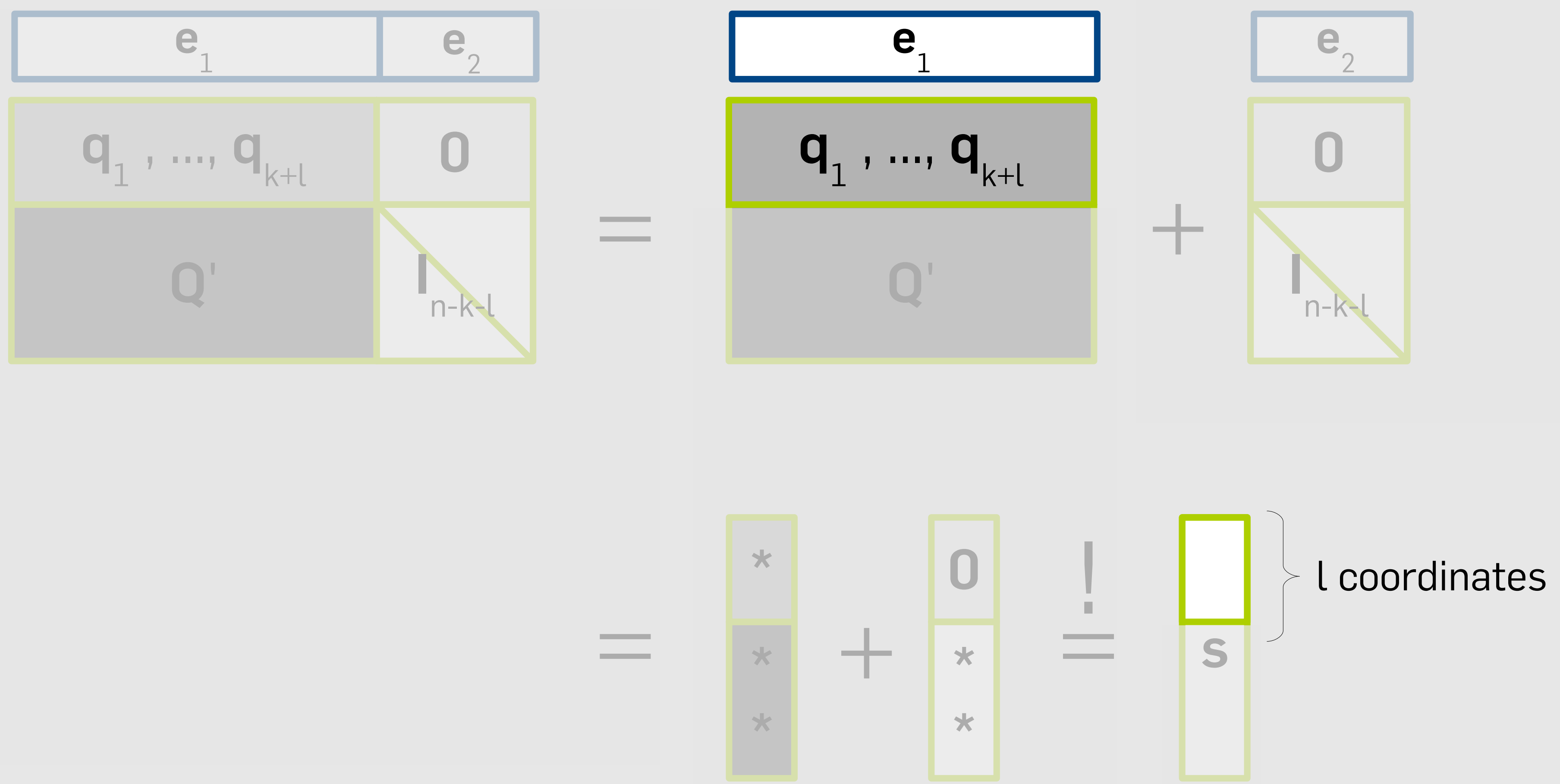


Focus on $\mathbf{e}_1$ matching
s on first l coordinates



The ISD Principle

Find all $\mathbf{e}_1$ of weight p matching $\mathbf{s}$ on first l coordinates



The ISD Principle

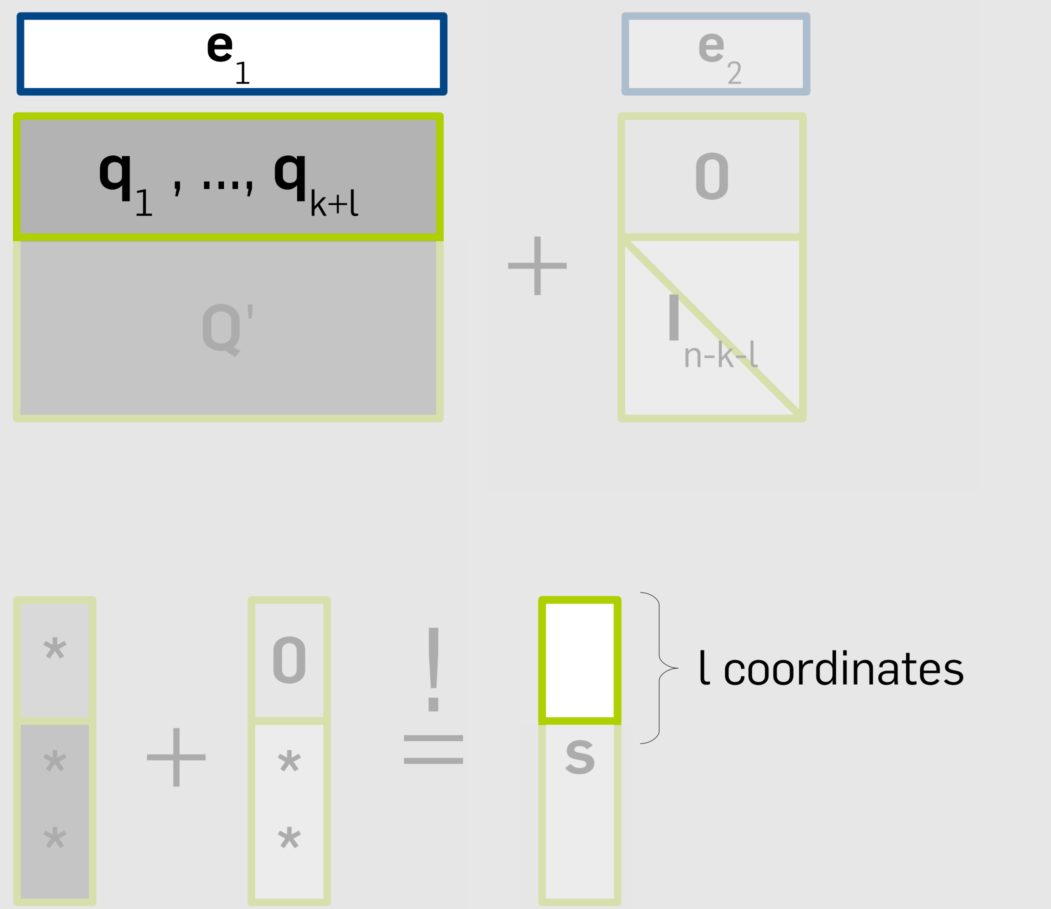
Find all $\mathbf{e}_1$ of weight p matching $\mathbf{s}$ on first l coordinates

