Lattices and Their Applications to Wireless Communications

Brian M. Kurkoski
Japan Advanced Institute of Science and Technology

JAIST
JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
1990

September 5, 2018
IT Ken
Morioka, Iwate, Japan

Sabbatical FY 2017

Universities visited:

- June 2017 Technion (Israel Institute of Technology) — Eitan Yaakobi
- October 2017 Texas A&M University (USA) — Krishna Narayanan
- March 2018 Monash University (Melbourne, Australia) — Emanuele Viterbo
- March–May, July–Sept, Nov–Dec 2017 Oregon State University (USA) — Bella Bose

Impressions of my sabbatical:

Excellent time to "clear the mind." Stimulate research progress by finding new ideas.

Giving presentation at university is good way to meet people.

Visiting many universities was a chance to get many new ideas.

But, not enough time to complete a paper etc.

Continued to advise 3 PhD students and 2 masters students remotely.

“Introduction to Lattice Coding Theory”

At the universities I visited, I gave multiple lectures on “Lattice Coding Theory”

The target audience is first-year graduate students who have already studied coding theory.

Today’s presentation summarizes those lectures.

B. M. Kurkoski, “Introduction to lattice coding theory.” ECSE Seminar, Monash University, March 2018. 15, 22 and 29 March 2018.

Contributors

Mohammad Nur Hasan

Fan Zhou

Siyu Chen

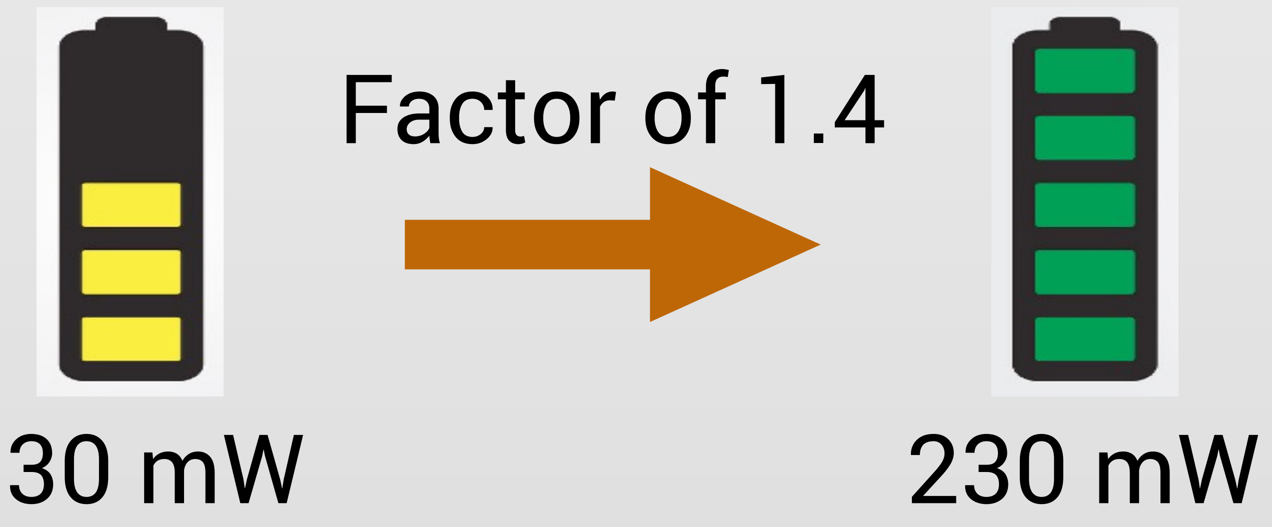
Lattices and Their Applications to Wireless Communications

Central question: How might lattices effectively be used in wireless communication systems?

1. Lattice shaping is a practical way to gain 1.53 dB in SNR

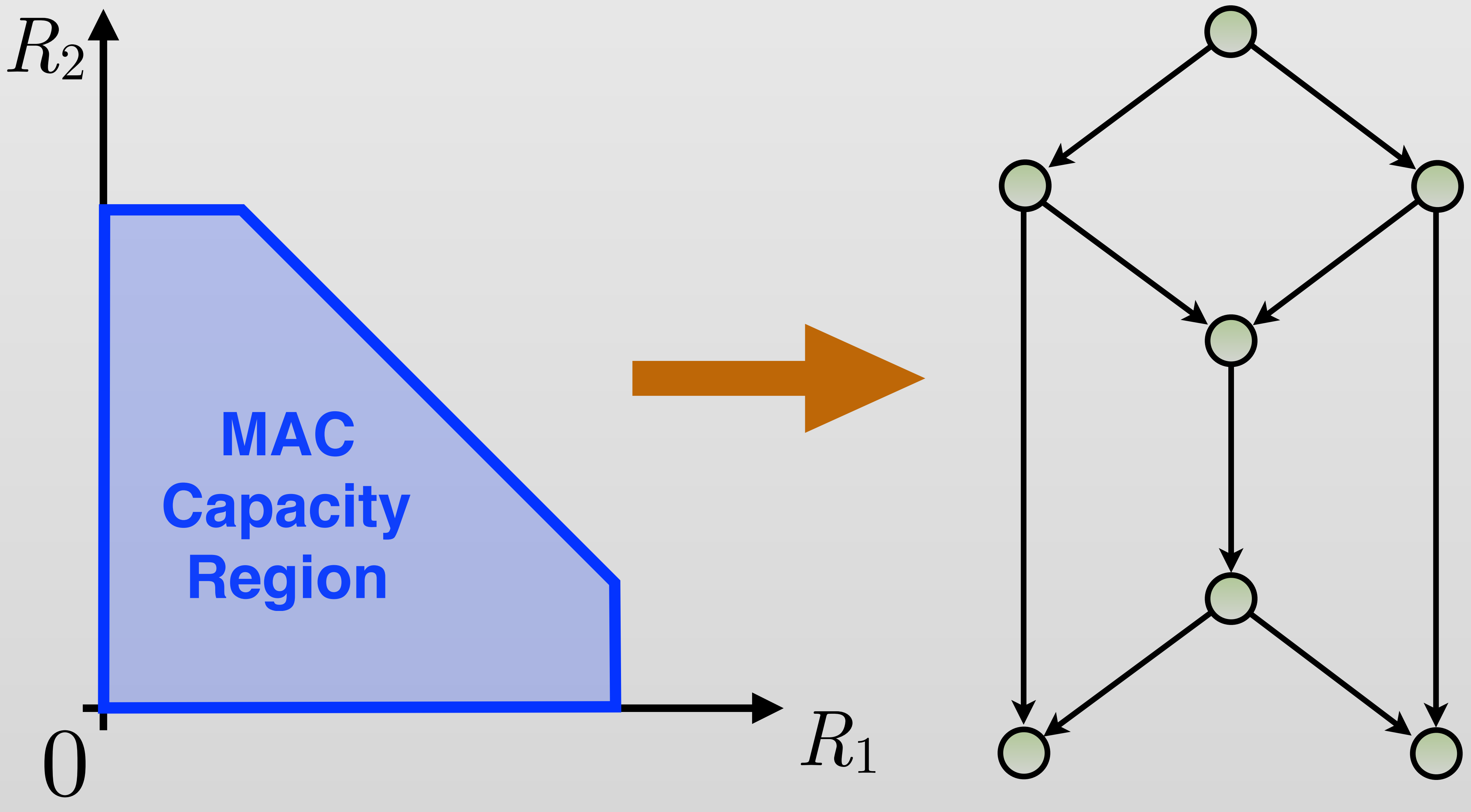
2. Lattice-based physical layer network coding brings benefits of network coding to wireless communications

How much is 1.53 dB?



Significant reduction in transmit power:
- Smartphone battery lasts longer, efficient base stations
- Typical smartphone battery is 10000 mW-hour

From MAC to Wireless Networks



Outline of Semi-Tutorial

1. Introduction to Lattices

- Tutorial and background on lattices

2. Lattices from Construction D and D'

- Form lattices from binary codes
- Since binary codes are well understood, promising candidate for practical lattices
- Lattices based on quasi-cyclic LDPC codes

3. Nested Lattices Codes for the AWGN Channel

- Classify nested lattice codes. Lattices with inflated lattice decoding achieve capacity
- Convolutional code lattices with good shaping gain

4. Physical Layer Network Coding

- Compute-Forward: Network coding when wireless signals add over the air
- Two channels: Bidirectional relay channel and the multiple access relay channel (MARC)

Lattice Definition

Definition 1 An n-dimensional *lattice* Λ is a discrete additive subgroup of $\mathbb{R}^n$.

Intuition A lattice is an error-correcting code defined on the real numbers (rather than a finite field)

Lattice Definition

Definition 1 An n-dimensional *lattice* Λ is a discrete additive subgroup of $\mathbb{R}^n$.

Vector addition in $\mathbb{R}^n$:

$$\begin{aligned}\mathbf{x} &= [x_1, \ldots, x_n] \\ \mathbf{y} &= [y_1, \ldots, y_n] \\ \mathbf{x}+\mathbf{y} &= [x_1+y_1, \ldots, x_n+y_n]\end{aligned}$$

Group properties:

- has identity
- has inverse
- associative
- closure
- (commutative)

Lattices in $\mathbb{R}^2$

Lattices in $\mathbb{R}^2$

Lattices in $\mathbb{R}^2$

Lattices in $\mathbb{R}^2$

Lattices in $\mathbb{R}^2$

Lattices in $\mathbb{R}^2$

Lattice Generator Matrix

The n-by-n generator matrix G is:

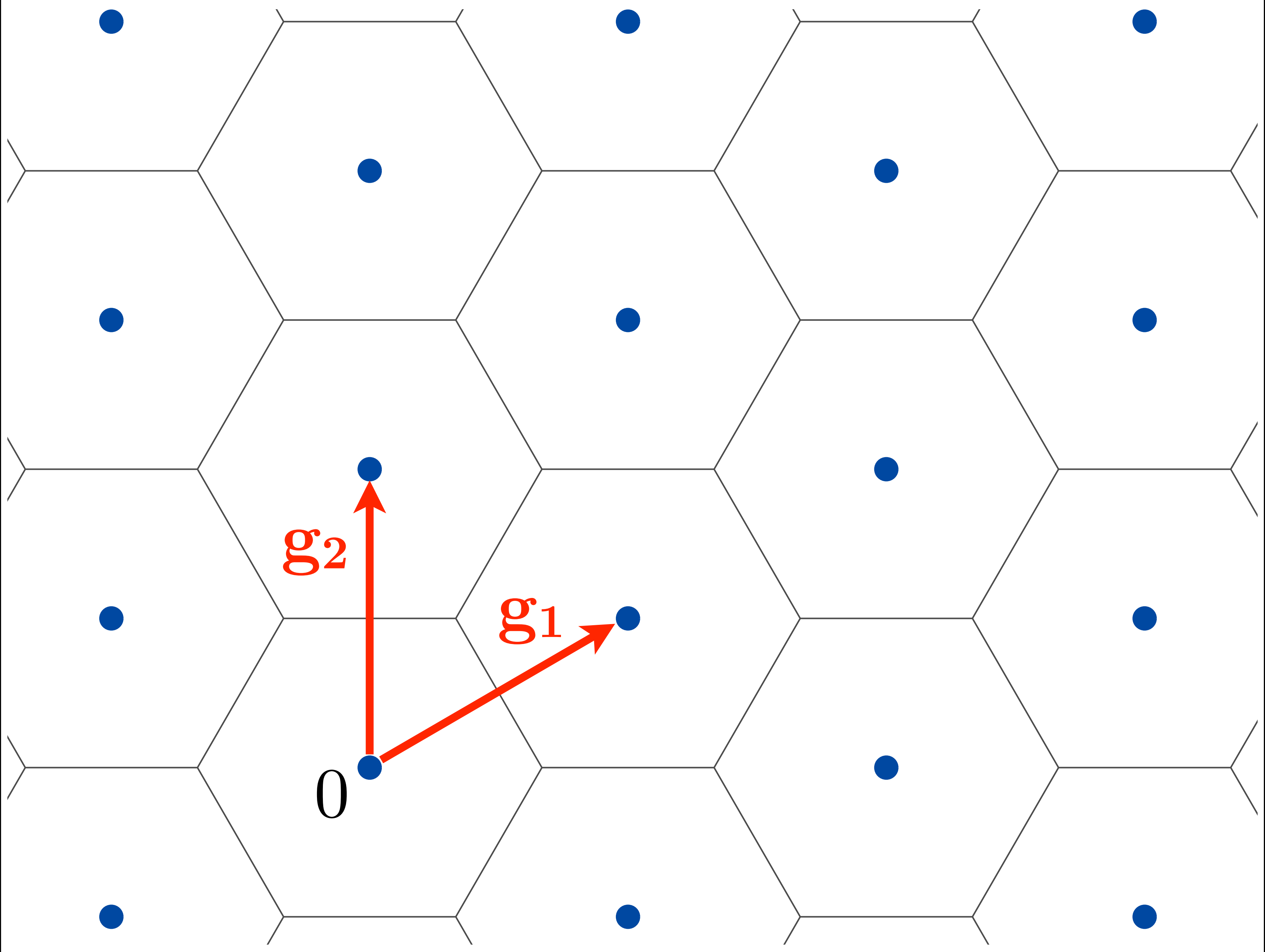
$$G = \begin{bmatrix} | & | & & | \\ \mathbf{g}_1 & \mathbf{g}_2 & \cdots & \mathbf{g}_n \\ | & | & & | \end{bmatrix}$$

