List-Decoding Reed-Solomon Codes in the Lee, Euclidean, and Other Metrics

ACCESS

Alexandra Veliche Hostetler

University of South Florida

based on joint work with Chris Peikert

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}}$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

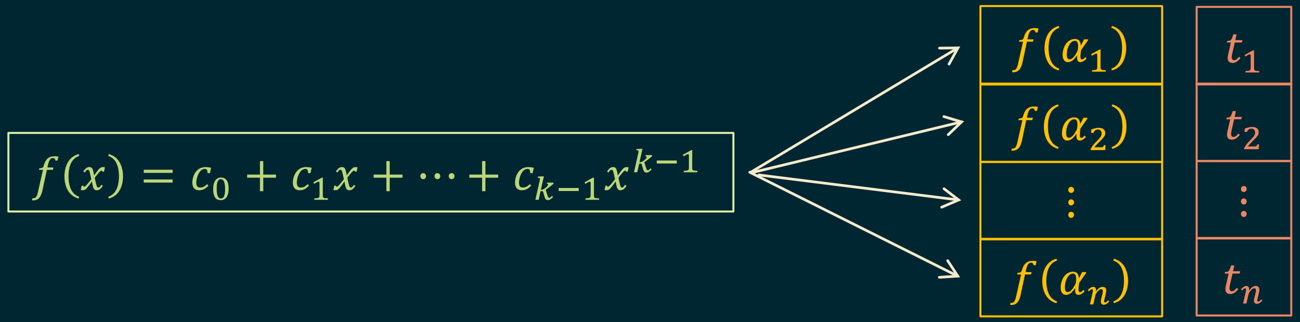
$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}} \rightarrow \begin{array}{|c|} \hline f(\alpha_1) \\ \hline f(\alpha_2) \\ \hline \vdots \\ \hline f(\alpha_n) \\ \hline \end{array}$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \{(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg(f) < k\} \subseteq \mathbb{F}_q^n.$$

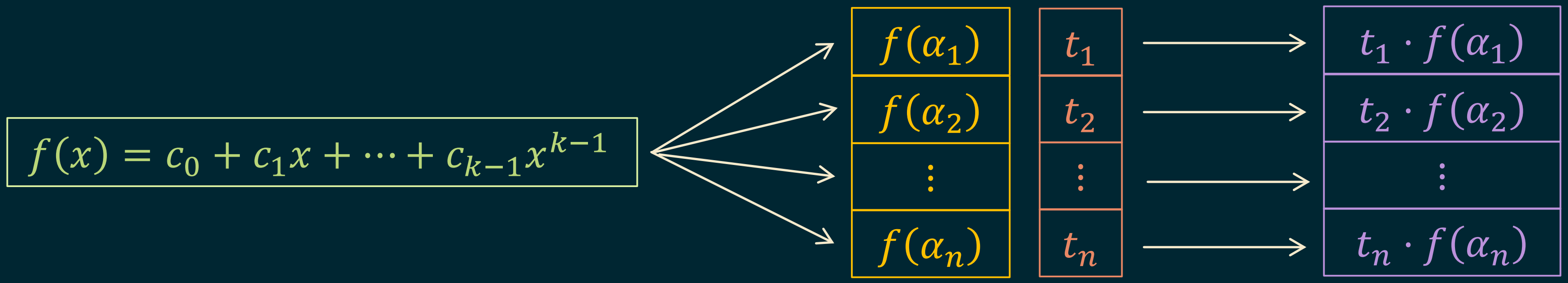


Generalized Reed-Solomon Codes

<u>GRS Code:</u> n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$



Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

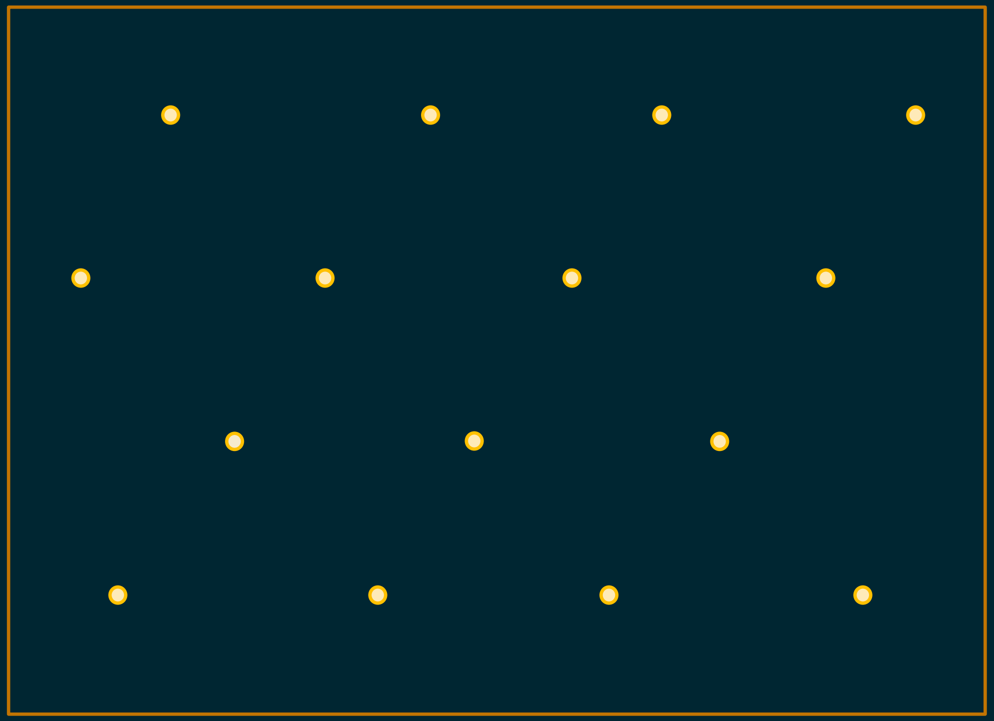
$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}} \rightarrow \begin{bmatrix} f(\alpha_1) \\ f(\alpha_2) \\ \vdots \\ f(\alpha_n) \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} \begin{matrix} \longrightarrow \\ \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{matrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \in GRS$$