- Method only recovers particular error patterns

$k+l$	$n-k-l$
p	$w-p$

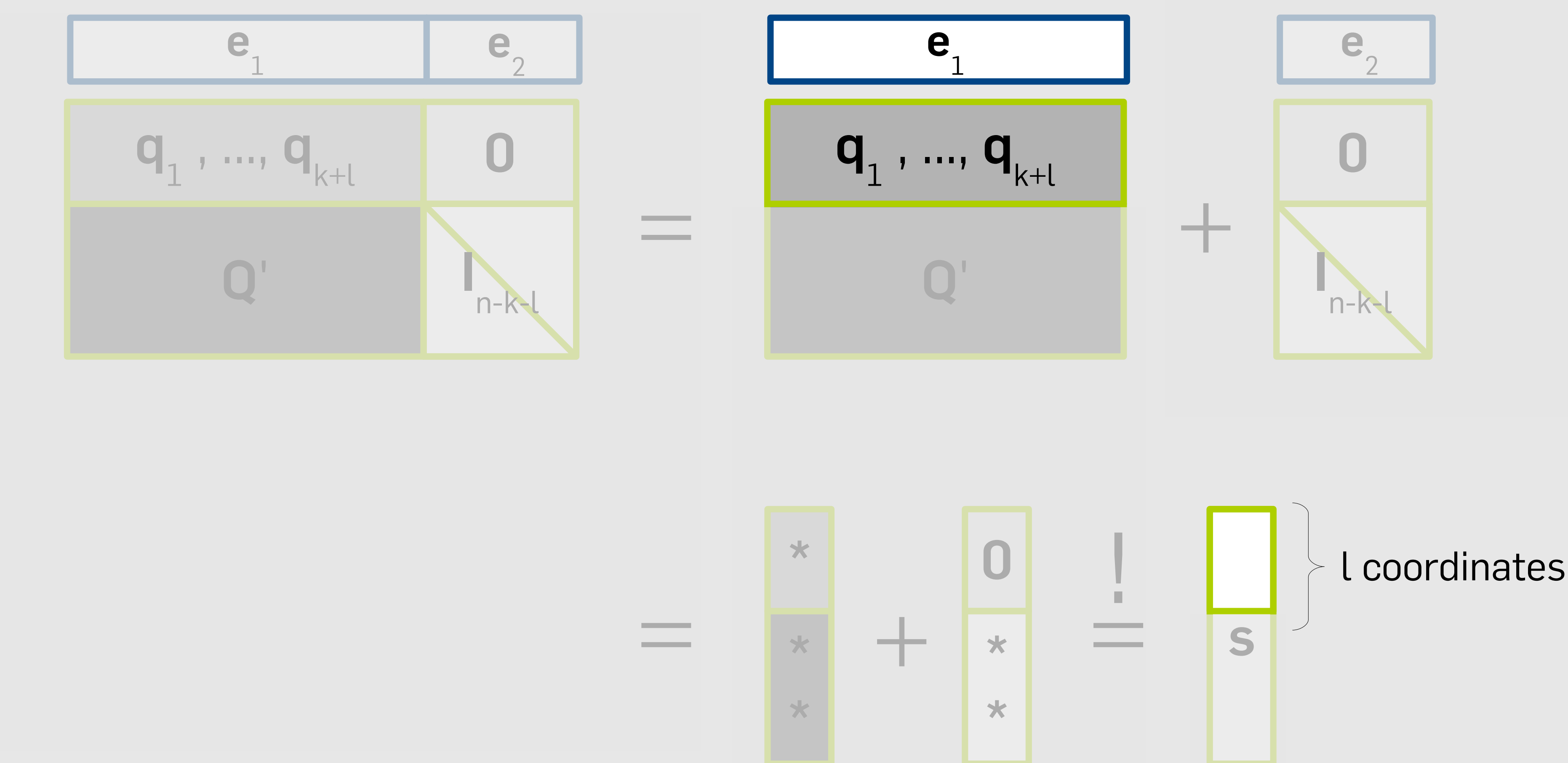
- If we fail to find $\mathbf{e}_1$:

  → Rerandomize $\mathbf{H}$



The ISD Principle

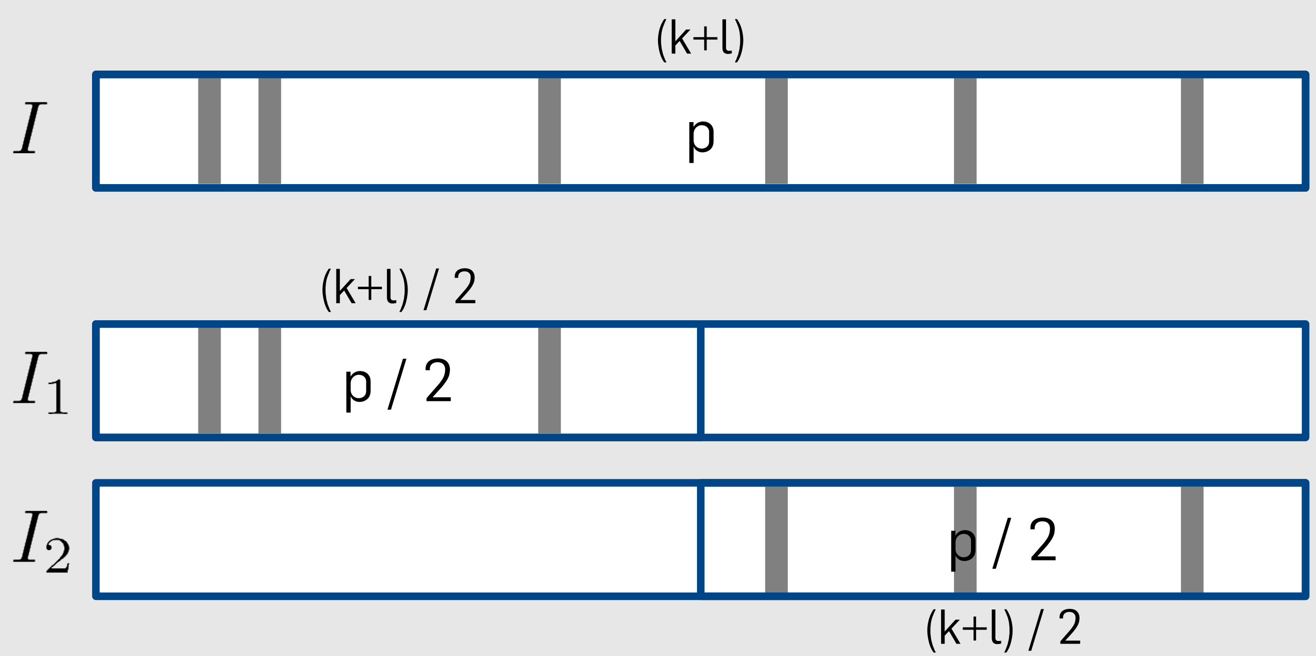
We exploit 1+1=0 to find e_1 more efficiently!



A Meet-in-the-Middle Approach

Find a selection $I \subset [1, \dots, k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

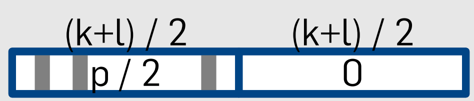
- Disjoint partition $I = I_1 \dot{\cup} I_2$ into left and right half



A Meet-in-the-Middle Approach

Find a selection $I \subset [1, \ldots, k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- To find $I = I_1 \dot{\cup} I_2$ run a Meet-in-the-Middle algorithm based on $\sum_{i \in I_1} q_i = \sum_{j \in I_2} q_j + s$
- Haystack = set of all 

- Needle = unique 

- Same F(k) as recent Ball-Collision decoding [BLP11] as shown in [MMT11]

A Meet-in-the-Middle Approach

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- To find $I = I_1 \dot\cup I_2$ run a Meet-in-the-Middle algorithm based on $\sum_{i \in I_1} q_i = \sum_{j \in I_2} q_j + s$
- Haystack = set of all

(k+l) / 2	(k+l) / 2
p / 2	0

- Needle = unique

(k+l) / 2	(k+l) / 2
p / 2	0

F(k) ≤ 0.0556

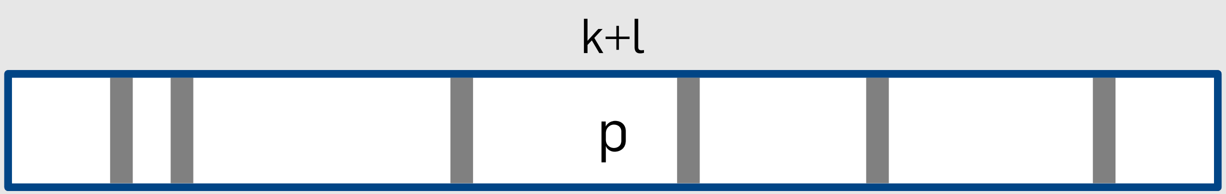
- Same F(k) as recent Ball-Collision decoding [BLP11] as shown in [MMT11]

Using Representations [MMT11]

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Basic representation technique
- Arbitrary disjoint partition