so that:

$$\mathbf{x} = \mathbf{G} \cdot \mathbf{b}$$

where $\mathbf{b} \in \mathbb{Z}^n$ is a vector of integers.

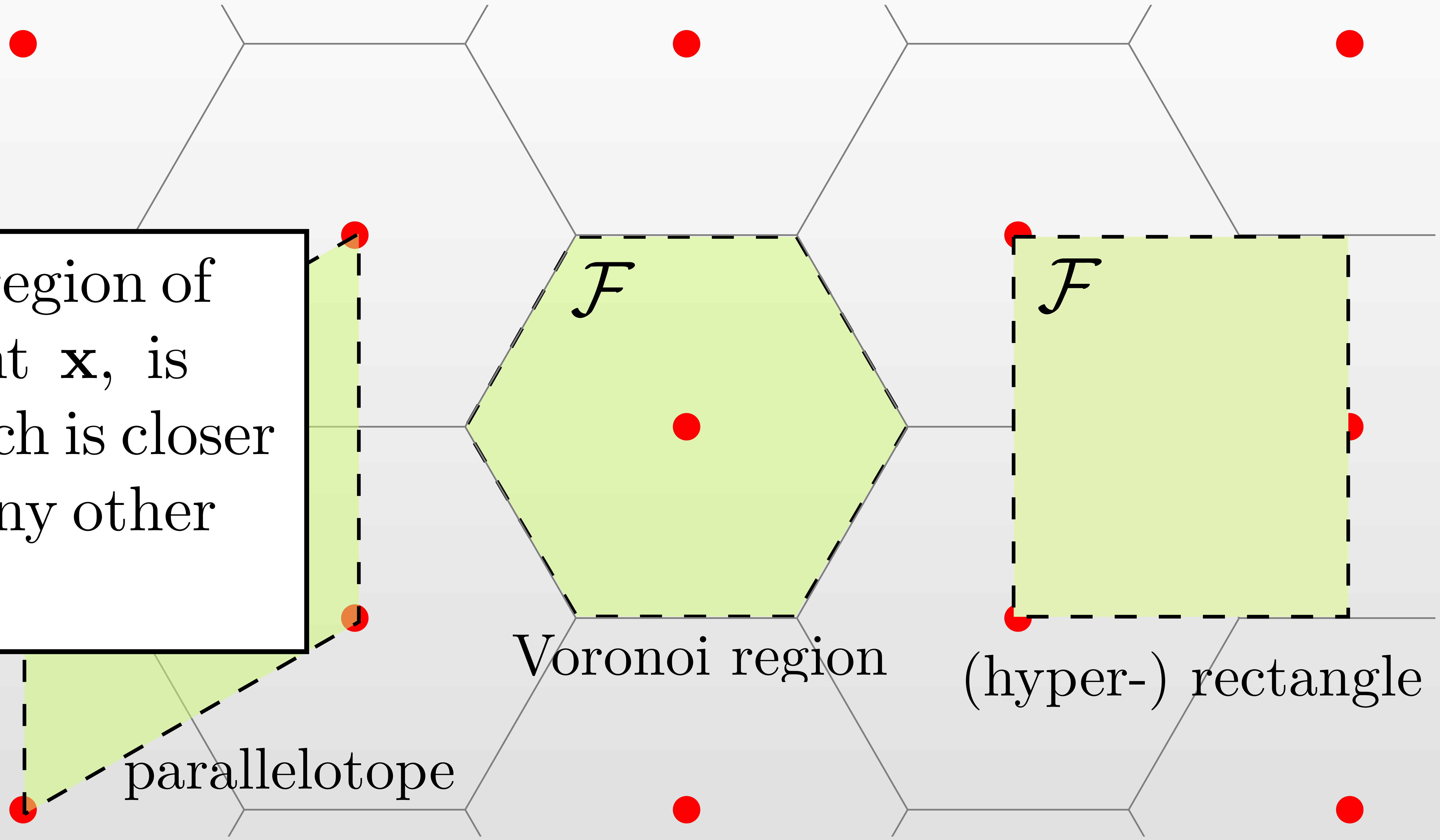
Example



$$\mathbf{G} = \begin{array}{c} \begin{bmatrix} \frac{\sqrt{3}}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix} \\ \quad\ \mathbf{g}_1 \quad \mathbf{g}_2 \end{array}$$

Fundamental Region

The Voronoi region of a lattice point $\mathbf{x}$, is the space which is closer to $\mathbf{x}$ than to any other lattice point.



A fundamental region $\mathcal{F} \subset \mathbb{R}^n$ is a shape that, if shifted by each lattice point, will exactly cover the whole real space.

Volume of $\mathcal{F}$ is $V(\Lambda) = |\det \mathbf{G}|$, and is a constant.

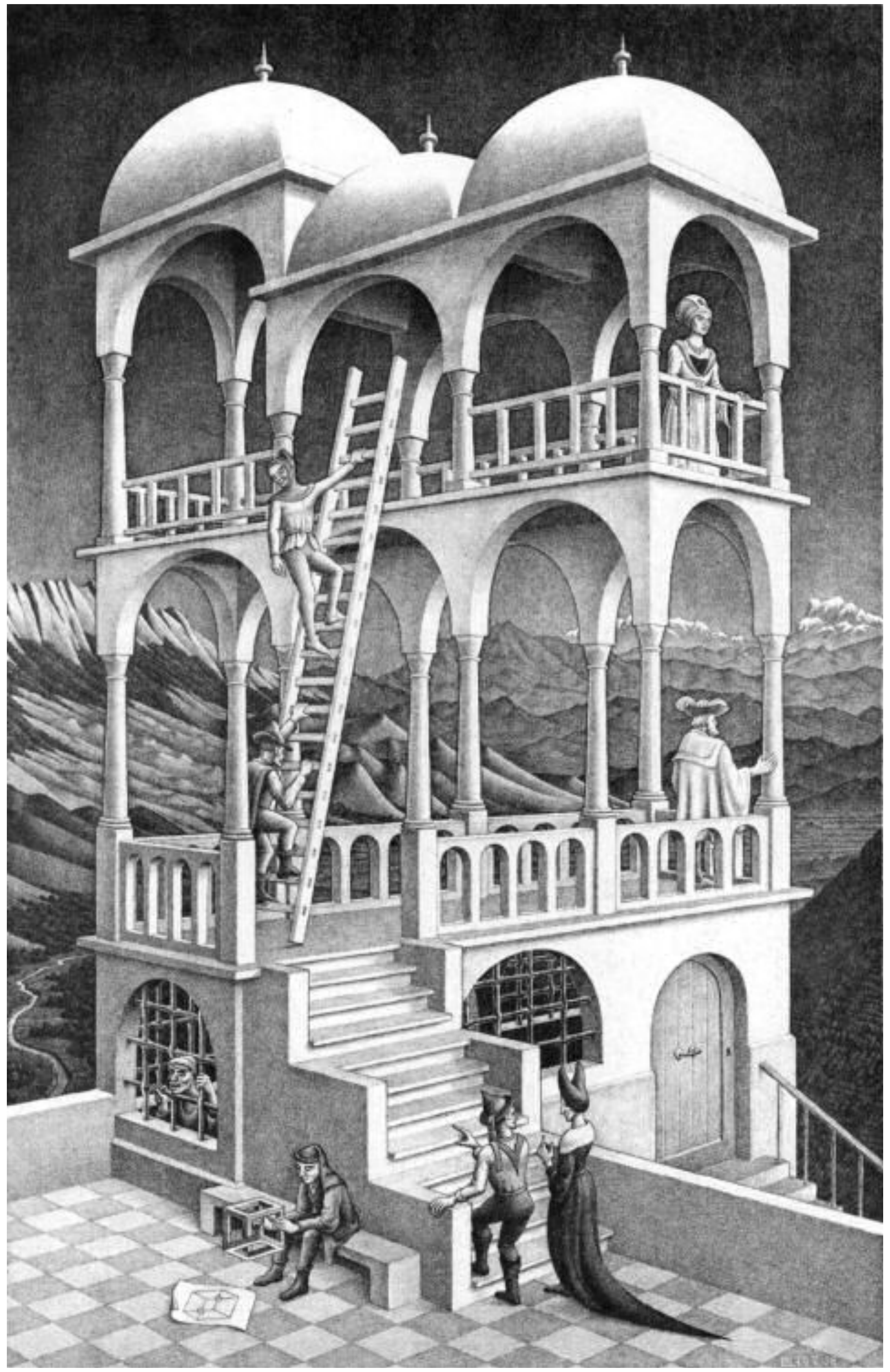
$$\mathbf{G} = \begin{bmatrix} 0.6 & 0 \\ -0.3 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0.6 & 0 \\ -0.3 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 0.6 & 0 \\ -0.3 & 1 \end{bmatrix}$$