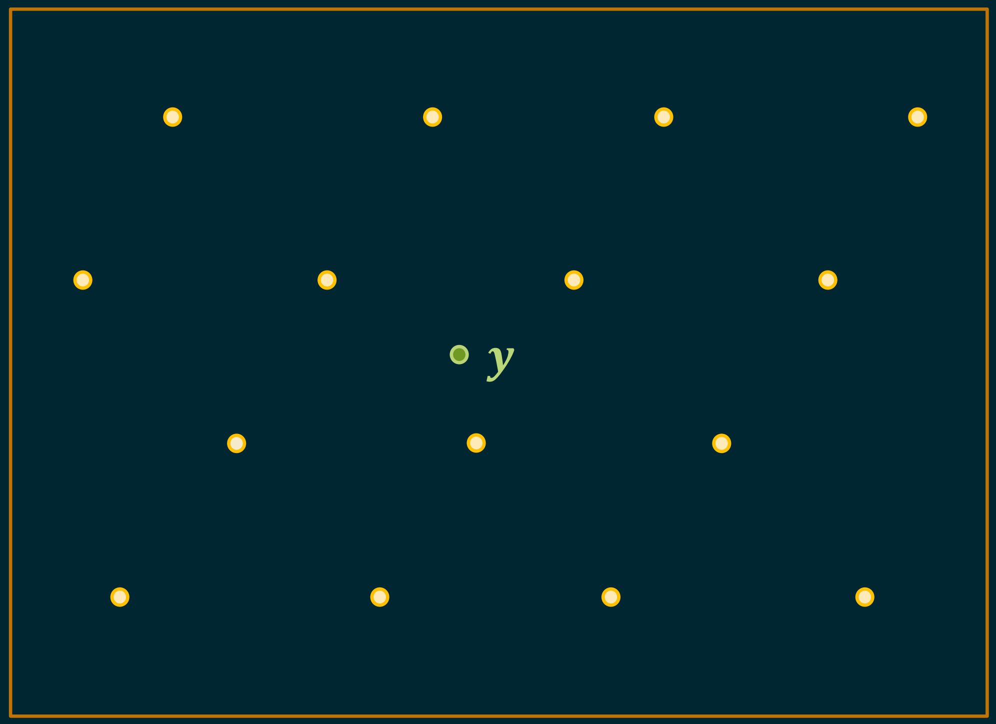
List-Decoding Problem

$\mathcal{C}$ code



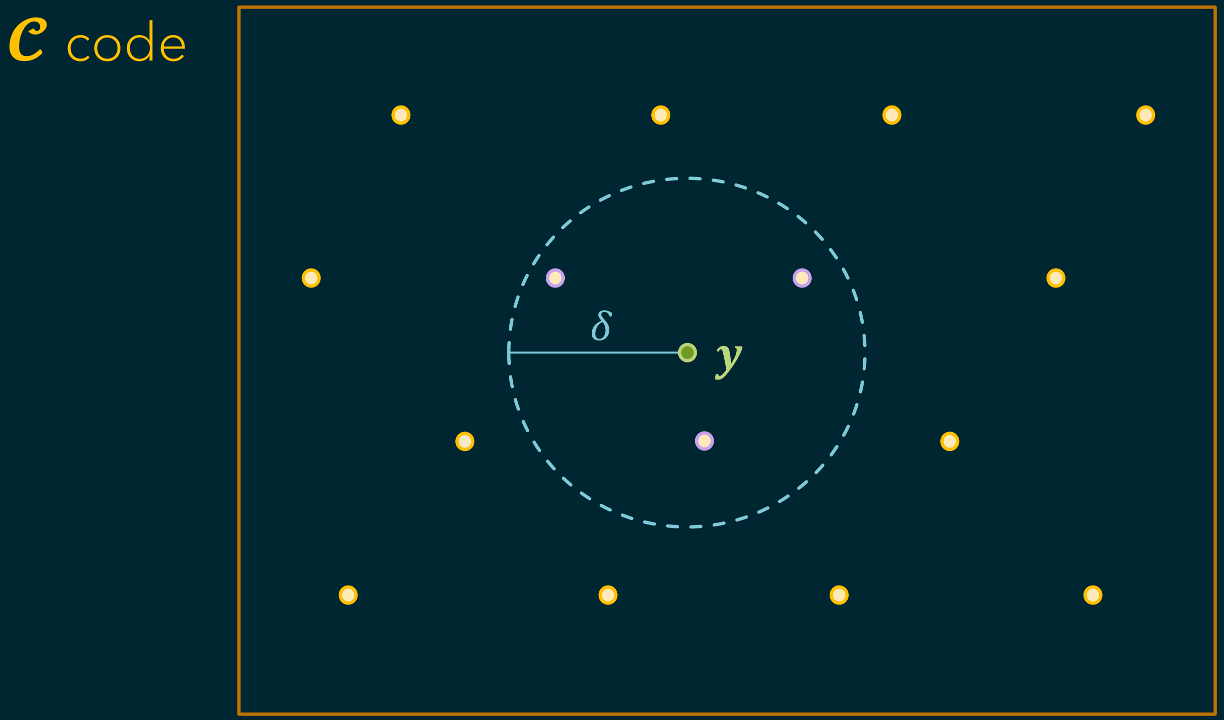
List-Decoding Problem

$\mathcal{C}$ code



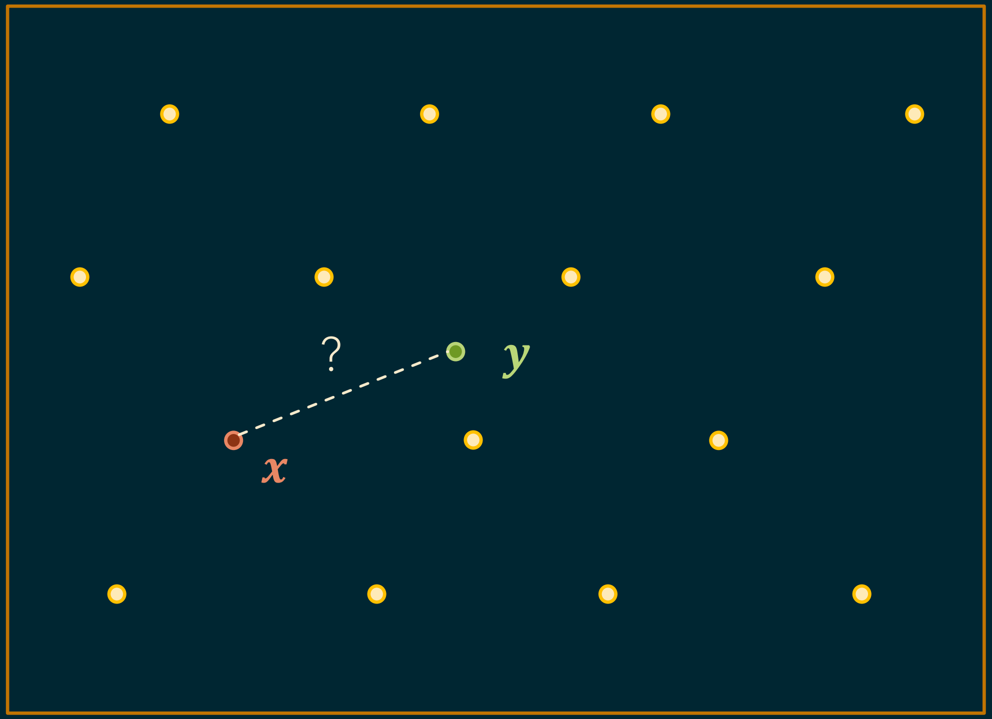
Given a received word y

List-Decoding Problem



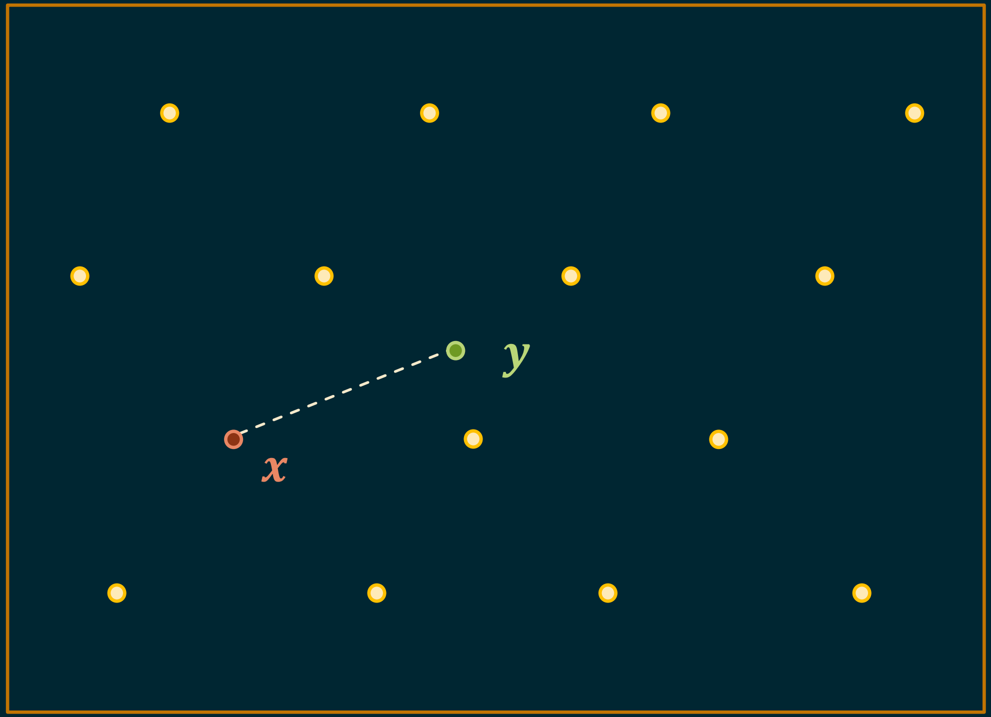
find all codewords within distance δ of y.