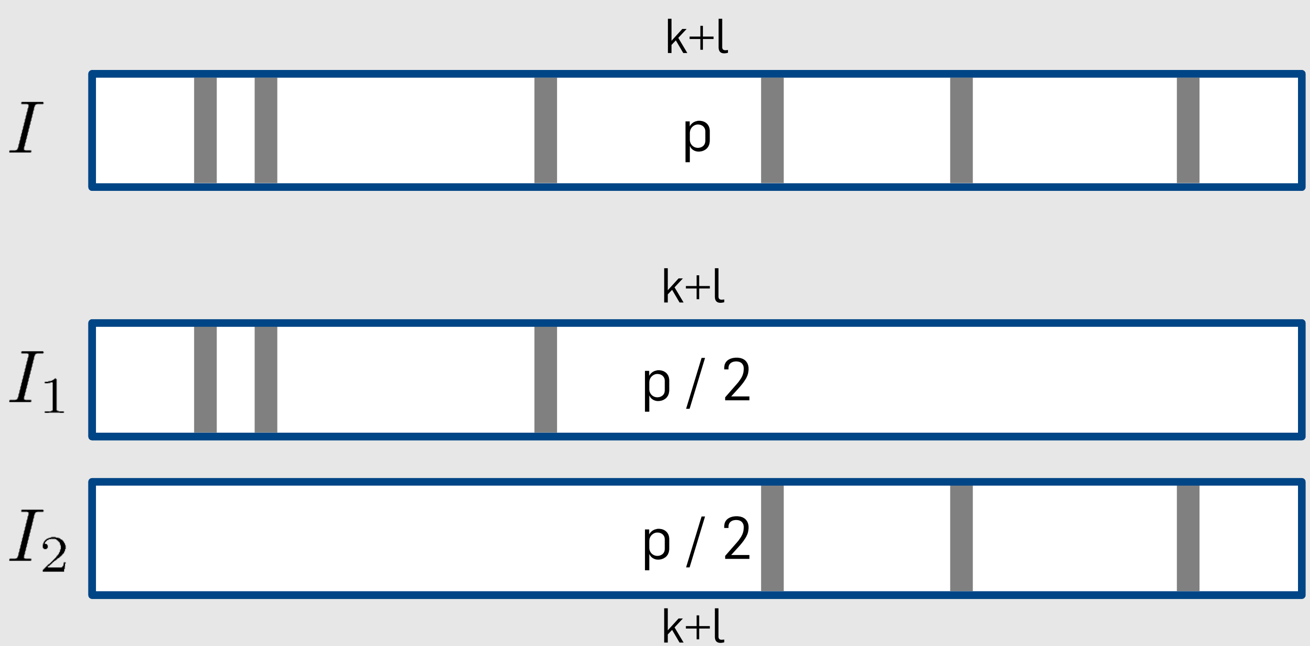
I



Using Representations [MMT11]

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i\in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

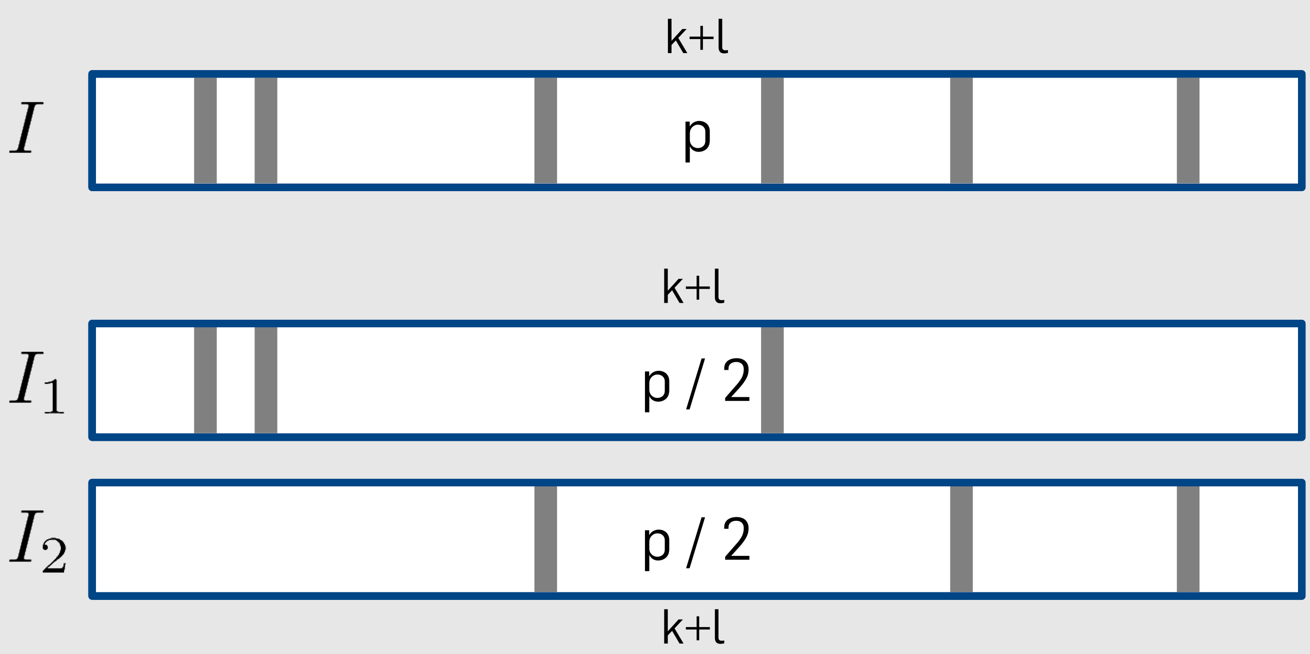
- Basic representation technique
- Arbitrary disjoint partition



Using Representations [MMT11]

Find a selection $I \subset [1, \ldots, k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

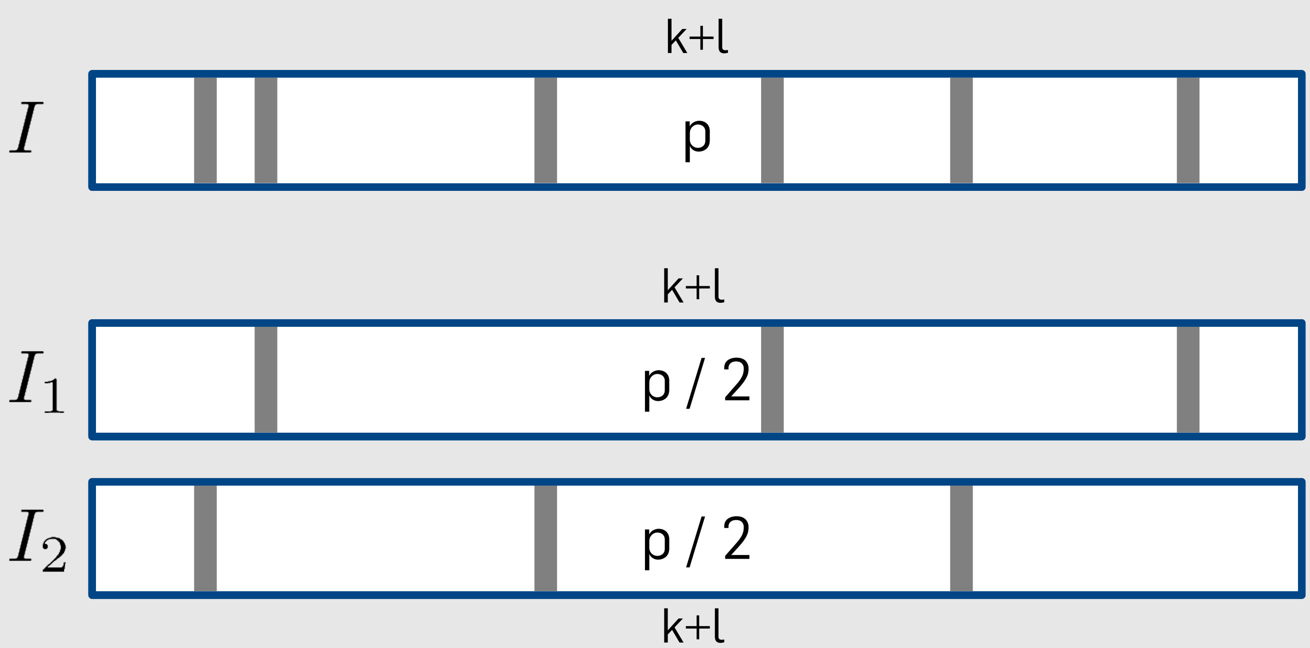
- Basic representation technique
- Arbitrary disjoint partition