$\mathbf{G} = \begin{bmatrix} 0.6 & 0 \\ -0.3 & 1 \end{bmatrix}$

$$\mathbf{G} = \begin{bmatrix} 0.6 & 0 \\ -0.3 & 1 \end{bmatrix}$$

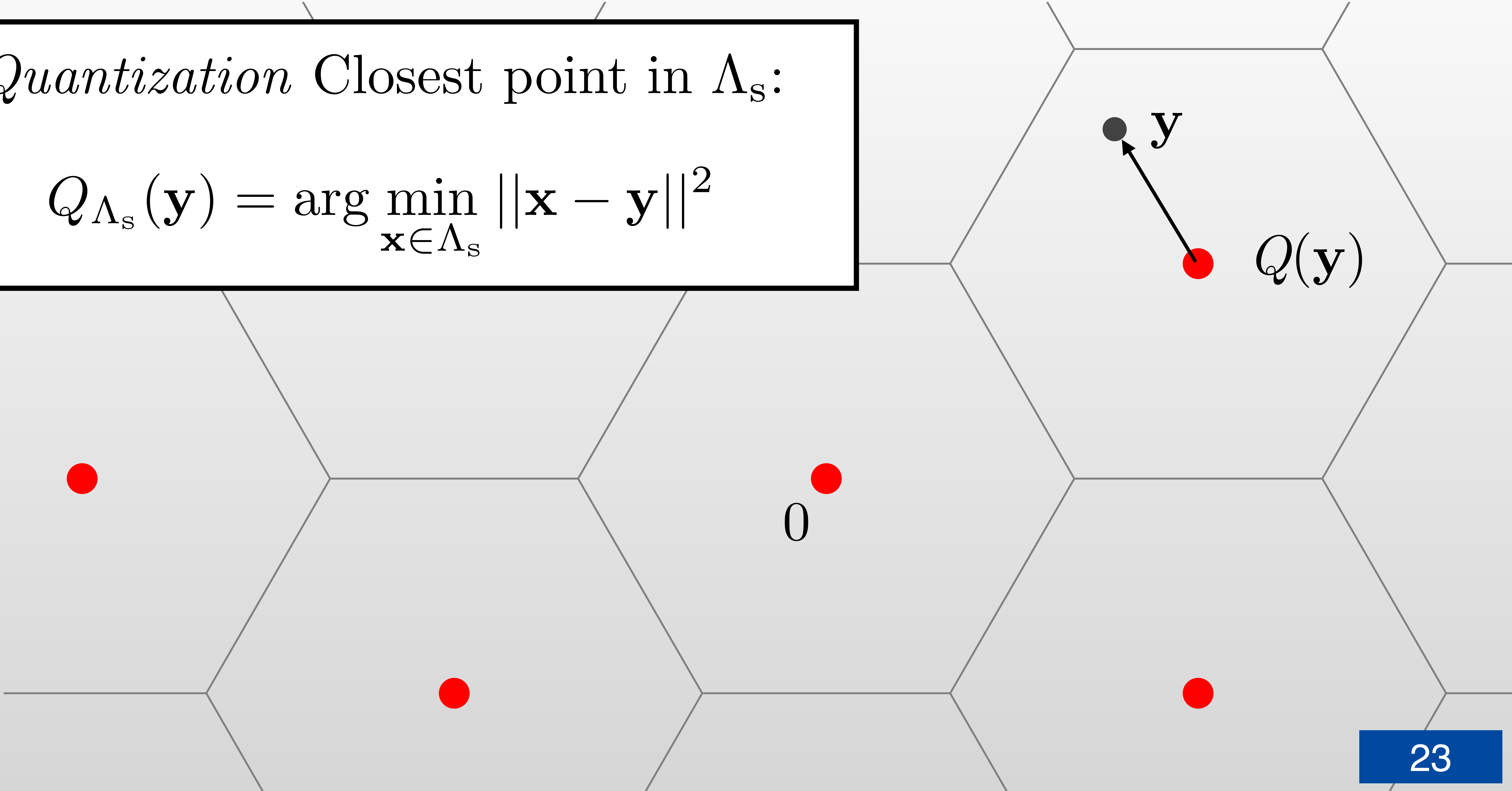


M C Escher
マウリッツ・エッシャー

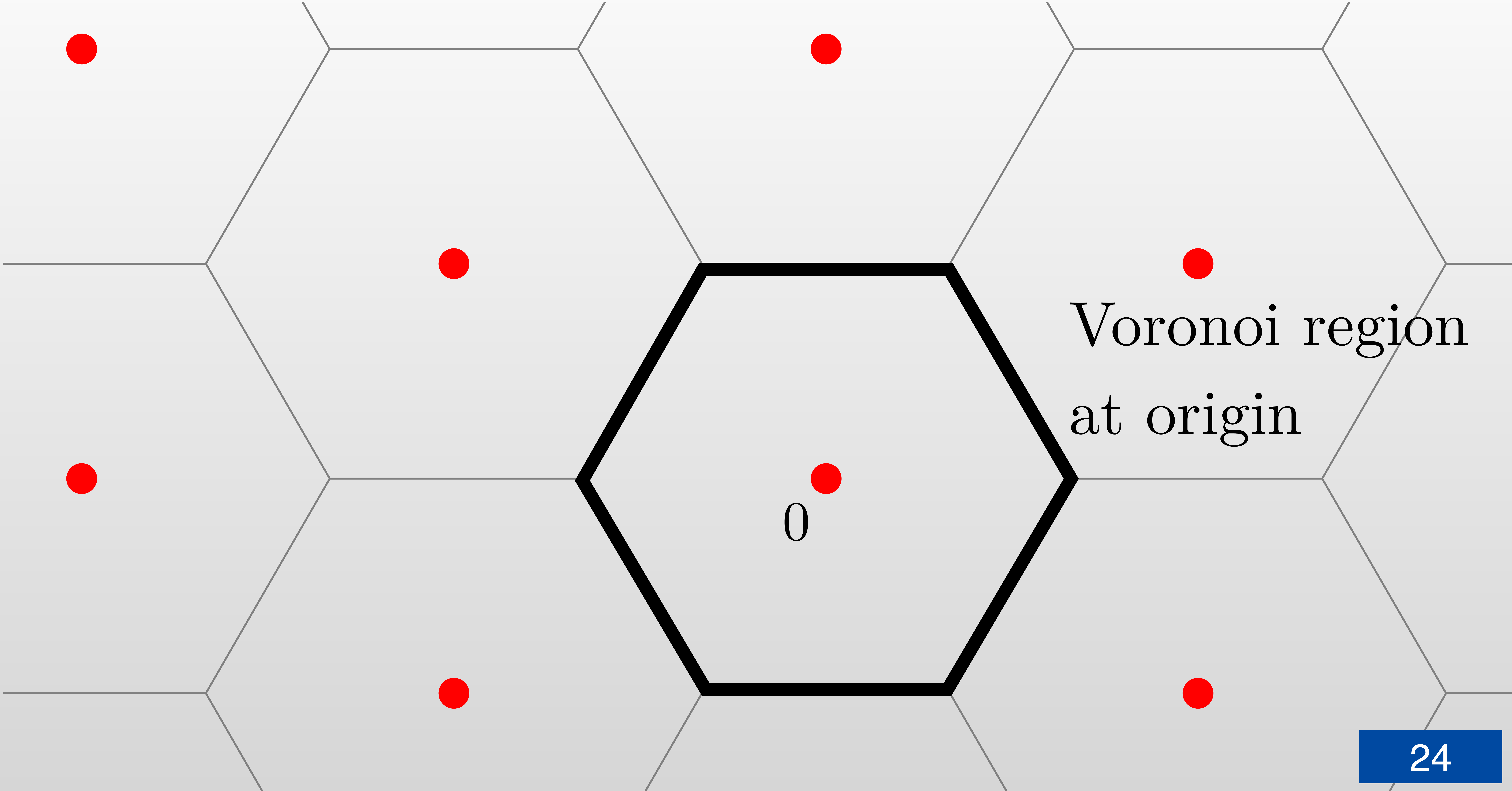
Quantization and Modulo

Quantization Closest point in Λ_s:

$$Q_{\Lambda_\mathrm{s}}(\mathbf{y}) = \arg\min_{\mathbf{x}\in\Lambda_\mathrm{s}} ||\mathbf{x} - \mathbf{y}||^2$$



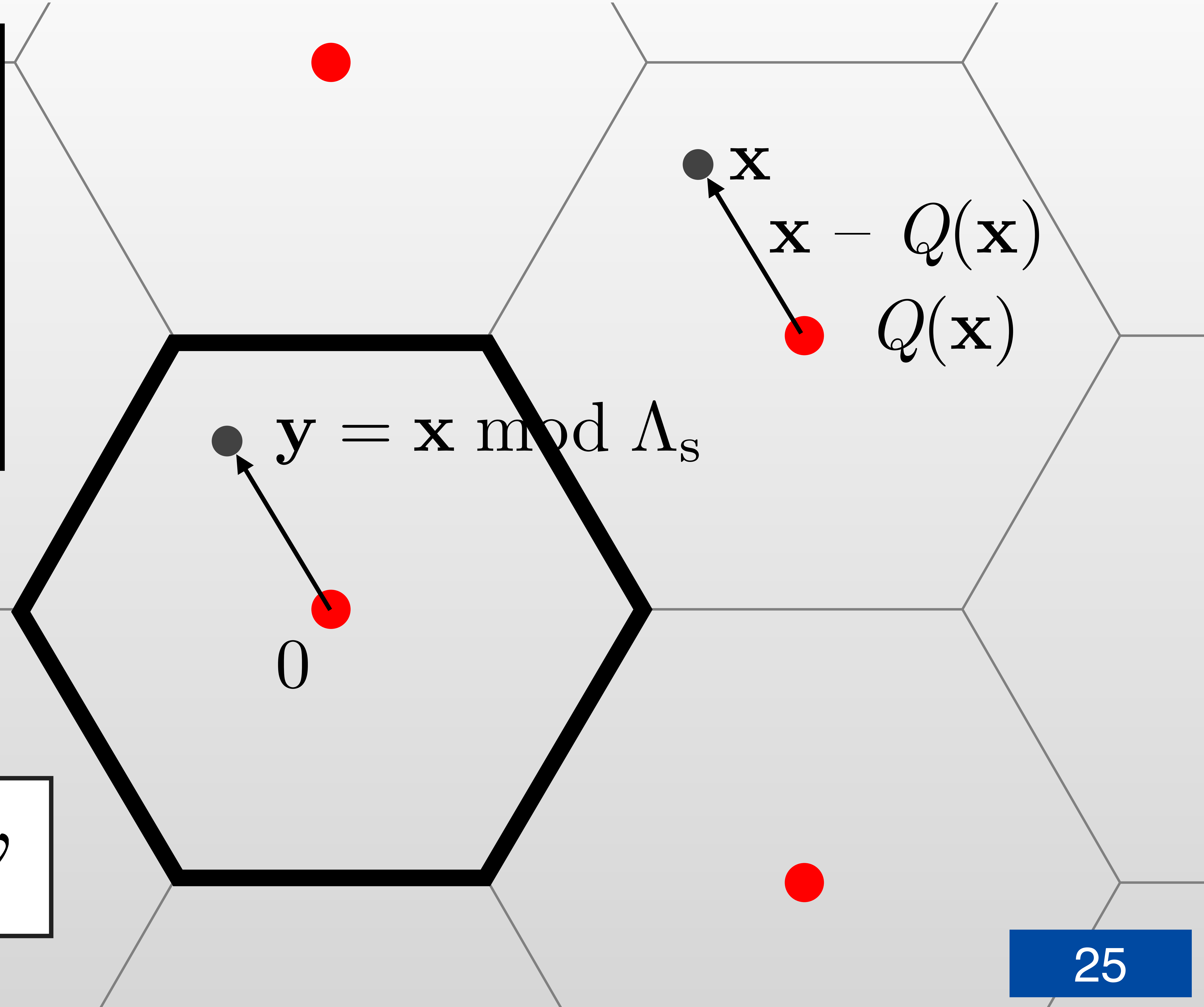
Quantization and Modulo



Quantization and Modulo

Modulo operation:

$$\begin{aligned}\mathbf{y} &= \mathbf{x} \bmod \Lambda_{\mathrm{s}} \\ &= \mathbf{x} - Q_{\Lambda_{\mathrm{s}}}(\mathbf{x})\end{aligned}$$



find the image of $\mathbf{x}$ in $\mathcal{V}$

Construction D and Construction D’

Construction D and D’ are methods to construct lattices from binary codes

Many binary codes have lattice counterpart through Construction D or D’:

- Barnes-Wall lattice (from Reed-Muller code)
- LDPC code lattices
- Polar code lattices
- Turbo code lattices

Construction D: Uses binary code’s generator matrix

Construction D’: Uses binary code’s parity-check matrix

Because binary codes are very well studied, Construction D/D’ are the most promising method to construct practical lattices

A Tale of Construction D

Chapter 1 Early Days

Once upon a time, Barnes and Sloane made lattices from binary codes, which they called “Construction D” [CJM 1985]

Soon after that, Forney created the Code Formula construction, to show special lattices can be written as coset codes [IT 1988]

Chapter 2 Glory Days

Many years pass. Invigorated by Zamir’s lattices, Forney shows that the Code Formula Construction achieves capacity & gives multilevel decoding [IT 2000].