Measuring Distance



How is distance measured?

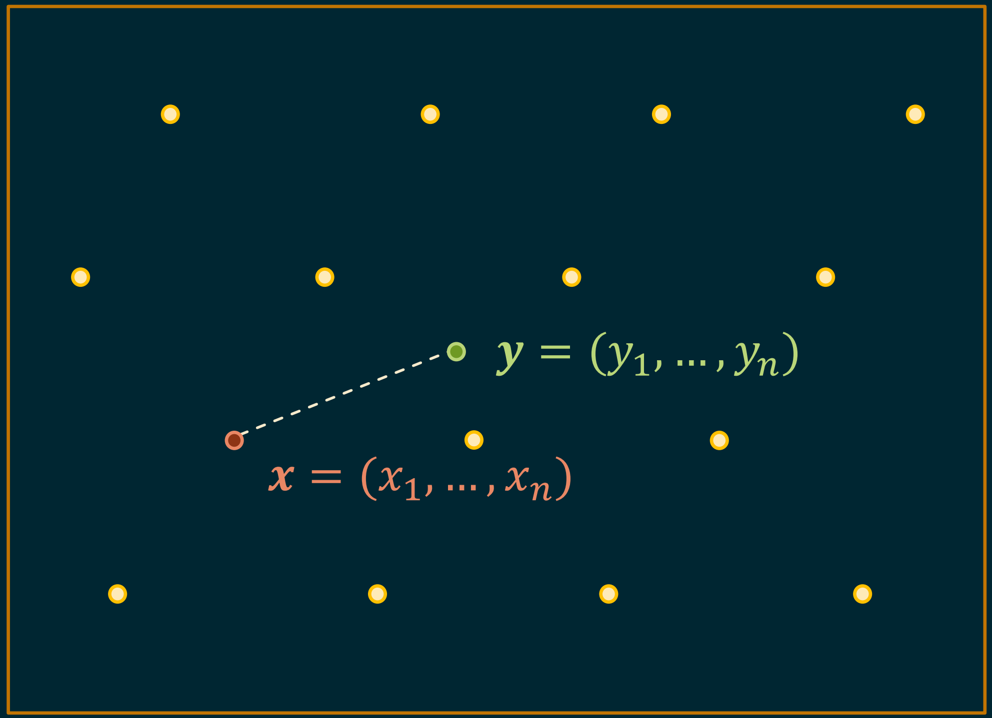
Measuring Distance

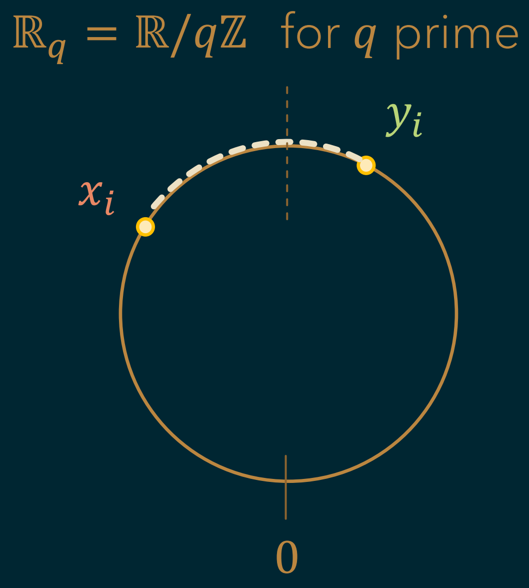


Often in Hamming metric,

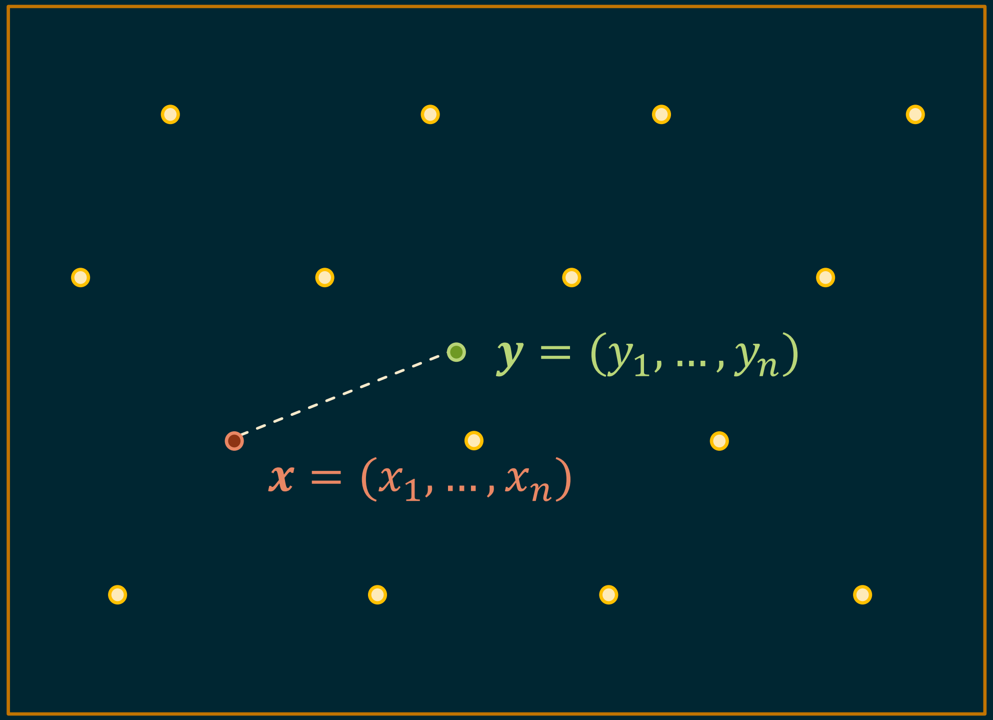
but we consider Lee (ℓ_1), Euclidean (ℓ_2), and other ℓ_p metrics.