Using Representations [MMT11]

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Basic representation technique
- Arbitrary disjoint partition

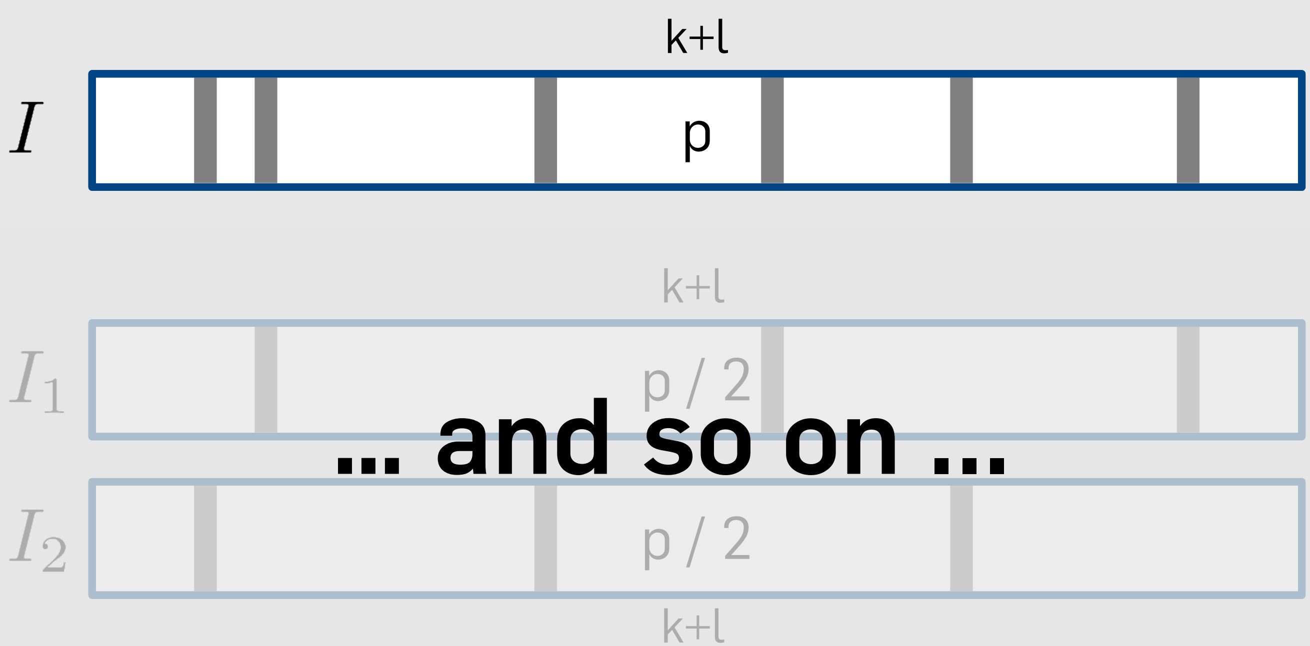


Using Representations [MMT11]

RUB

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Basic representation technique
- Arbitrary disjoint partition

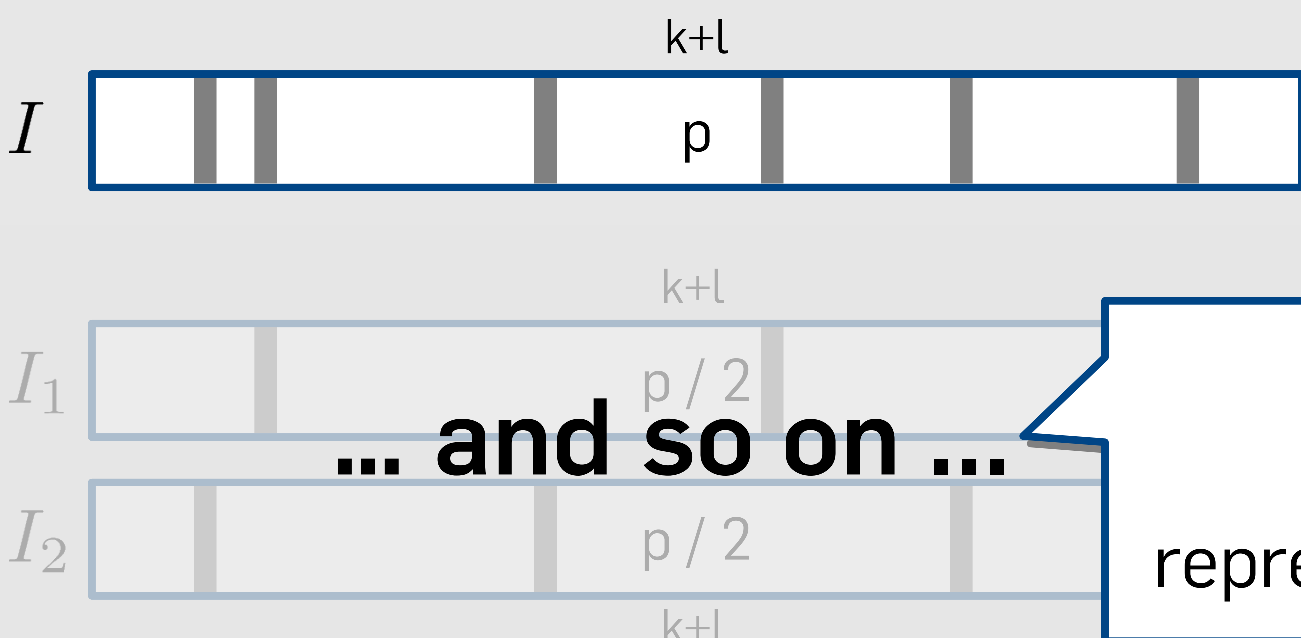


… and so on …

Using Representations [MMT11]

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Basic representation technique
- Arbitrary disjoint partition

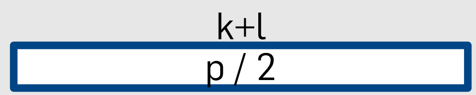


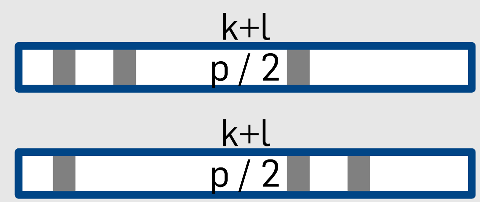
... and so on ...

$$\binom{p}{p/2}$$
representations

Using Representations [MMT11]

Find a selection $I \subset [1,\ldots,k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Haystack = set of all 

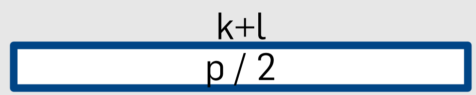
- Needles = $\binom{p}{p/2}$ representations 

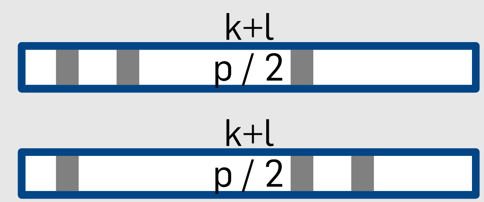
, , ...

- Bottleneck: Efficient computation of a $\frac{1}{\binom{p}{p/2}}$ - fraction of the haystack

Using Representations [MMT11]

Find a selection $I \subset [1, \ldots, k+l], |I| = p$ with $\sum_{i \in I} q_i = \begin{pmatrix} s_1 \\ \vdots \\ s_l \end{pmatrix}$

- Haystack = set of all 

- Needles = $\binom{p}{p/2}$ representations 

, , ...