Excited by Code Formula decoding, several researchers create new codes from LDPC, turbo and & codes (2006, 2011, 2013). Multilevel decoding is excellent.

All seems well in the kingdom, until…

Chapter 3 Dark Days

It is a dark time for Construction D/D′. Kositwattanarerk and Oggier show that Construction D/D′ and the Code Formula Construction agree only in some special cases [DCC 2014].

Code Formula Construction is not a lattice, generally.

In some papers, LDPC “lattices”, turbo “lattices”, polar “lattices” are valid structures, but multilevel decoding is their Code Formula version.

How to Decode Construction D?

Krishna Narayanan,
Texas A&M Univ

Those previous “lattices” were decoded as Code Formula, not lattices. How to decode Construction D/D’ lattices?

How to decode Construction D is known.

Actually, *you* showed that.

Not clear yet how to decode Construction D’.
(at that time)

Chapter 3 Dark Days

It is a dark time for Construction D/D′. Kositwattanarerk and Oggier show that Construction D/D′ and the Code Formula Construction agree only in some special cases [DCC 2014].

Code Formula Construction is not a lattice, generally.

LDPC “lattices”, turbo “lattices”, polar “lattices” are valid structures, but multilevel decoding is their Code Formula version.

Chapter 4 A New Beginning

Vem, Huang, Narayanan, Pfister make a decoder for Construction D (but not for Construction D’) [ISIT 2014]

Finally a decoder for Construction D′! Branco da Silva and Silva show how to decode lattice based on binary LDPC codes. [ISIT 2018]

And the lattices lived happily ever after.

Construction D': LDPC-like Example

LDPC check matrix 9×12 for two nested codes

$$\widetilde{\mathbf{H}}_0 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$\mathcal{C}_0$

$\mathcal{C}_1$

Construction D′: LDPC-like Example

Lattice check matrix 12×12

$$\mathbf{H} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & {}^1/_2 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & {}^1/_2 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
{}^1/_2 & 0 & 0 & 0 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & {}^1/_2 & 0 & {}^1/_2 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 & 0 \\
{}^1/_2 & 0 & 0 & 0 & 0 & {}^1/_2 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 \\
0 & {}^1/_2 & 0 & {}^1/_2 & 0 & 0 & 0 & 0 & {}^1/_2 & 0 & 0 & 0 \\
{}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 \\
0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 \\
0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4 & 0 & 0 & {}^1/_4
\end{bmatrix}
\quad \mathcal{C}_0 \quad \mathcal{C}_1$$

Two Methods for LDPC Lattice Construction

$$\widetilde{\mathbf{H}}_0 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \mathcal{C}_0 \\ \\ \\ \mathcal{C}_1 \\ \\ \end{array}$$

Problem: Both Code C0 and C1 should have column weight 3. Code C0 should have higher row weight than Code C1. (assuming regular codes)

Solution 1: **Check node splitting** Design code C0 such that linear combination of two rows has no overlaps, and can be used to form rows of higher degree for code C1. Designed using PEG algorithm and extensive simulations [Branco da Silva and Silva]

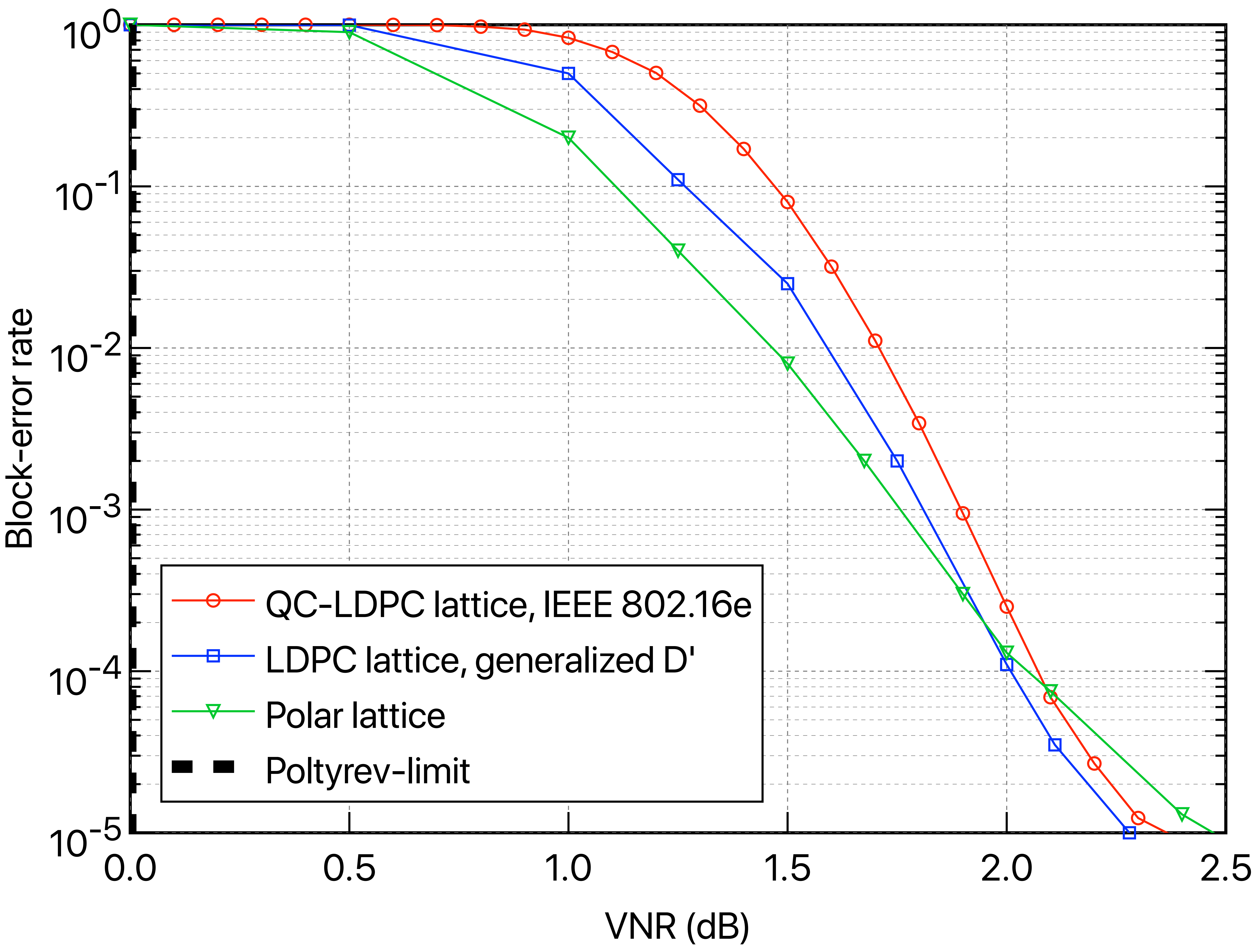
Solution 2: **Minimum distance design** Code C1 should have dmin = 4. Code C0 should have dmin = 16. C1 is a product code of single-parity check codes. C0 is a quasi-cyclic LDPC code from IEEE 802.16e with dmin ≈ 16 [Chen, K, Rosnes]

Error Rate for LDPC Code Lattices

Proposed QC-LPDC code lattices loose about 0.1 dB w.r.t PEG

Minimum distance design rule is a more systematic design approach than PEG/simulations

QC-LDPC codes are widely used in practice. If lattices are to be used in practice, construction D' with QC-LDPC codes are a likely candidate.



S. Chen, B. M. Kurkoski and E. Rosnes, "Construction D' lattices from quasi-cyclic low-density parity-check codes," in 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC'18), (Hong Kong, P. R. China), December 2018

Nested Lattice Codes (Voronoi Codes, Voronoi Constellations)

Definition 1.1. Let Λ_c and Λ_s be two lattices with $\Lambda_\mathrm{s} \subseteq \Lambda_\mathrm{c}$. Let $\mathcal{F}$ be a fundamental region for Λ_s. Then:

$$\mathcal{C} = \Lambda_\mathrm{c} \cap \mathcal{F} \tag{1.1}$$

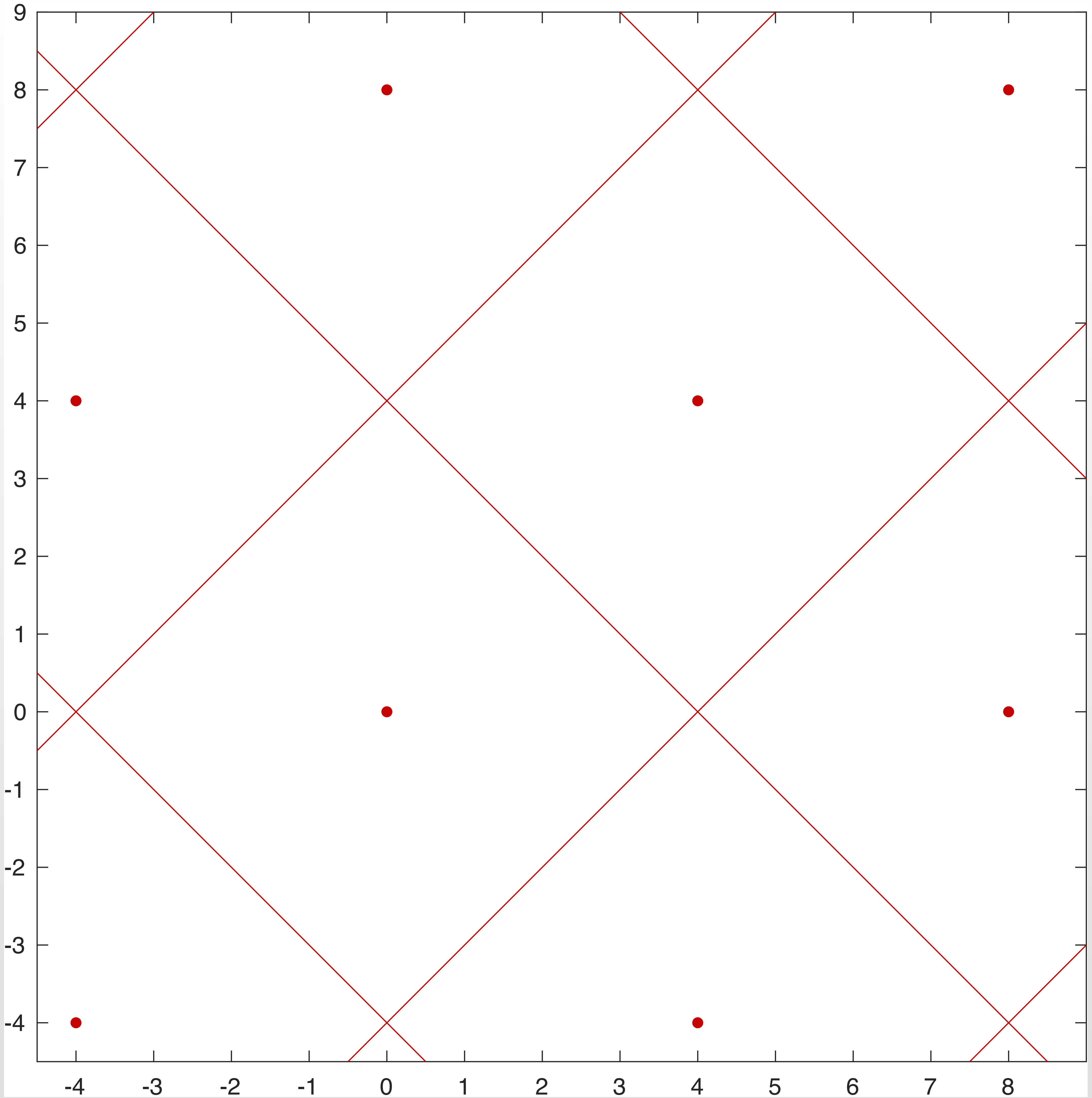
is a *nested lattice code.*

Λ_c is called the coding lattice, Λ_s is called the shaping lattice.