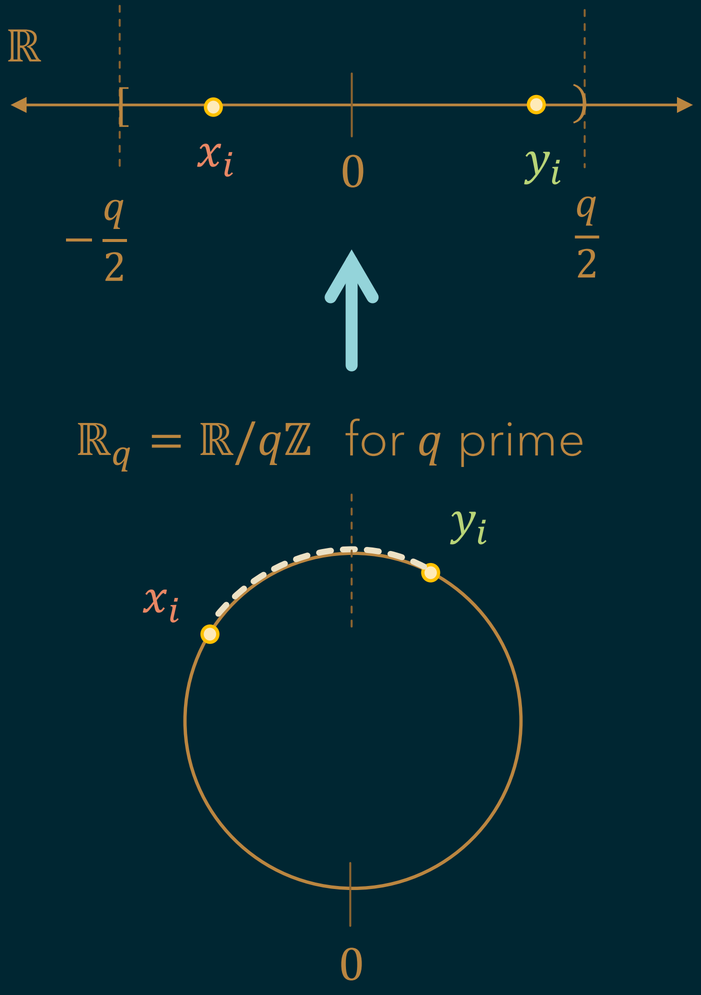
ℓ_p (Semi)Metric





ℓ_p (Semi)Metric





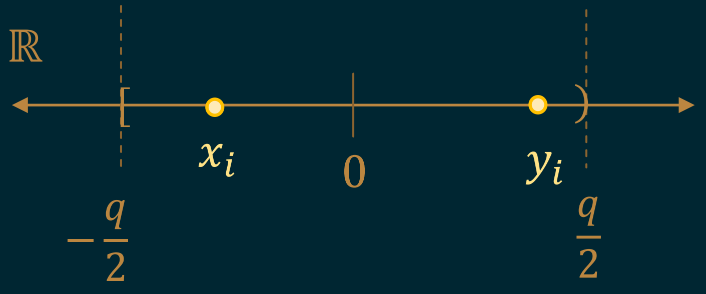
ℓ_p (Semi)Metric

ℓ_p (semi)metric: (on $\mathbb{R}_q^n$) For any $p > 0$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}_q^n$,

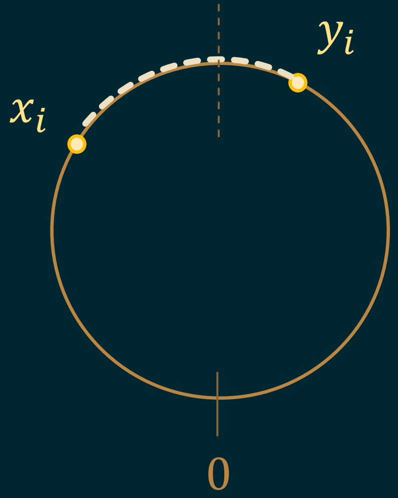
$$d(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \underbrace{\|\overline{\boldsymbol{x} - \boldsymbol{y}}\|_p}.$$

ℓ_p (quasi)norm on $\mathbb{R}^n$ given by

$$\|\boldsymbol{z}\|_p \coloneqq \left(z_1^p + \cdots + z_n^p\right)^{1/p}$$



$\mathbb{R}_q = \mathbb{R}/q\mathbb{Z}$ for q prime



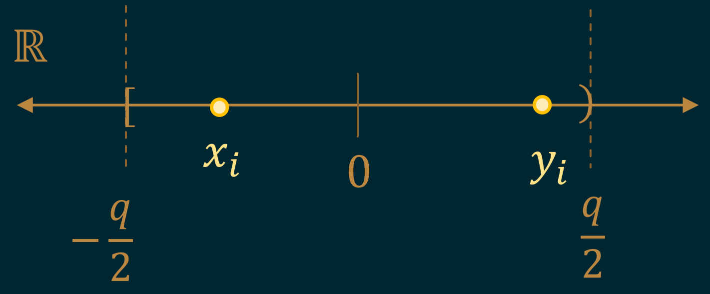
ℓ_p (Semi)Metric

ℓ_p (semi)metric: (on $\mathbb{R}_q^n$) For any $p > 0$ and $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}_q^n$,

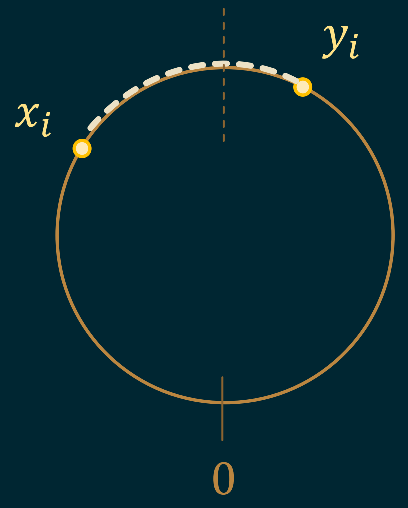
$$d(\boldsymbol{x}, \boldsymbol{y}) := \underbrace{\|\overline{\boldsymbol{x} - \boldsymbol{y}}\|_p}.$$

ℓ_p (quasi)norm on $\mathbb{R}^n$ given by

$$\|\boldsymbol{z}\|_p := \left(z_1^p + \cdots + z_n^p\right)^{1/p}$$



$\mathbb{R}_q = \mathbb{R}/q\mathbb{Z}$ for q prime



Relative decoding distance: $\delta = d/n^{1/p}$.

Our (Informal) Results

Theorem: *(worst-case)* There is an efficient algorithm that list-decodes any prime-field GRS code from *continuous error* in the Lee, Euclidean, or other ℓ_p (semi)metric for $0 < p \leq 2$, up to *arbitrarily large* (relative) distance $\delta > 0$ for corresponding small enough rate R.

Our (Informal) Results

Theorem: *(worst-case)* There is an efficient algorithm that list-decodes any prime-field GRS code from *continuous error* in the Lee, Euclidean, or other ℓ_p (semi)metric for $0 < p \leq 2$, up to *arbitrarily large* (relative) distance $\delta > 0$ for corresponding small enough rate R.

Theorem: *(average-case)* …and similarly for Laplacian / Gaussian channels, with better rate-distance trade-off than for the Lee / Euclidean metrics (respectively).

Prior Algorithms

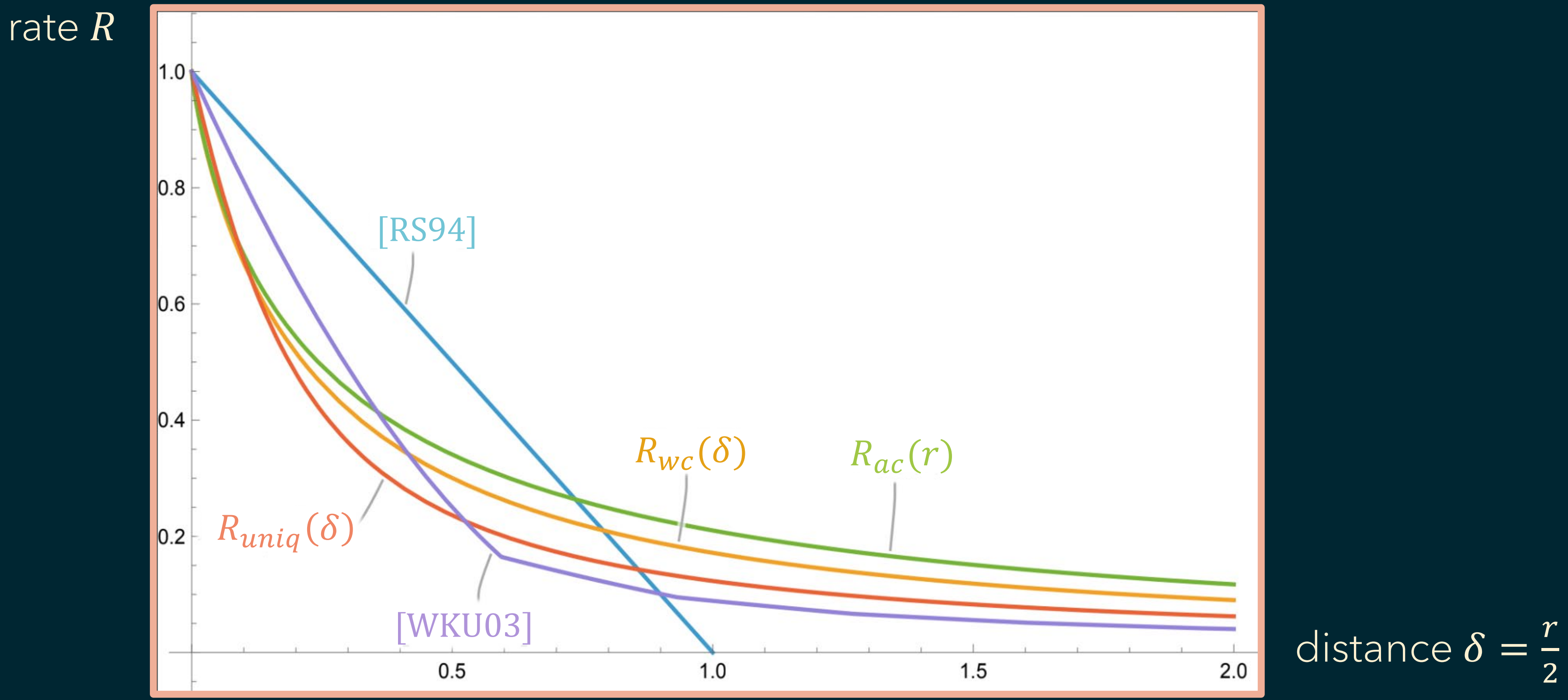
Work	Metric	Codes	Decoder Type	Error Type	Rel. Decoding Distance ($\boldsymbol{\delta}$)
[Guruswami-Sudan, 1999]	Hamming	GRS	list	discrete	$\leq 1 - \sqrt{R}$
[Mook-Peikert, 2022]	Euclidean (ℓ_2)	prime-field GRS	list	continuous	$\leq \sqrt{(1-R)/2}$
[Roth-Siegel, 1994]	Lee (ℓ_1)	subclass of GRS, BCH	unique	discrete	$\leq 1 - R$
[Wu-Kuijper-Udaya, 2003]	Lee (ℓ_1)	prime-field GRS	list	discrete	$\leq g(R)$ piecewise
[Peikert-V. Hostetler, 2025]	any ℓ_p, $0 < p \leq 2$	prime-field GRS	list	continuous	$\leq 1/(R \cdot c_p(e\,p)^{1/p})$