- Bottleneck: Efficient computation of a $\frac{1}{\binom{p}{p/2}}$ - fraction of the haystack

F(k) ≤ 0.0537

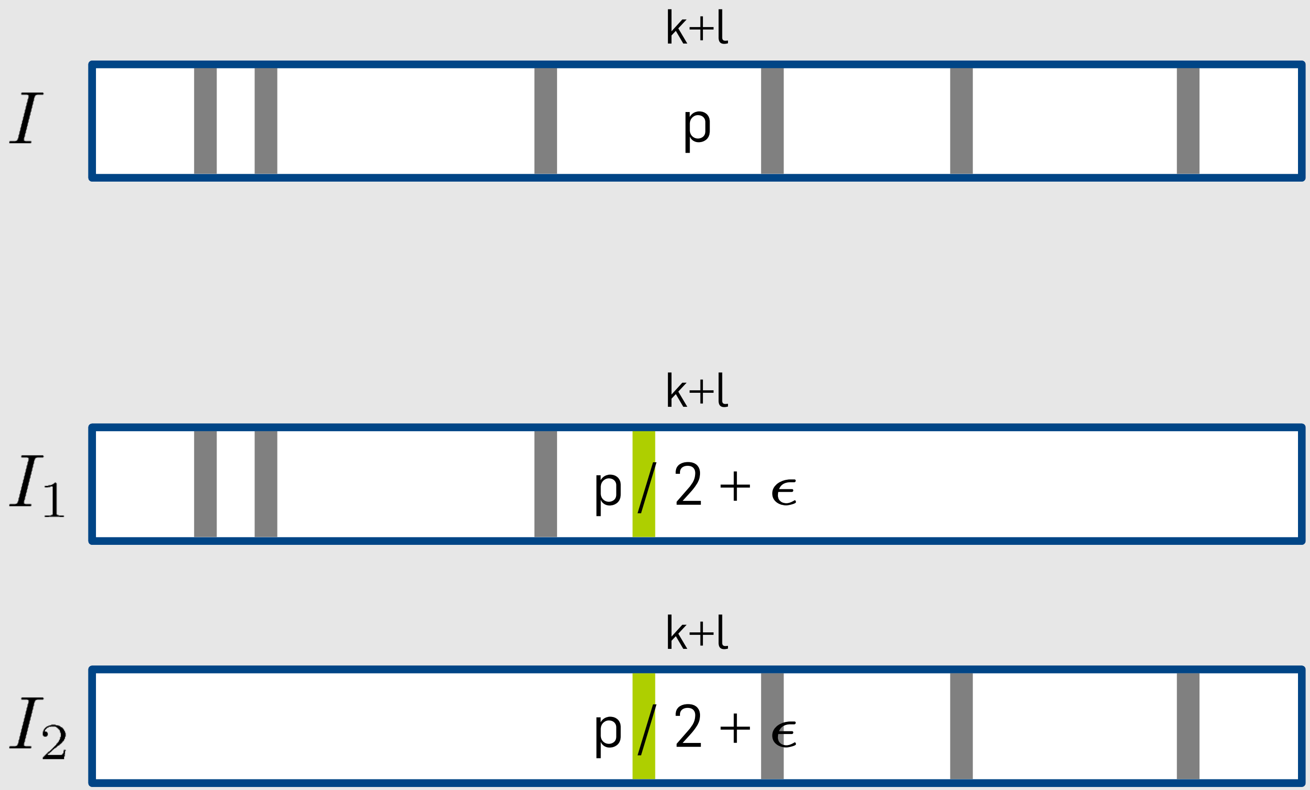
Using 1 + 1 = 0

How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

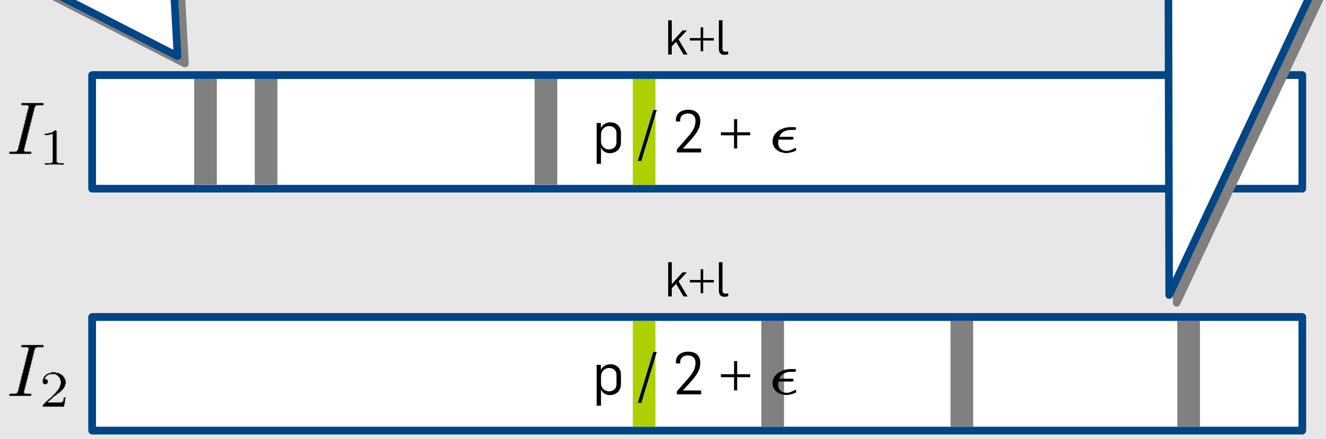


How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

$$\sum_{i \in I_1} q_i$$

$$= \sum_{j \in I_2} q_j + s$$



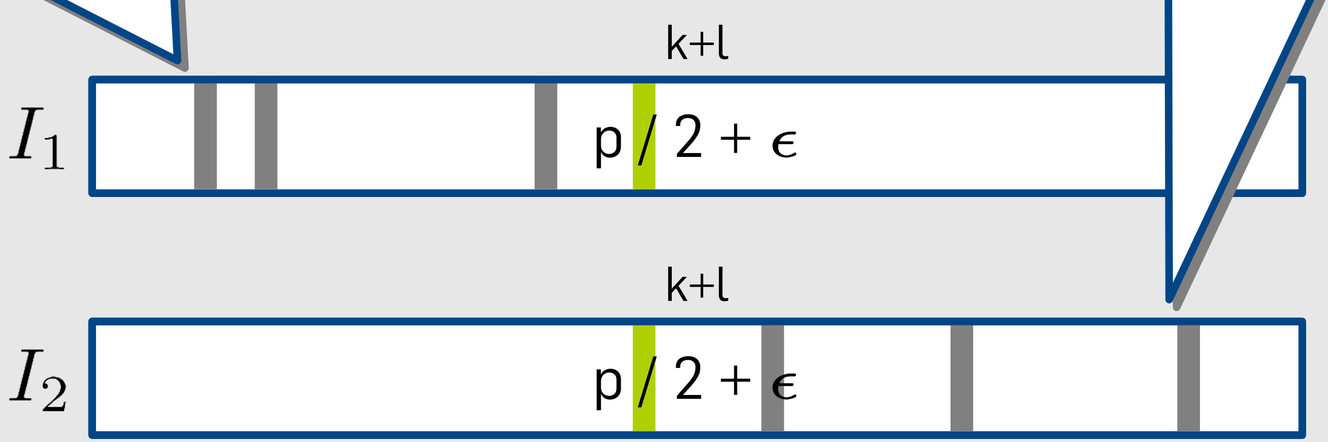
How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