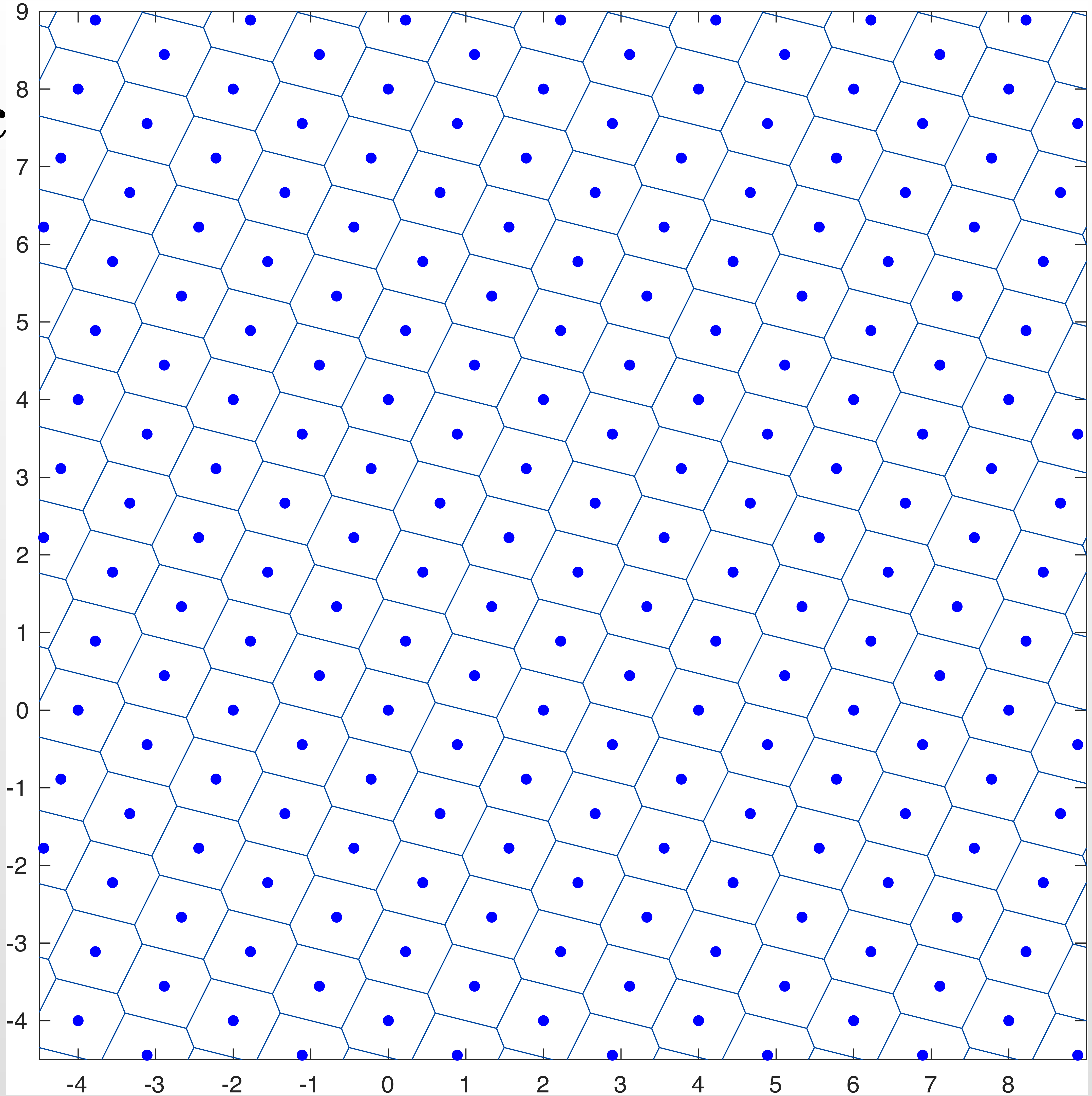
The code rate of a nested lattice code is:

$$R = \frac{1}{n} \log \frac{V(\Lambda_\mathrm{s})}{V(\Lambda_\mathrm{c})} = \frac{1}{n} \log \frac{|\det(\mathbf{G}_\mathrm{s})|}{|\det(\mathbf{G}_\mathrm{c})|}.$$