Prior Algorithms

Work	Metric	Codes	Decoder Type	Error Type	Rel. Decoding Distance (δ)
[Guruswami-Sudan, 1999]	Hamming	GRS	list	discrete	$\leq 1 - \sqrt{R}$
[Mook-Peikert, 2022]	Euclidean (ℓ_2)	prime-field GRS	list	continuous	$\leq \sqrt{(1-R)/2}$
[Roth-Siegel, 1994]	Lee (ℓ_1)	subclass of GRS, BCH	unique	discrete	$\leq 1 - R$
[Wu-Kuijper-Udaya, 2003]	Lee (ℓ_1)	prime-field GRS	list	discrete	$\leq g(R)$ piecewise
[Peikert-V. Hostetler, 2025]	any ℓ_p, $0 < p \leq 2$	prime-field GRS	list	continuous	$\leq 1/(R \cdot c_p (e\,p)^{1/p})$

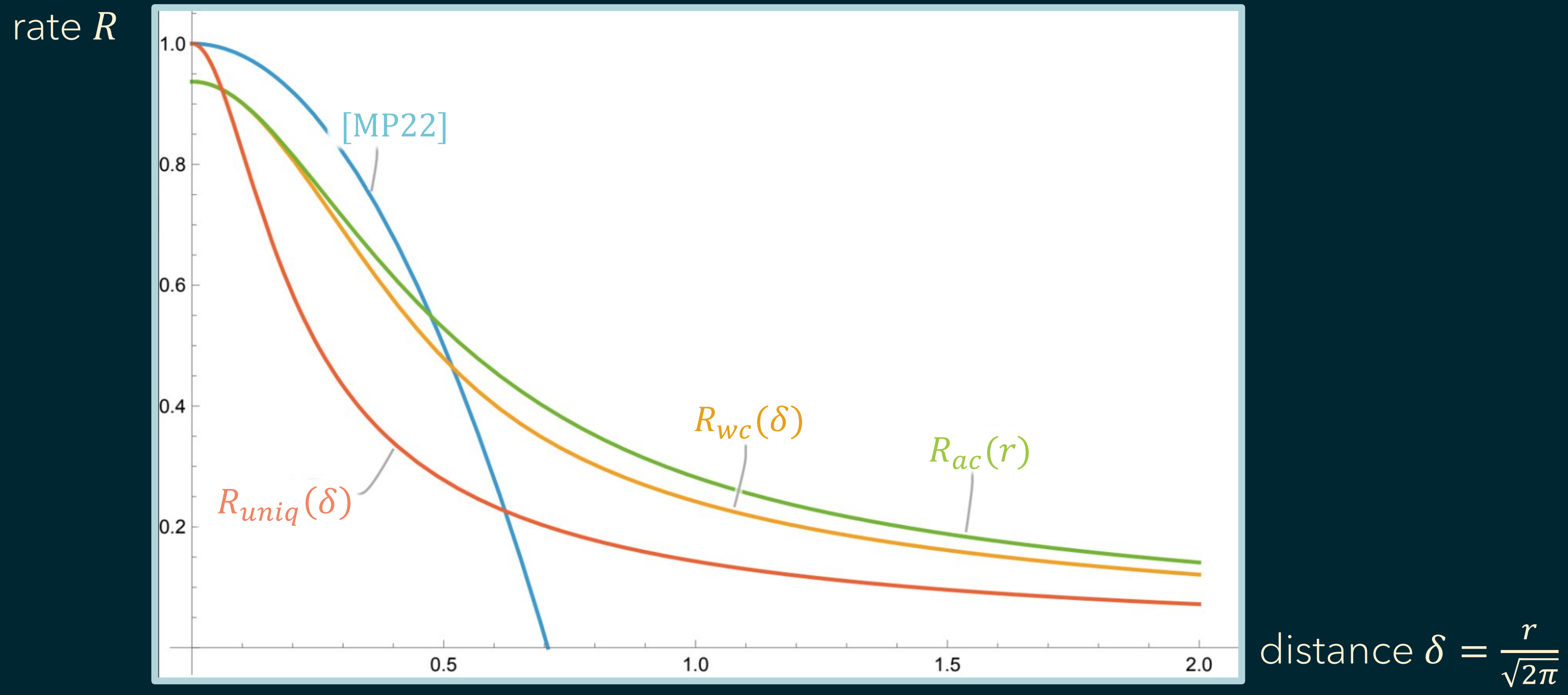
(dimension-normalized) volume of n-dim. ℓ_p ball of radius $n^{1/p}$

Comparison to Prior Algorithms



Rate-distance trade-off for Lee (ℓ_1) metric

Comparison to Prior Algorithms



Rate-distance trade-off for Euclidean (ℓ_2) metric

Our List-decoding Algorithm

received word
$\boldsymbol{y} \in \mathbb{R}_q^n$

transform $\boldsymbol{y}$ into
weight vector $\boldsymbol{W} \in [0,1]^{qn}$

GS Algorithm

codewords correlated with $\boldsymbol{W}$
$\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_L \in \mathcal{C}$

codewords close to $\boldsymbol{y}$
$\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_L \in \mathcal{C}$

Guruswami-Sudan Algorithm

[Guruswami-Sudan, 1998], [Koetter-Vardy, 2003], [Guruswami, 2001]

There is a deterministic *soft-decoding* algorithm for (Generalized) Reed-Solomon codes $\mathcal{C} \subseteq \mathbb{F}_q^n$,

dimension k, adjusted rate $R^* = \frac{k-1}{n}$, with

Input: weight vector $\mathrm{W} = (\overrightarrow{\mathrm{w}_1}, \ldots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$,

tolerance parameter $\tau > 0$

Output: list of all codewords $\boldsymbol{c} \in \mathcal{C}$ that are "closely correlated" with W

$$\operatorname{corr}(\mathrm{W}, \boldsymbol{c}) := \frac{\langle \mathrm{W}, [\boldsymbol{c}] \rangle}{\|\mathrm{W}\| \cdot \sqrt{n}} \gtrsim \sqrt{R^*}.$$

running in poly $\left(n, q, \frac{1}{\tau \|W\|}\right)$ time.