$$\sum_{i \in I_1} q_i$$

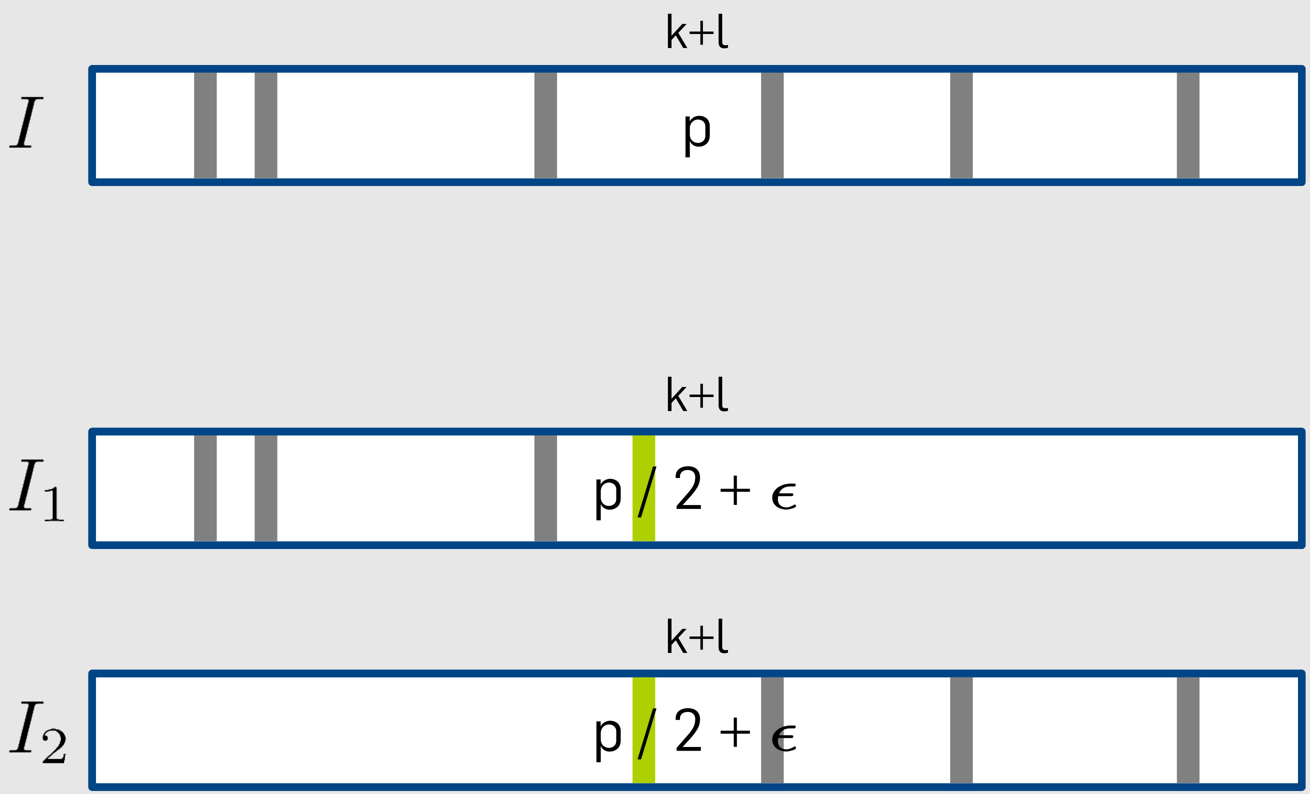
Double columns cancel out due to 1+1=0 !

$$= \sum_{j \in I_2} q_j + s$$



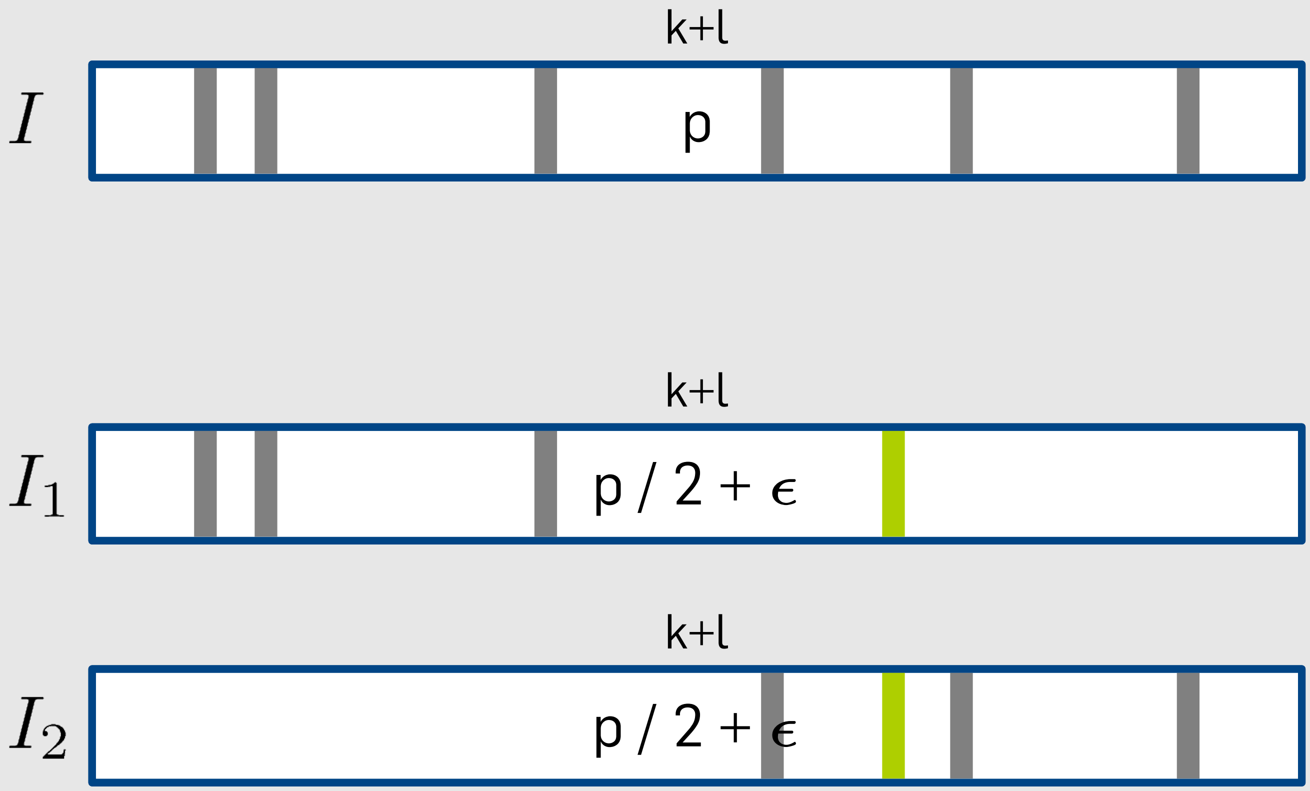
How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$