Λ_{s}



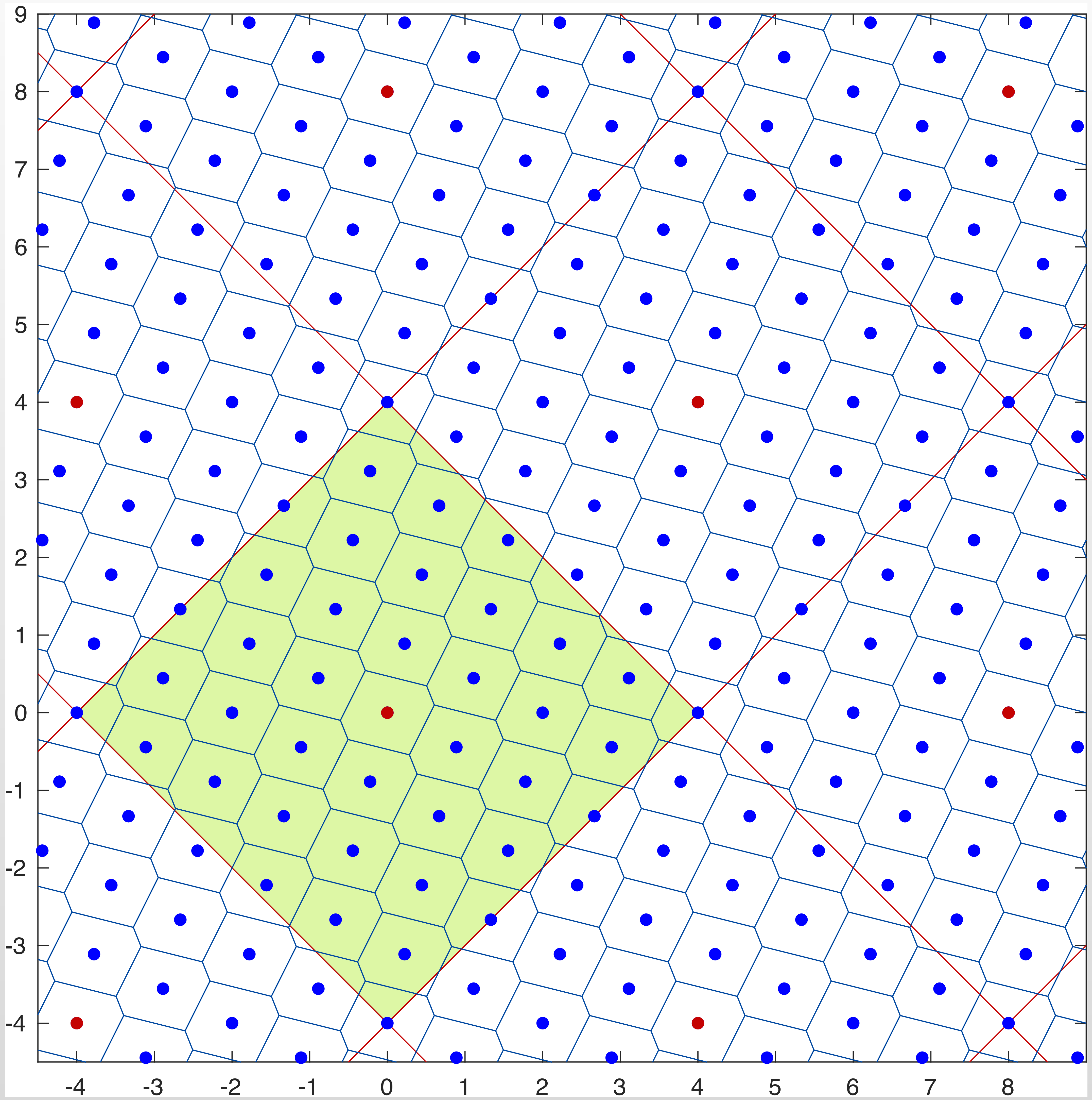
Λ_{c}

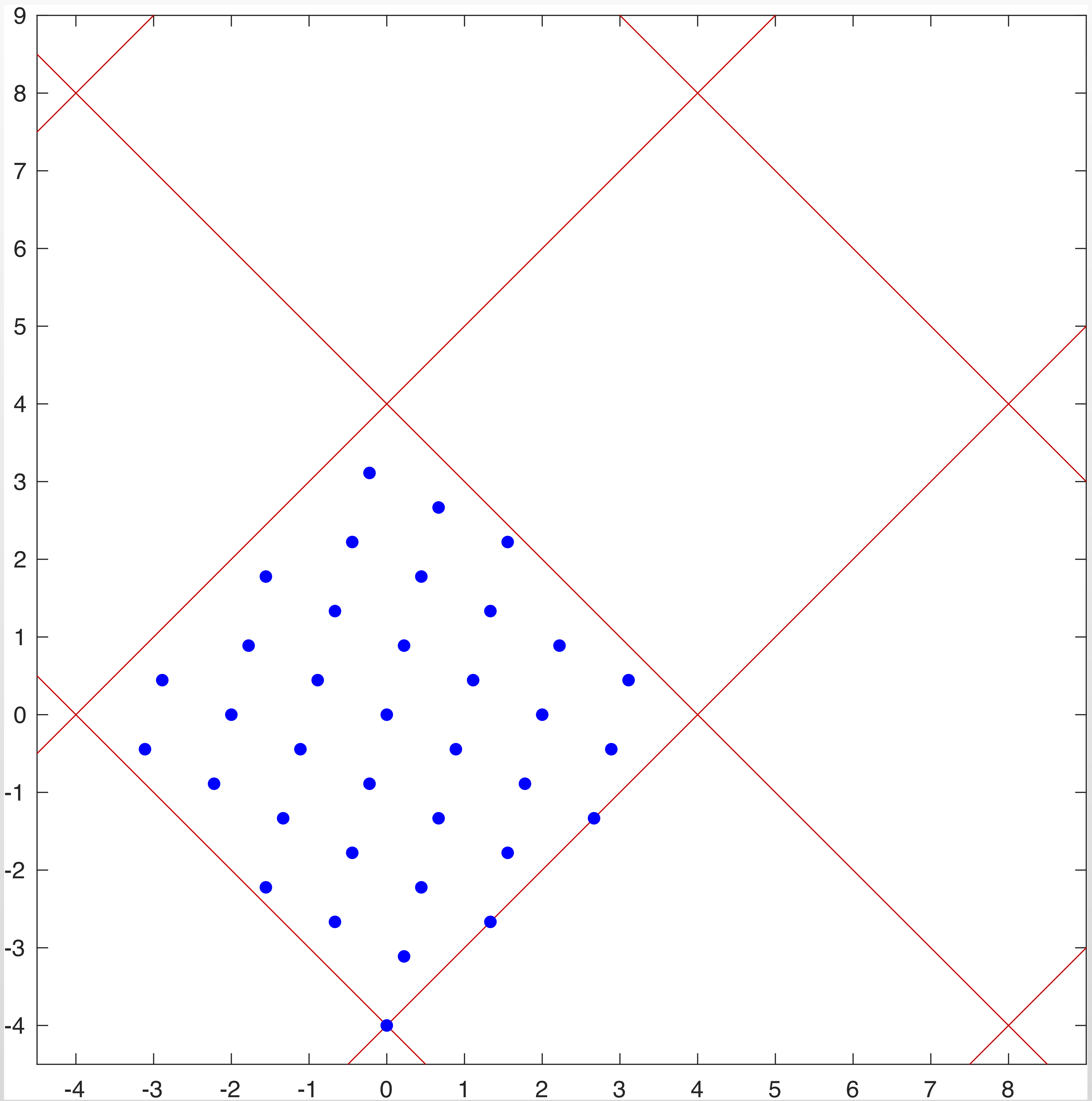


Example

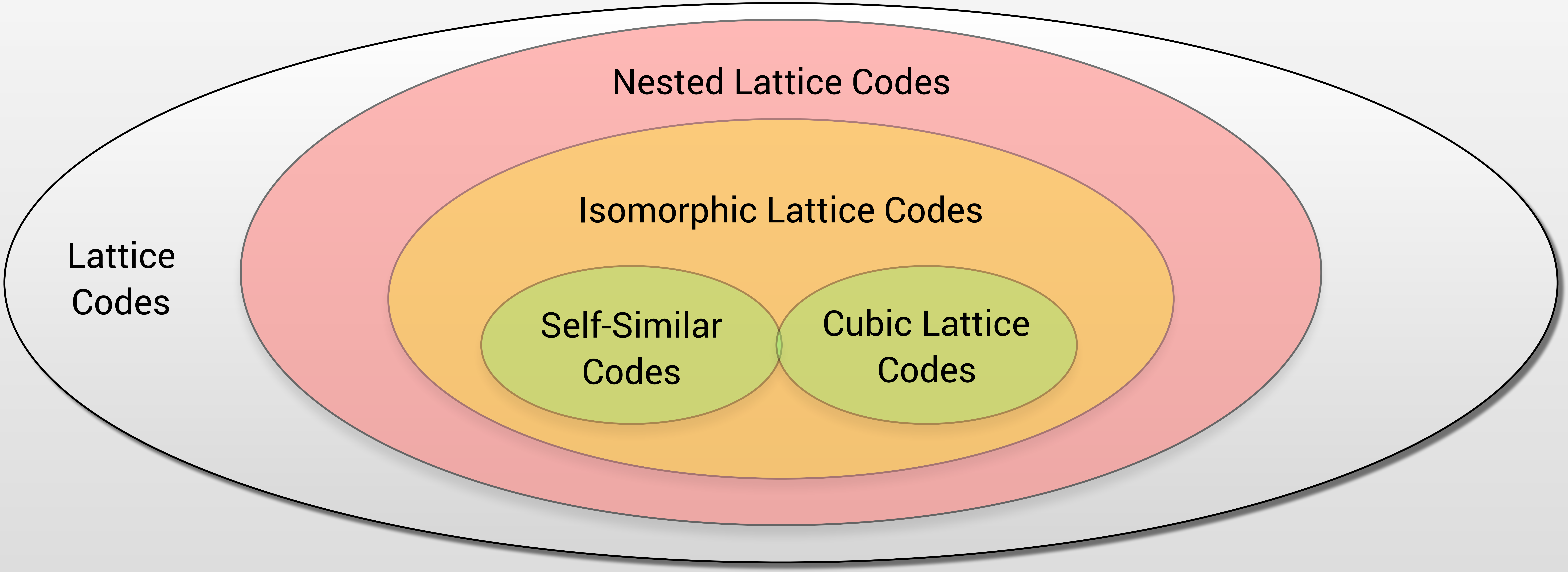
$$\mathbf{G}_{\mathrm{s}} = \begin{bmatrix} 4 & 0 \\ 4 & 8 \end{bmatrix}$$

$$\mathbf{G}_{\mathrm{c}} = \begin{bmatrix} \frac{4}{3} & \frac{2}{9} \\ \frac{4}{3} & \frac{8}{9} \end{bmatrix}$$


-4
-3
-2
-1
0
1
2
3
4
5
6
7
8
9


-4
-3
-2
-1
0
1
2
3
4
5
6
7
8
9

Classification of Lattice Codes



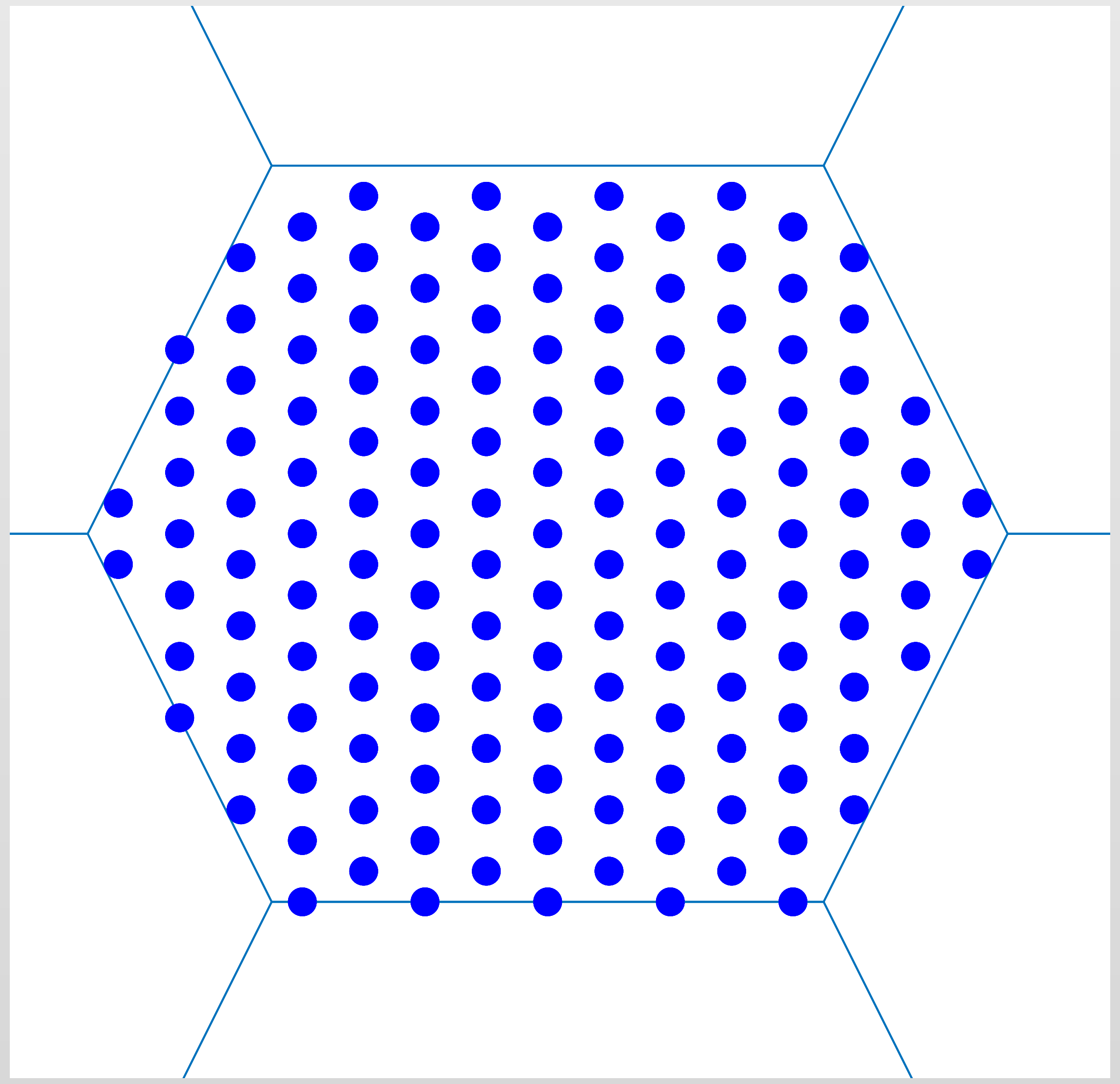
Isomorphism is important for compute-and-forward.

Self-Similar & Cubic Lattice Codes

Self-Similar Lattice Code

Shaping lattice is scaled version of coding lattice

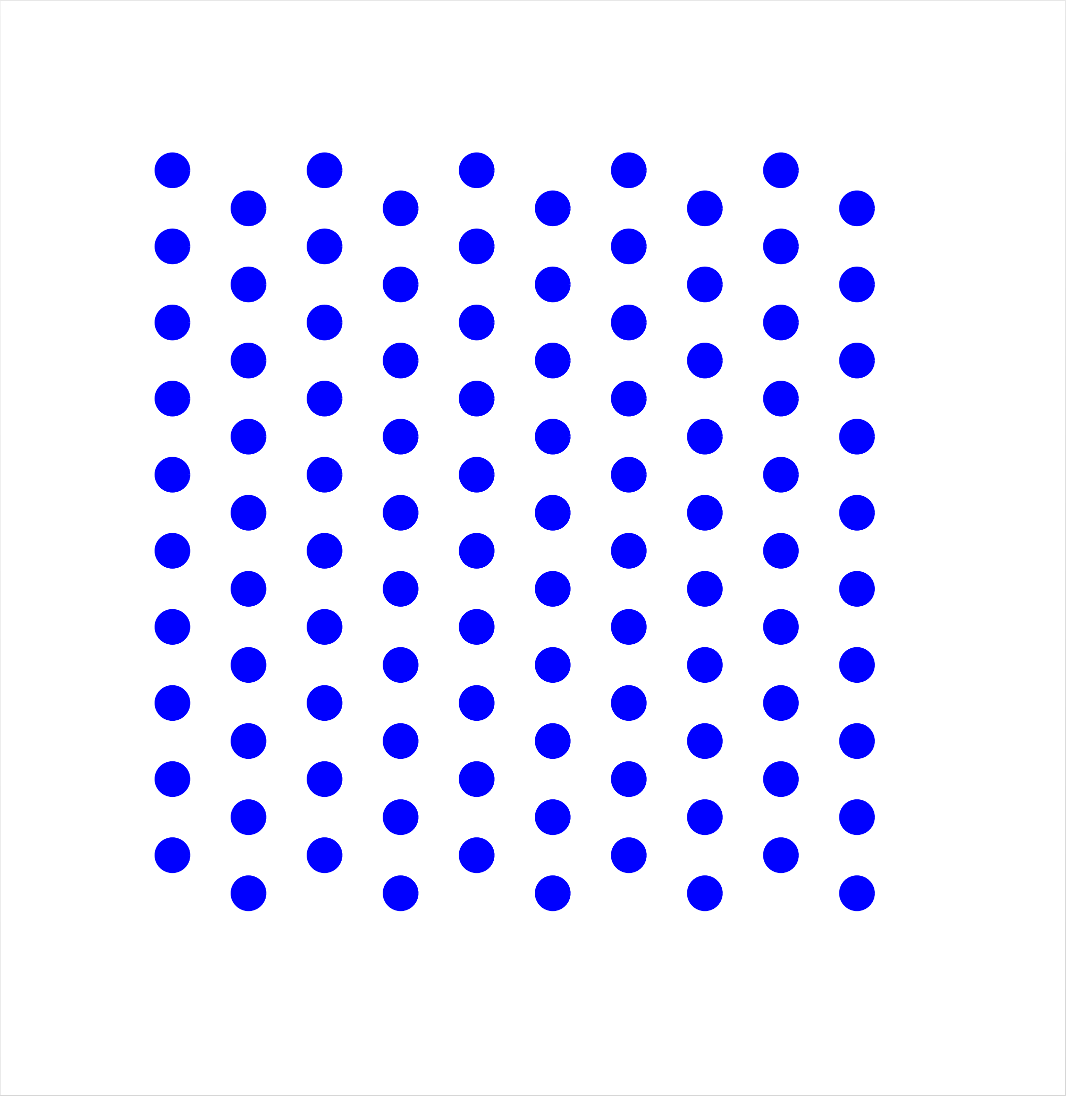
- ✅ Good shaping gain
- ✅ Group isomorphism
- ❌ High encoding complexity