Guruswami-Sudan Algorithm

[Guruswami-Sudan, 1998], [Koetter-Vardy, 2003], [Guruswami, 2001]

There is a deterministic *soft-decoding* algorithm for (Generalized) Reed-Solomon codes $\mathcal{C} \subseteq \mathbb{F}_q^n$, dimension k, adjusted rate $R^* = \frac{k-1}{n}$, with

Input: weight vector $\mathrm{W} = (\overrightarrow{\mathrm{w}_1}, \dots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$,

tolerance parameter $\boldsymbol{\tau} > 0$

Output: list of all codewords $\boldsymbol{c} \in \mathcal{C}$ that are "closely correlated" with W

$$\mathrm{corr}(\mathrm{W}, \boldsymbol{c}) := \frac{\langle \mathrm{W}, [\boldsymbol{c}] \rangle}{\|\mathrm{W}\| \cdot \sqrt{n}} \geq \sqrt{R^*} + \boldsymbol{\tau}.$$

running in $\mathrm{poly}\left(n, q, \frac{1}{\tau \|W\|}\right)$ time.

Transforming into Weights

received word

$\boldsymbol{y} = \boxed{y_1} \boxed{y_2} \boxed{\dots} \boxed{y_n} \in \mathbb{R}_q^n$

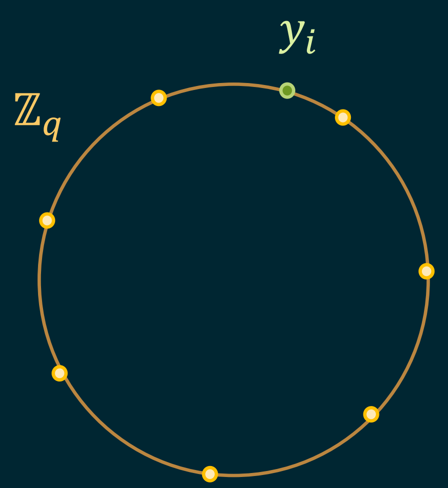
transform $\boldsymbol{y}$ into
weight vector $\boldsymbol{W} \in [0,1]^{qn}$

GS Algorithm

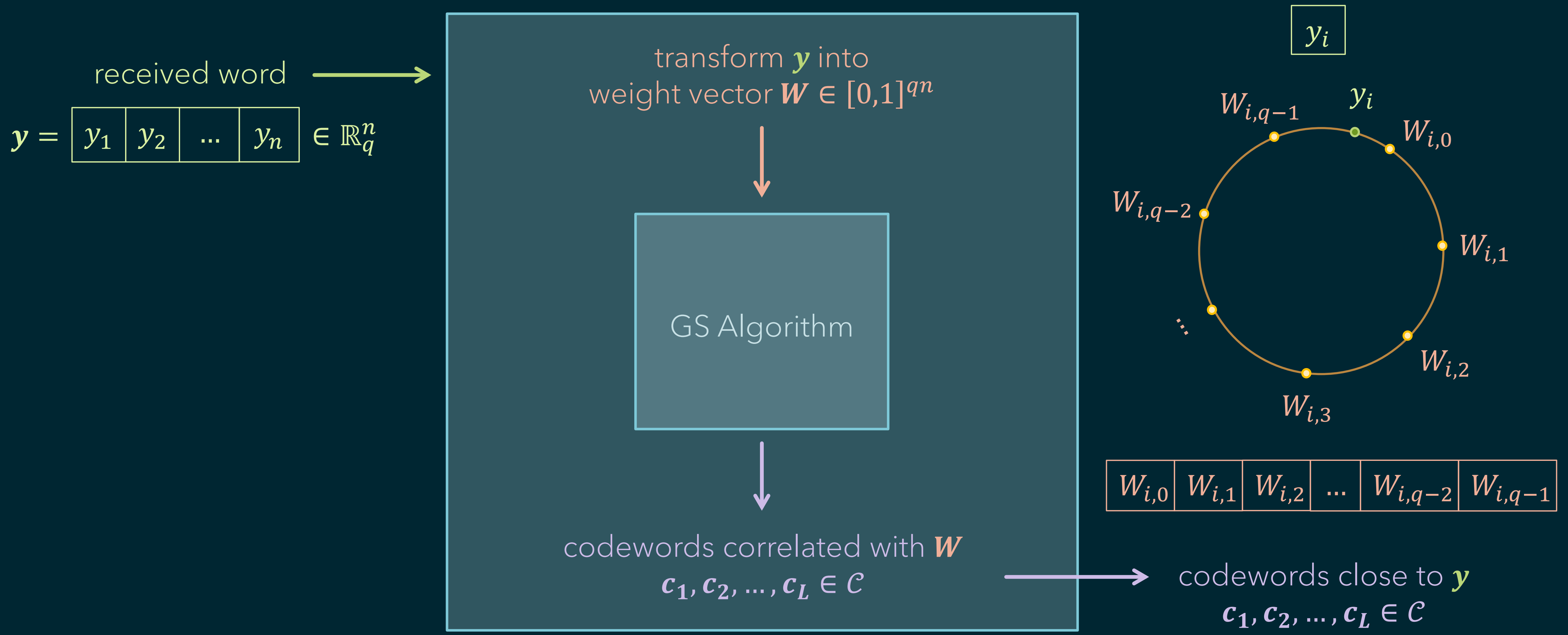
codewords correlated with $\boldsymbol{W}$
$\boldsymbol{c_1}, \boldsymbol{c_2}, \dots, \boldsymbol{c_L} \in \mathcal{C}$

codewords close to $\boldsymbol{y}$
$\boldsymbol{c_1}, \boldsymbol{c_2}, \dots, \boldsymbol{c_L} \in \mathcal{C}$

$\boxed{y_i}$

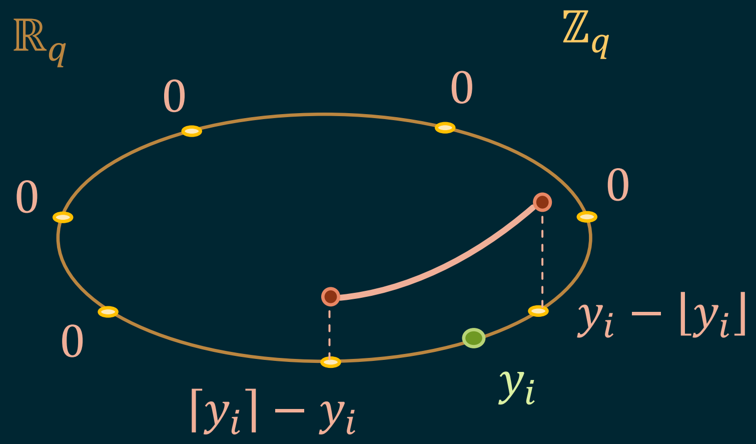


Transforming into Weights



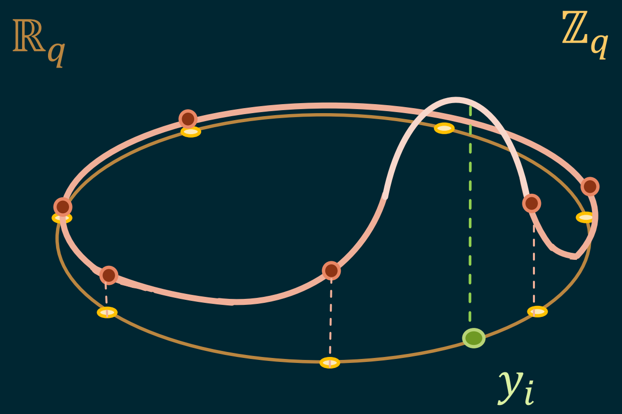
Transforming into Weights

[Mook-Peikert, 2022] :