How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

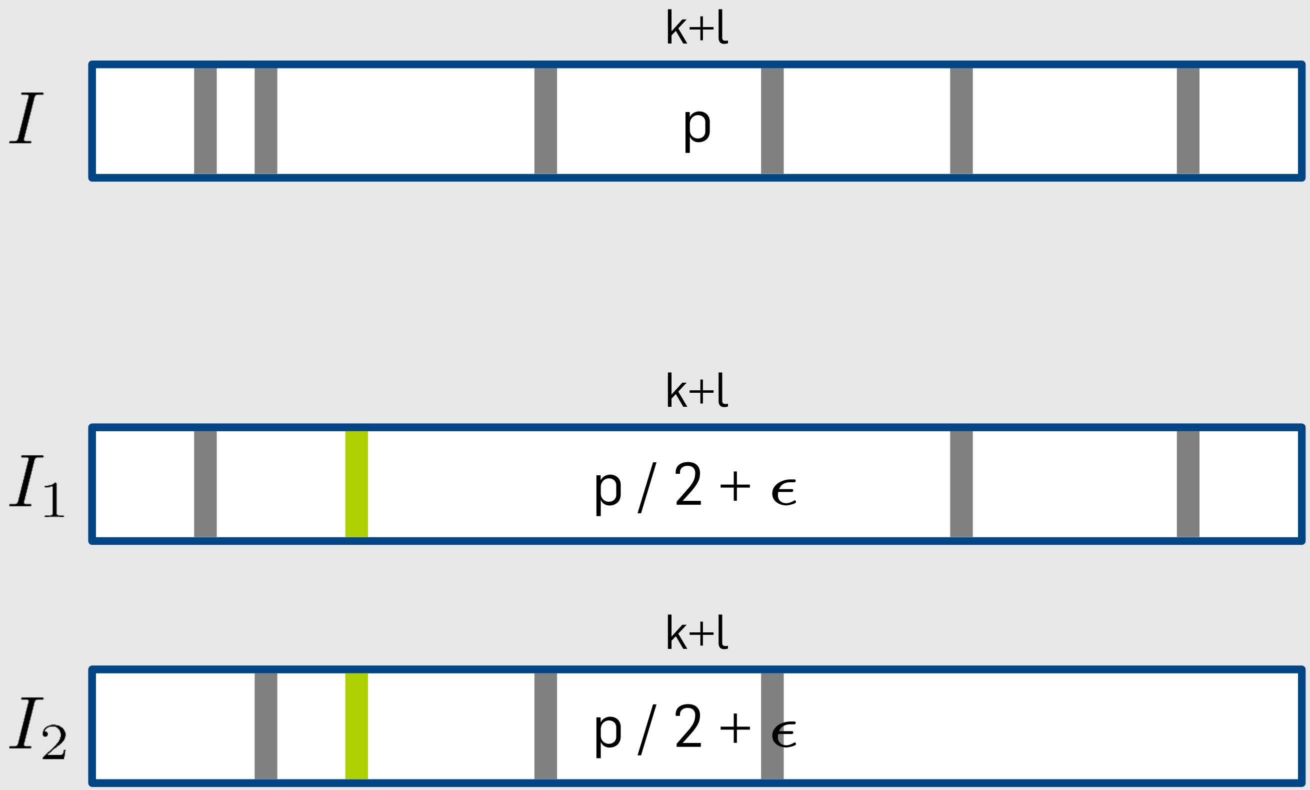


How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

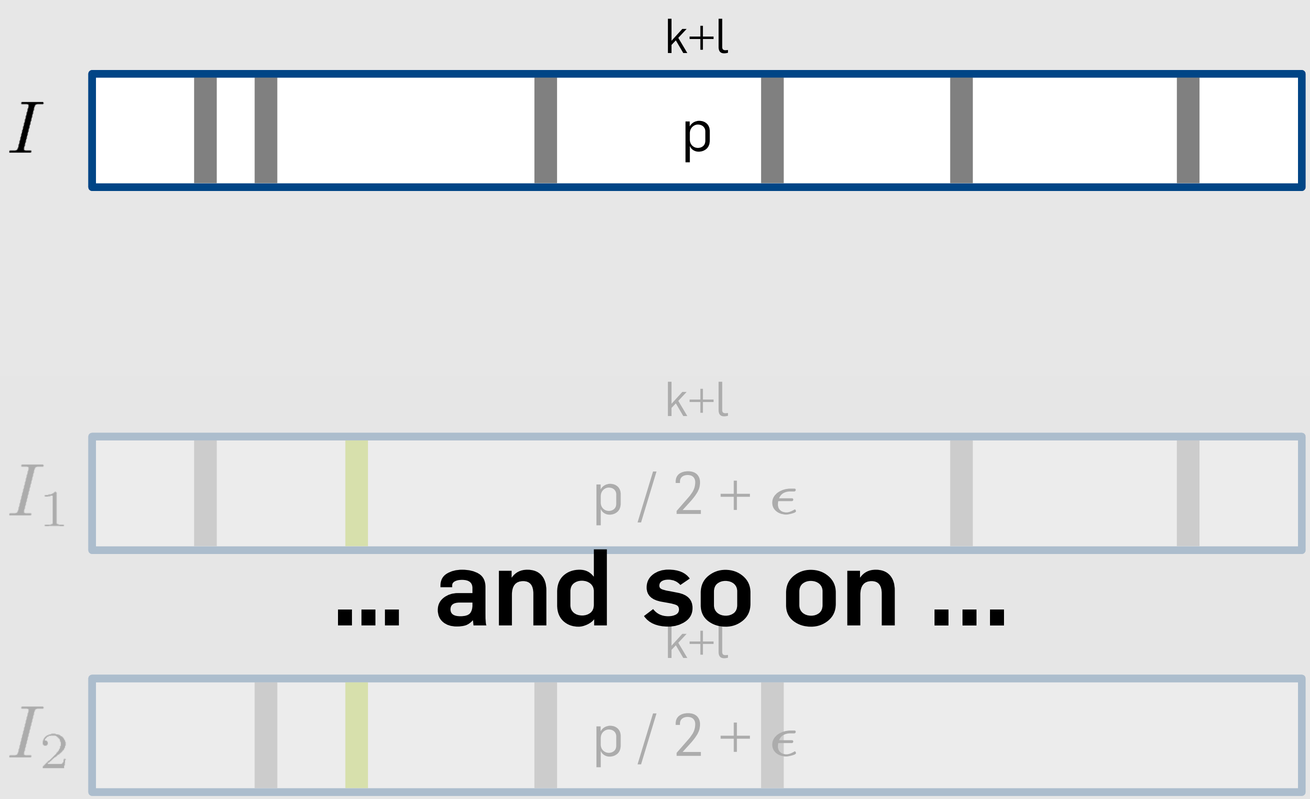
How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$



How to use 1 + 1 = 0

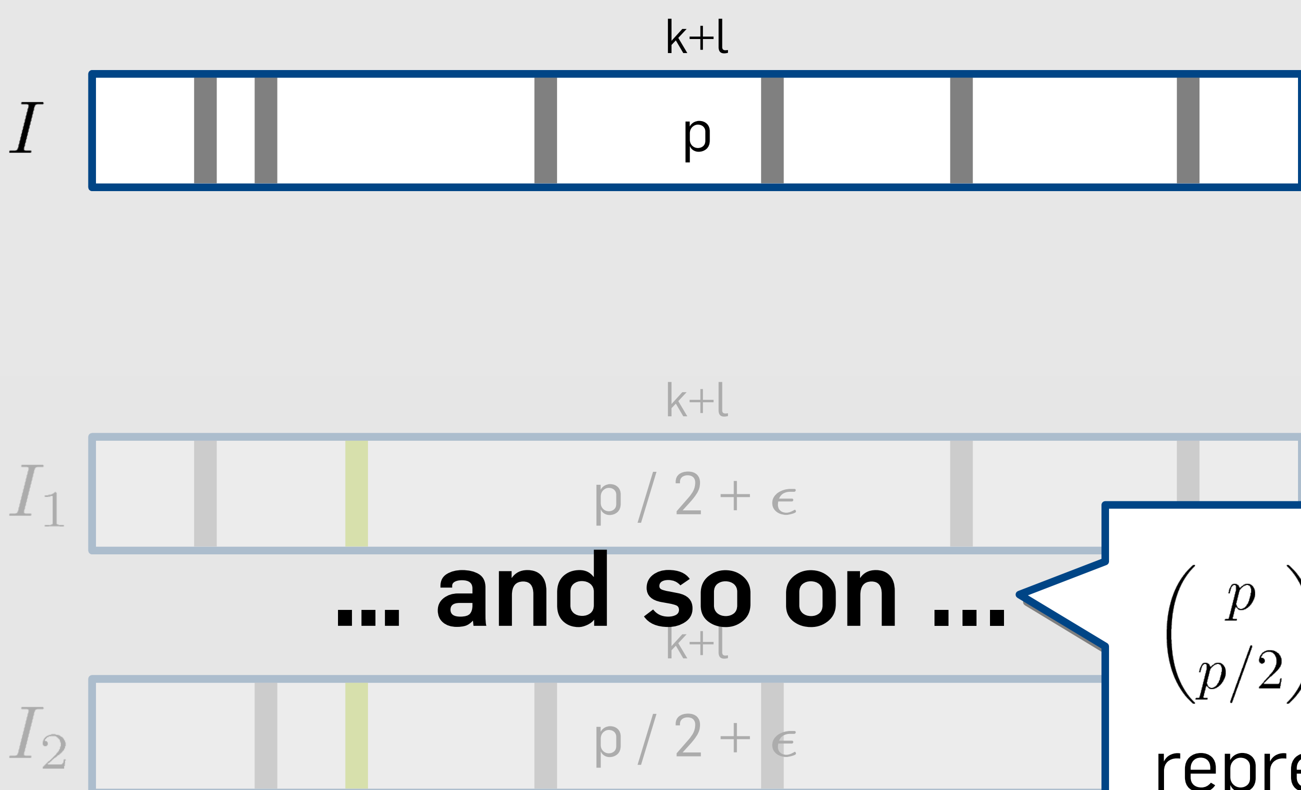
Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$



… and so on …

How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$



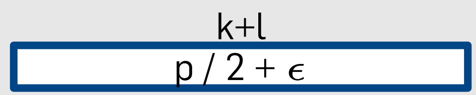
… and so on …

$$\binom{p}{p/2} \cdot \binom{k+l-p}{\varepsilon}$$

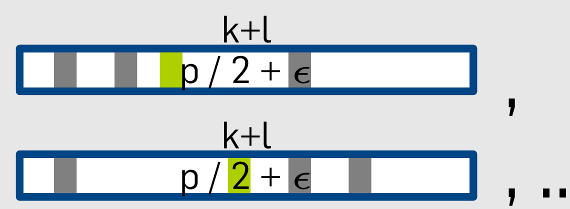
representations

How to use 1 + 1 = 0

Write $I = I_1 \Delta I_2 := (I_1 \cup I_2) \setminus (I_1 \cap I_2)$ as the symmetric difference of intersecting sets $|I_1 \cap I_2| = \varepsilon$

- Haystack = set of all 

- Needles = $\underbrace{\binom{p}{p/2}\binom{k+l-p}{\varepsilon}}_{=:R}$ representations



How can we compute a 1/R – fraction of the haystack ?