Cubic Lattice Code

Shaping lattice is a cube

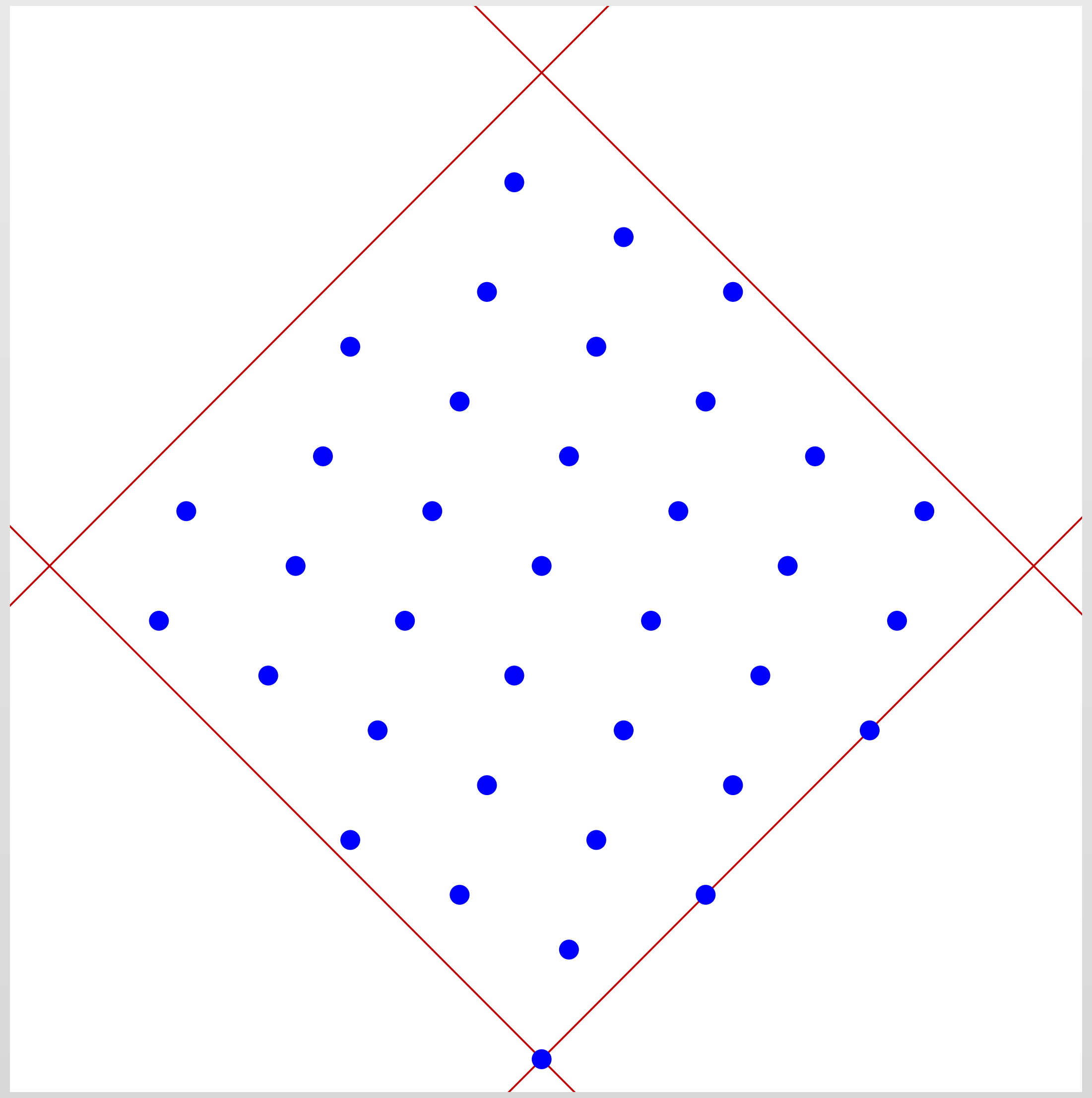
- ❌ No shaping gain
- ✅ Group isomorphism
- ✅ Low encoding complexity



General Nested Lattice Code

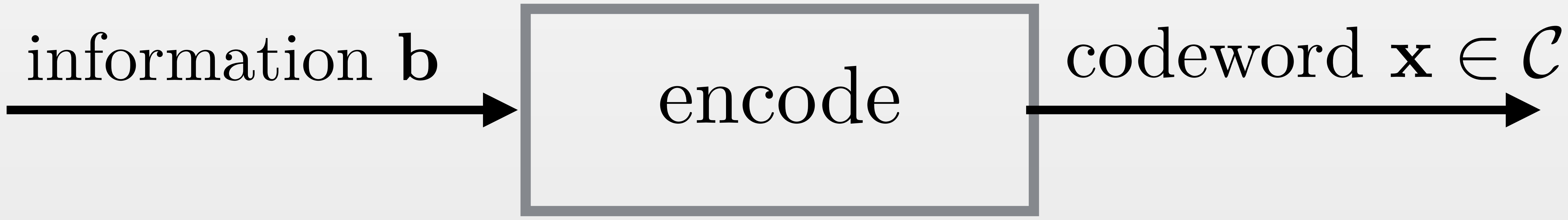
Shaping lattice is sub lattice of coding lattice

- ✅ Good shaping gain
- ❌ No gr. isomorphism (in general)
- ✅ Low encoding complexity



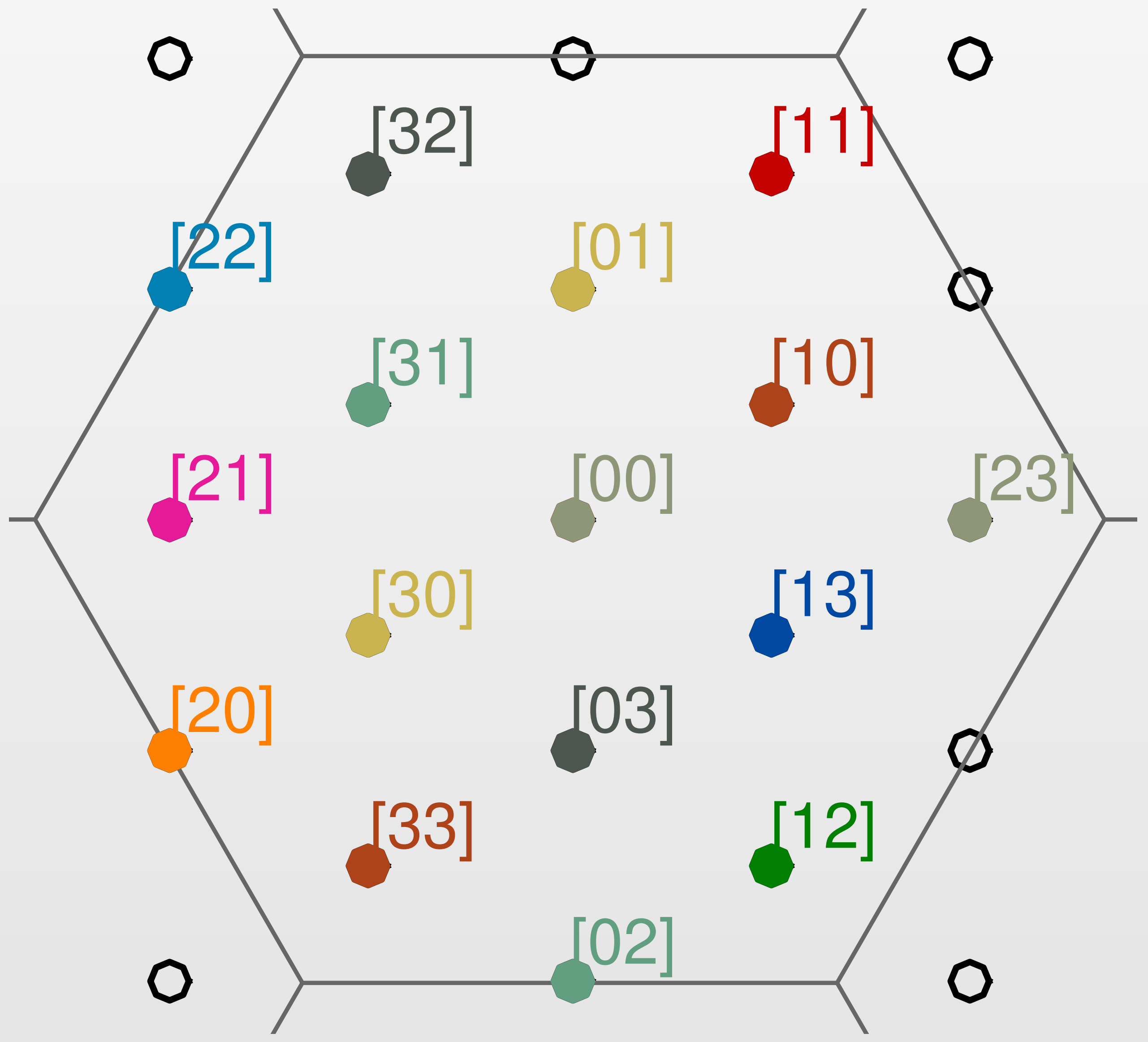
Encoding and Indexing

An encoding function maps information (indices) $\mathbf{b}$ to codewords $\mathbf{x} \in \mathcal{C}$



Indexing is the inverse of encoding maps, codewords $\mathbf{x} \in \mathcal{C}$ to information (indices) $\mathbf{b}$.

Not decoding: there is no noise.