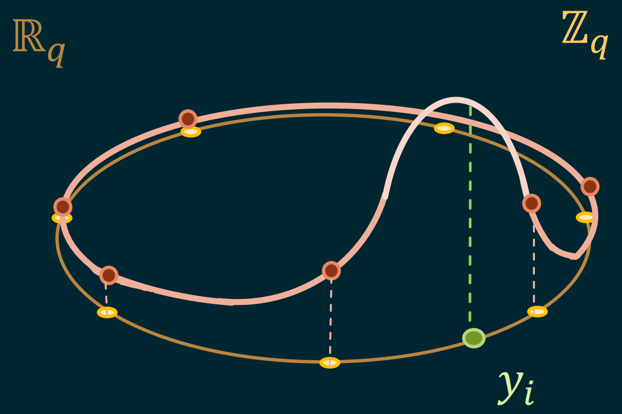
Transforming into Weights

Our weight function :



Transforming into Weights

Our weight function :



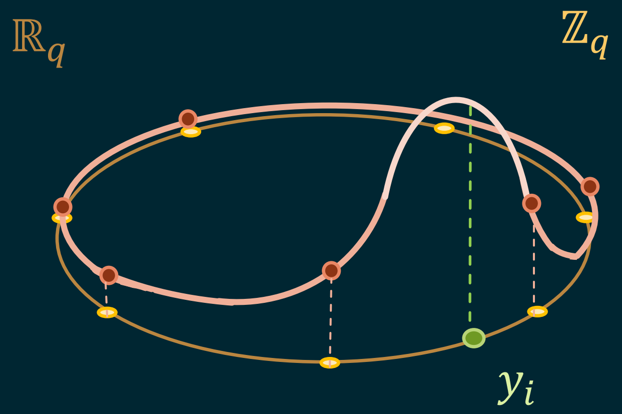
weights given by a

function f_s of width $s > 0$

$$f_s(x) \coloneqq \exp(-(c_p |x/s|)^p)$$

Transforming into Weights

Our weight function :



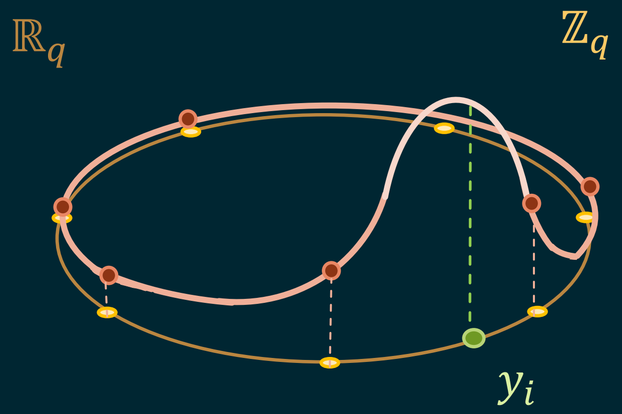
weights given by a function f_s of width $s > 0$

$$f_s(x) := \exp(-(\underbrace{c_p}_{=\, 2 \cdot \Gamma\left(1 + \frac{1}{p}\right)}|x/s|)^p)$$

makes $\widehat{f_s(0)} := 1$

Transforming into Weights

Our weight function :



weights given by a function f_s of width $s > 0$

$$f_s(x) := \exp(-(c_p|x/s|)^p)$$

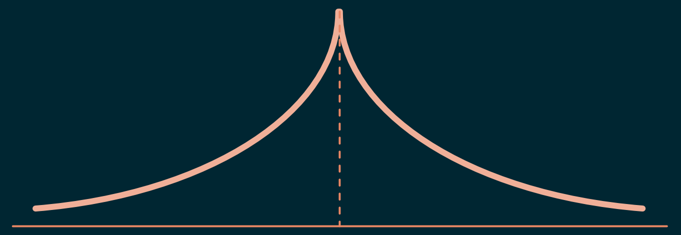
We cannot handle $p > 2$ because $\widehat{f_s(x)}$ can be negative.

Our Weight Function

For distances measured in the Lee (ℓ_1) metric:

$$f_s(x) \coloneqq \exp(-2\,|x/s|^1)$$

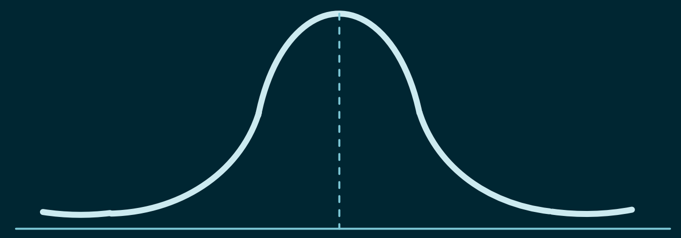
Laplacian function



For distances measured in the Euclidean (ℓ_2) metric:

$$f_s(x) \coloneqq \exp(-\pi\,|x/s|^2)$$

Gaussian function



Our Analysis

received word

$\boldsymbol{y} \in \mathbb{R}_q^n$

transform $\boldsymbol{y}$ into
weight vector $\boldsymbol{W} \in [0,1]^{qn}$

GS Algorithm

codewords correlated with $\boldsymbol{W}$
$\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_L \in \mathcal{C}$

codewords close to $\boldsymbol{y}$
$\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_L \in \mathcal{C}$

We show that any codeword $\boldsymbol{c}$
close enough to $\boldsymbol{y}$, i.e.

$$\text{rel dist}(\boldsymbol{c}, \boldsymbol{y}) \le \delta$$

has large enough correlation
with weight vector $\boldsymbol{W}$, i.e.

$$\text{corr}(\boldsymbol{c}, \boldsymbol{W}) > \sqrt{R}.$$

Our Analysis

received word

$\boldsymbol{y} \in \mathbb{R}_q^n$

transform $\boldsymbol{y}$ into weight vector $\boldsymbol{W} \in [0,1]^{qn}$

GS Algorithm

codewords correlated with $\boldsymbol{W}$

$\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_L \in \mathcal{C}$

We show that any codeword $\boldsymbol{c}$ close enough to $\boldsymbol{y}$, i.e.

$$\text{rel dist}(\boldsymbol{c}, \boldsymbol{y}) \leq \delta$$

has large enough correlation with weight vector $\boldsymbol{W}$, i.e.

$$\text{corr}(\boldsymbol{c}, \boldsymbol{W}) > \sqrt{R}.$$

Then we optimize width s of weight function f_s to maximize R.

codewords close to $\boldsymbol{y}$

$\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_L \in \mathcal{C}$

Our Results

Theorem: *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$\underbrace{R^* <}_{\tau} \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)}$$

$\tau :=$ gap between rate and upper bound

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time.

Our Results

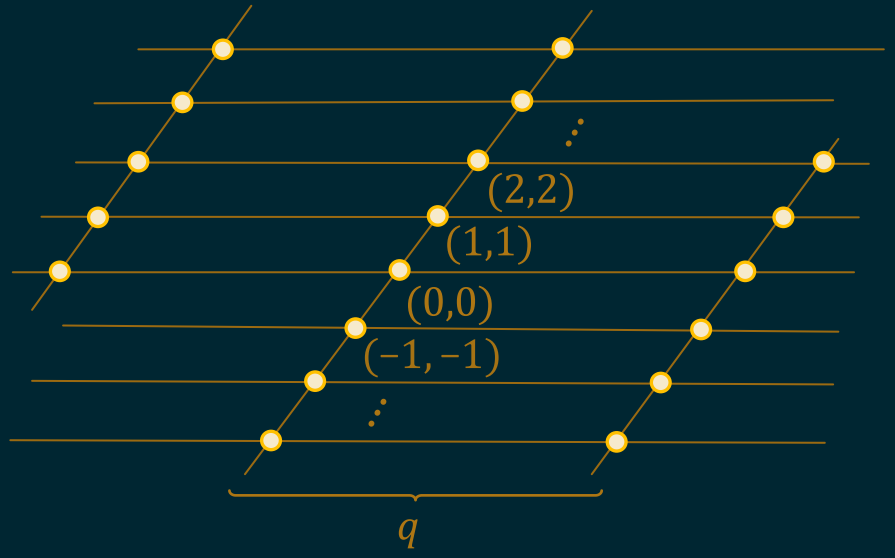
Theorem: *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{\underbrace{f_s(\mathcal{L}_q)}_{\textstyle = \sum_{x \in \mathcal{L}_q} f_s(x)}}$$

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time.