How to use 1 + 1 = 0

How can we compute a 1/R – fraction of the haystack ?

- Want to find *one* needle I_1 (and suitable I_2) with

$$\sum_{i \in I_1} q_i = \sum_{j \in I_2} q_j + s$$

$$\mathbf{q}_1 + \mathbf{q}_3 + \mathbf{q}_4 + \mathbf{q}_{11} = \mathbf{q}_2 + \mathbf{q}_4 + \mathbf{q}_7 + \mathbf{q}_{12} + \mathbf{s}$$

How to use 1 + 1 = 0

How can we compute a 1/R – fraction of the haystack ?

Uniform 0/1 - coordinates

he needle I_1 (and suitable I_2) with

$$\sum_{i \in I_1} q_i = \sum_{j \in I_2} q_j + s$$

$$\mathbf{q}_1 + \mathbf{q}_3 + \mathbf{q}_4 + \mathbf{q}_{11} = \mathbf{q}_2 + \mathbf{q}_4 + \mathbf{q}_7 + \mathbf{q}_{12} + \mathbf{s}$$

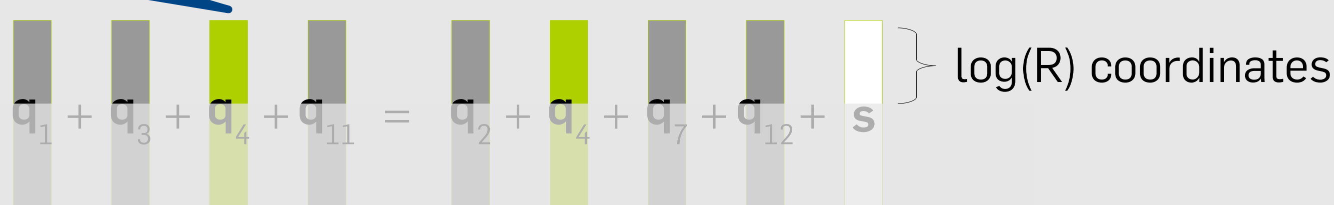
How to use 1 + 1 = 0

How can we compute a 1/R – fraction of the haystack ?

Uniform 0/1 - coordinates

ne needle I_1 (and suitable I_2) with

$$\sum_{i \in I_1} q_i = \sum_{j \in I_2} q_j + s$$



- Fix $\sum_{i \in I_1} q_i$ to **0** and $\sum_{j \in I_2} q_j$ to **s** on log(R) coordinates

→ **Expect one needle to fulfill the extra constraint!**

Some More Details

RUB

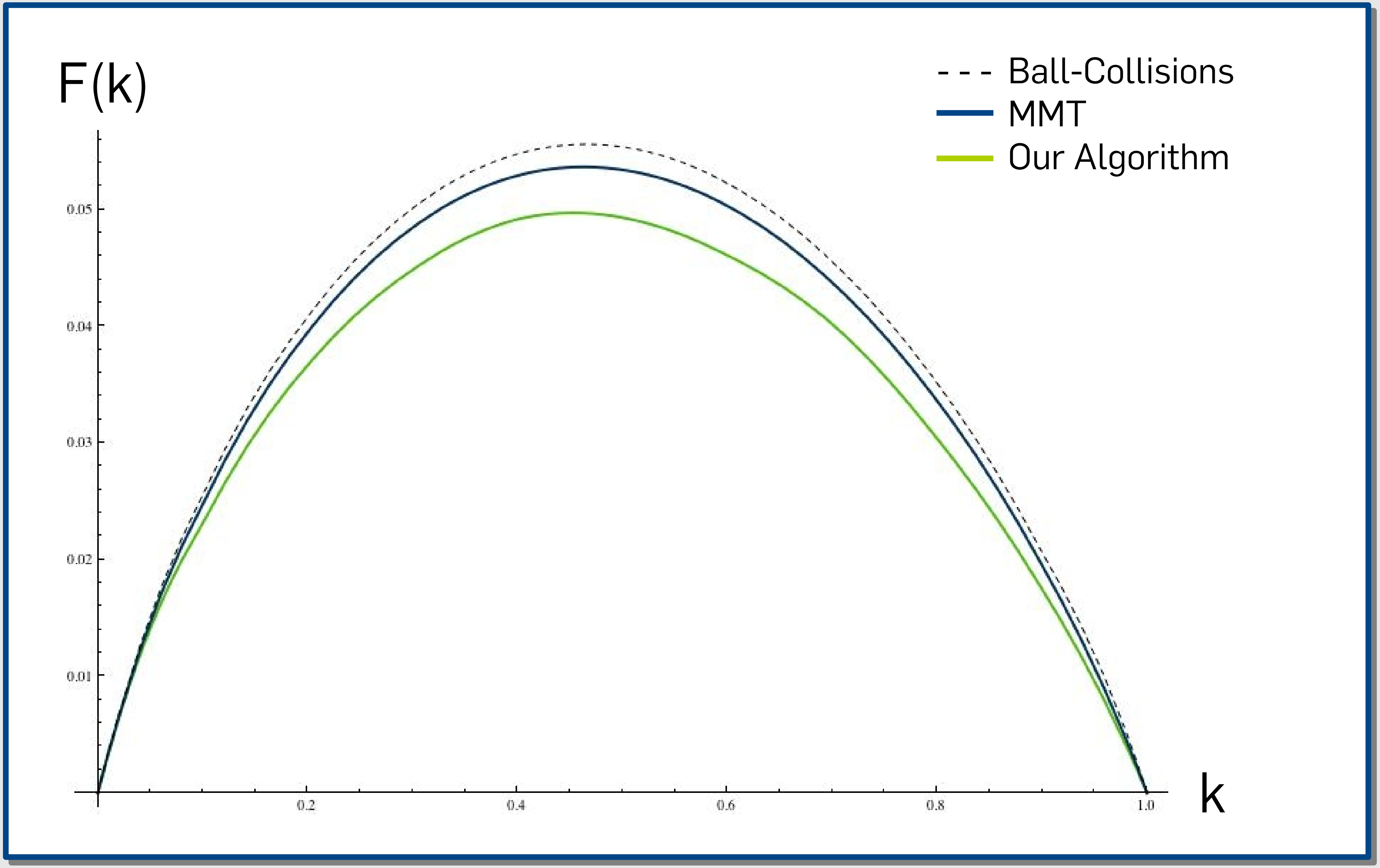
The actual search for the needle

- à la Wagner's Generalized Birthday Algorithm
- Three-layered binary computation tree

Some technicalities

- Need to exclude "badly distributed" $\mathbf{q}_1 , \ldots , \mathbf{q}_{k+l}$
- Method introduces extra inverse-polynomial failure probability

Main Result $F(k) \le 0.04934 < 1/20$



Wrapping up...

RUB

Summary

- Using 1+1=0 introduces extra representations
- Asymptotically fastest generic decoding algorithm
- Full Version ePrint 2012/026

Open Questions

- More representations? Over $\mathbb{F}_q$?
- (Low level) optimizations

Wrapping up...

RUB

Summary

- Using 1+1=0 introduces extra representations
- Asymptotically fastest generic decoding algorithm
- Full Version ePrint 2012/026

Thank you!

Open Questions

- More representations? Over $\mathbb{F}_q$?
- (Low level) optimizations