Our Results

<u>Theorem:</u> *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s)$$

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time.

Our Results

Theorem: *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s) \xrightarrow{s,q/s \to \infty} \frac{1}{\delta \cdot c_p \cdot (ep)^{1/p}}$$

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time.

Our Results

<u>Theorem:</u> *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s) \xrightarrow{s,\, q/s \to \infty} \frac{1}{\delta \cdot c_p \cdot (ep)^{1/p}}$$

$\parallel$

volume of the n-dim. ℓ_p ball of radius $n^{1/p}$
(dimension-normalized)

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time.

Our Results

Theorem: *(average-case)* For any metric $0 < p \leq 2, \alpha \in (0,1)$, prime q, channel parameter $r > 0$,

under a memoryless, additive (continuous or discrete) channel whose distribution is D_r,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$pdf\colon\ D_r(x) = f_r(x)/r$
$pmf\colon D_r(x) = f_r(x)/f_r(\mathbb{Z})$

$$R^* < \frac{s^2/\|(r,s)\|_p^2}{f_s(\mathcal{L}_q)} =: A_{q,r}^{(p)}(s)$$

in $\mathrm{poly}(n, q, \exp(1/s^p)/\tau)$ time, except with probability $< \exp\left(-2n \cdot f_s\left(\mathcal{L}_q\right) \cdot \alpha^2 \cdot \tau^2\right)$.

Open Directions

- The product of the rate R and distance δ for which our algorithm works approaches

  $R \cdot \delta \longrightarrow$ relative radius of a unit-volume n-dim. ℓ_p ball.

  Why should this be the case?

- What is the list-decoding capacity for decoding over general ℓ_p norms?

  How do our algorithmic bounds compare?

- Prove a better lower bound on the minimum distance for *all* GRS codes, not just a natural subclass.

- Extend our results to folded Reed-Solomon (FRS) codes.

  Can we achieve a better rate-distance trade-off?

Open Directions

- The product of the rate R and distance δ for which our algorithm works approaches

$$R \cdot \delta \longrightarrow \text{relative radius of a unit-volume } n\text{-dim. } \ell_p \text{ ball.}$$

  Why should this be the case?

- What is the list-decoding capacity for decoding over general ℓ_p norms?

  How do our algorithmic bounds compare?

- Prove a better lower bound on the minimum distance for *all* GRS codes, not just a natural subclass.

- Extend our results to folded Reed-Solomon (FRS) codes.

  Can we achieve a better rate-distance trade-off?

Open Directions

- The product of the rate R and distance δ for which our algorithm works approaches

$$R \cdot \delta \longrightarrow \text{relative radius of a unit-volume } n\text{-dim. } \ell_p \text{ ball.}$$

  Why should this be the case?

- What is the list-decoding capacity for decoding over general ℓ_p norms?

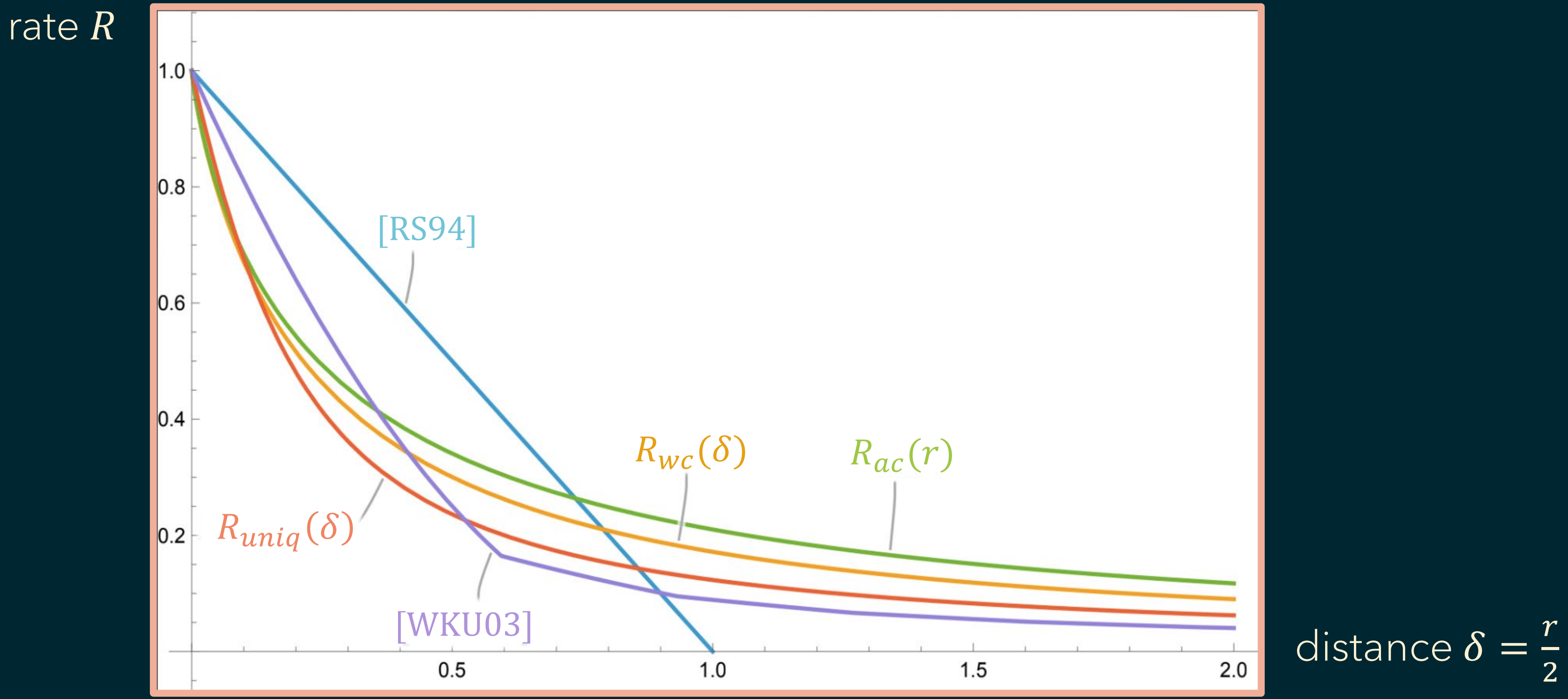
  How do our algorithmic bounds compare?

- Prove a better lower bound on the minimum distance for *all* GRS codes, not just a natural subclass.

- Extend our results to folded Reed-Solomon (FRS) codes.

  Can we achieve a better rate-distance trade-off?

Comparison to Prior Algorithms



Rate-distance trade-off for Lee (ℓ_1) metric

Open Directions

- The product of the rate R and distance δ for which our algorithm works approaches

  $$R \cdot \delta \;\longrightarrow\; \text{relative radius of a unit-volume } n\text{-dim. } \ell_p \text{ ball.}$$

  Why should this be the case?

- What is the list-decoding capacity for decoding over general ℓ_p norms?

  How do our algorithmic bounds compare?

- Prove a better lower bound on the minimum distance for *all* GRS codes, not just a natural subclass.

- Extend our results to folded Reed-Solomon (FRS) codes.

  Can we achieve a better rate-distance trade-off?

Questions?