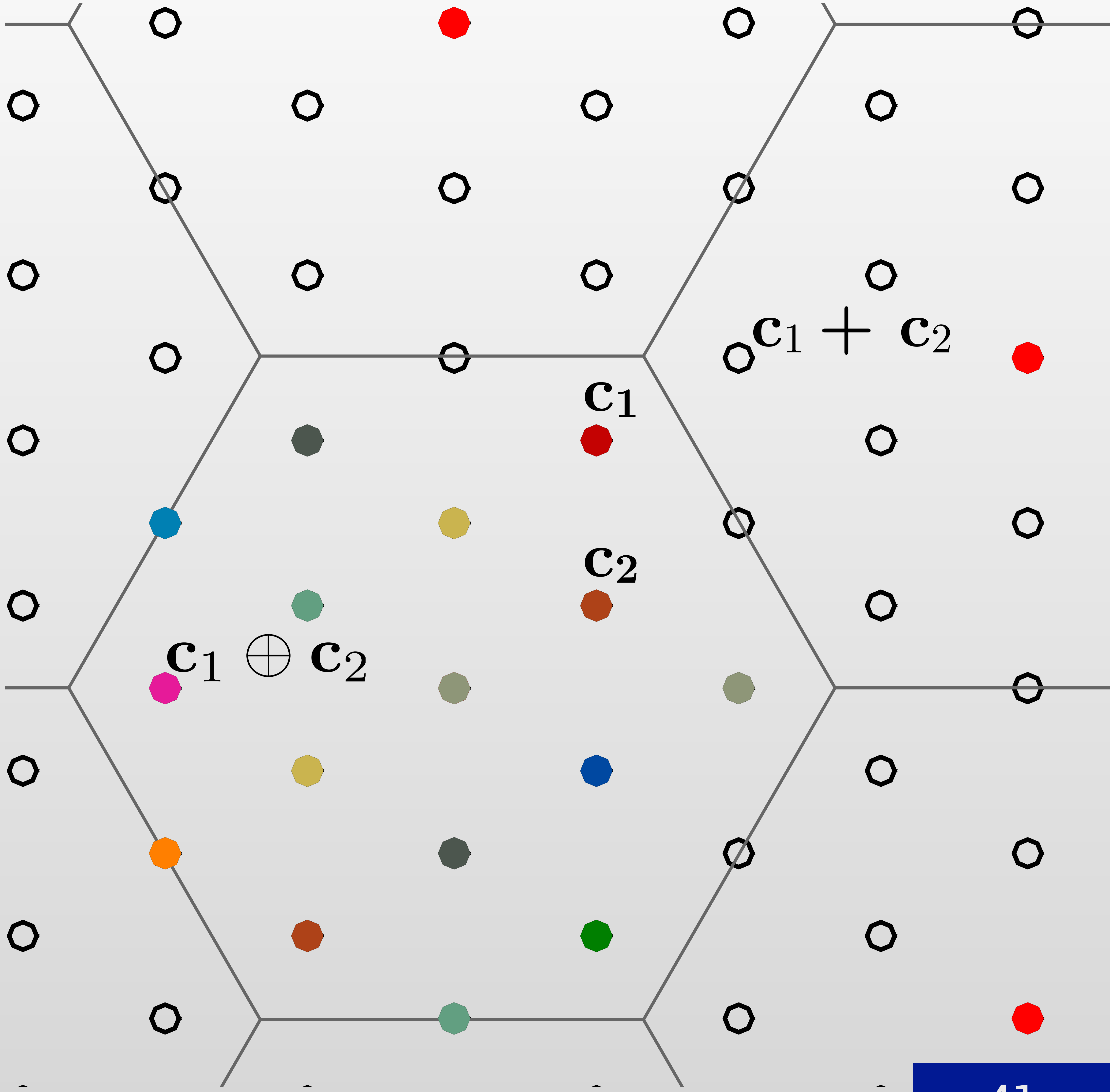
Main result encoding and indexing is possible if generator matrices are both in triangular form.

B. M. Kurkoski, "Encoding and indexing of lattice codes," IEEE Transactions on Information Theory, vol. 64, pp. 6320-6332, September 2018.

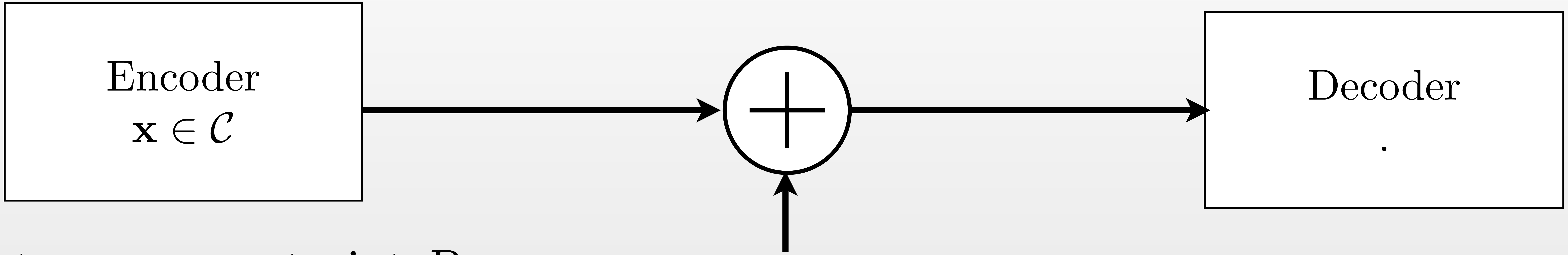
A Nested Lattice Code is a Group

- Lattice Λ is a group: $\mathbf{a}, \mathbf{b} \in \Lambda \Rightarrow \mathbf{a} + \mathbf{b} \in \Lambda$
- $\Lambda_{\mathrm{s}} \subseteq \Lambda_{\mathrm{c}}$. Thus Λ_{s} is a subgroup of Λ_{c}.
- The quotient group is $\Lambda_{\mathrm{c}}/\Lambda_{\mathrm{s}}$, and is the set of all cosets of Λ_{s} in Λ_{c}.
- Group operation. Let $\mathbf{c}_1, \mathbf{c}_2 \in \Lambda_{\mathrm{c}}/\Lambda_{\mathrm{s}}$, then:

$$\mathbf{c}_1 \oplus \mathbf{c}_2 = (\mathbf{c}_1 + \mathbf{c}_2) \bmod \Lambda_{\mathrm{s}}$$



AWGN Channel Capacity



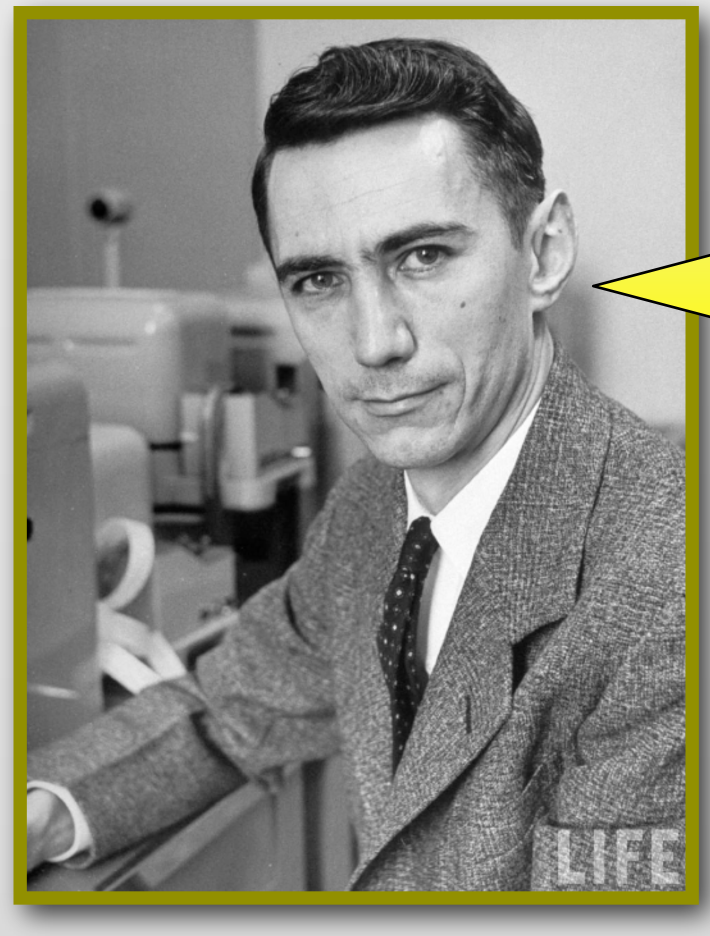
$$\text{AWGN} \sim \mathcal{N}(0, \sigma^2)$$

Input power constraint P :

$$\frac{1}{n}||\mathbf{x}||^2 \leq P$$

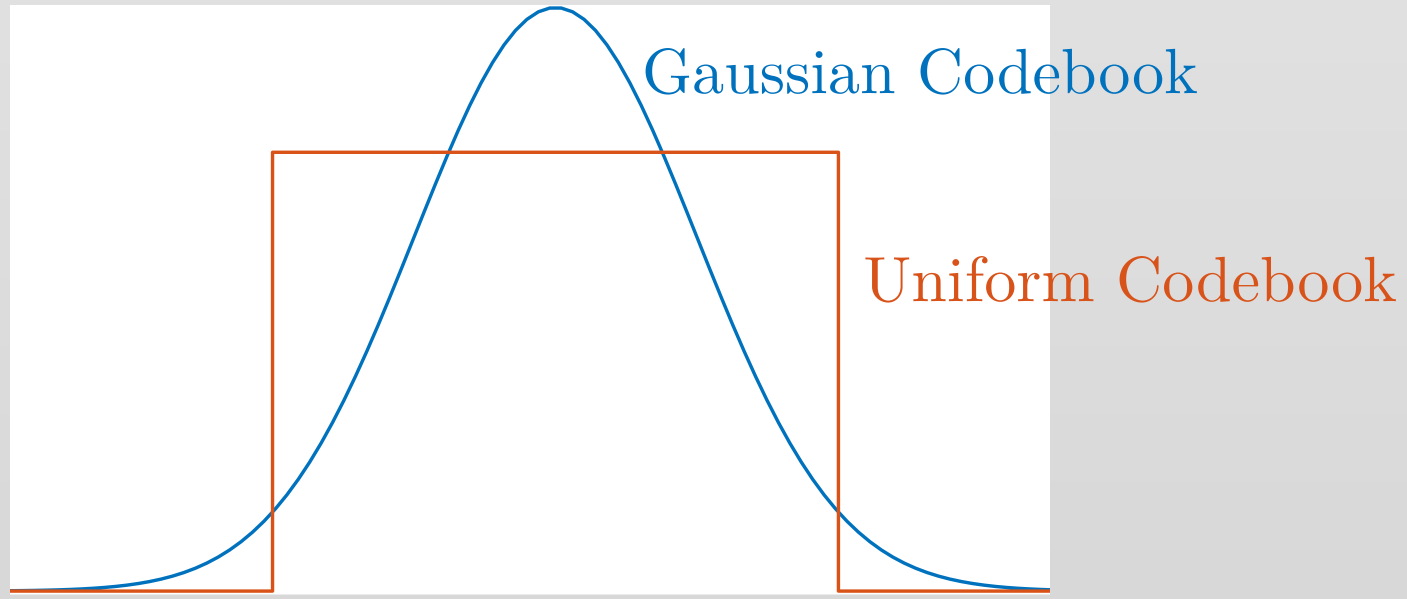
Capacity is:

$$R < C = \frac{1}{2}\log(1 + \frac{P}{\sigma^2})$$

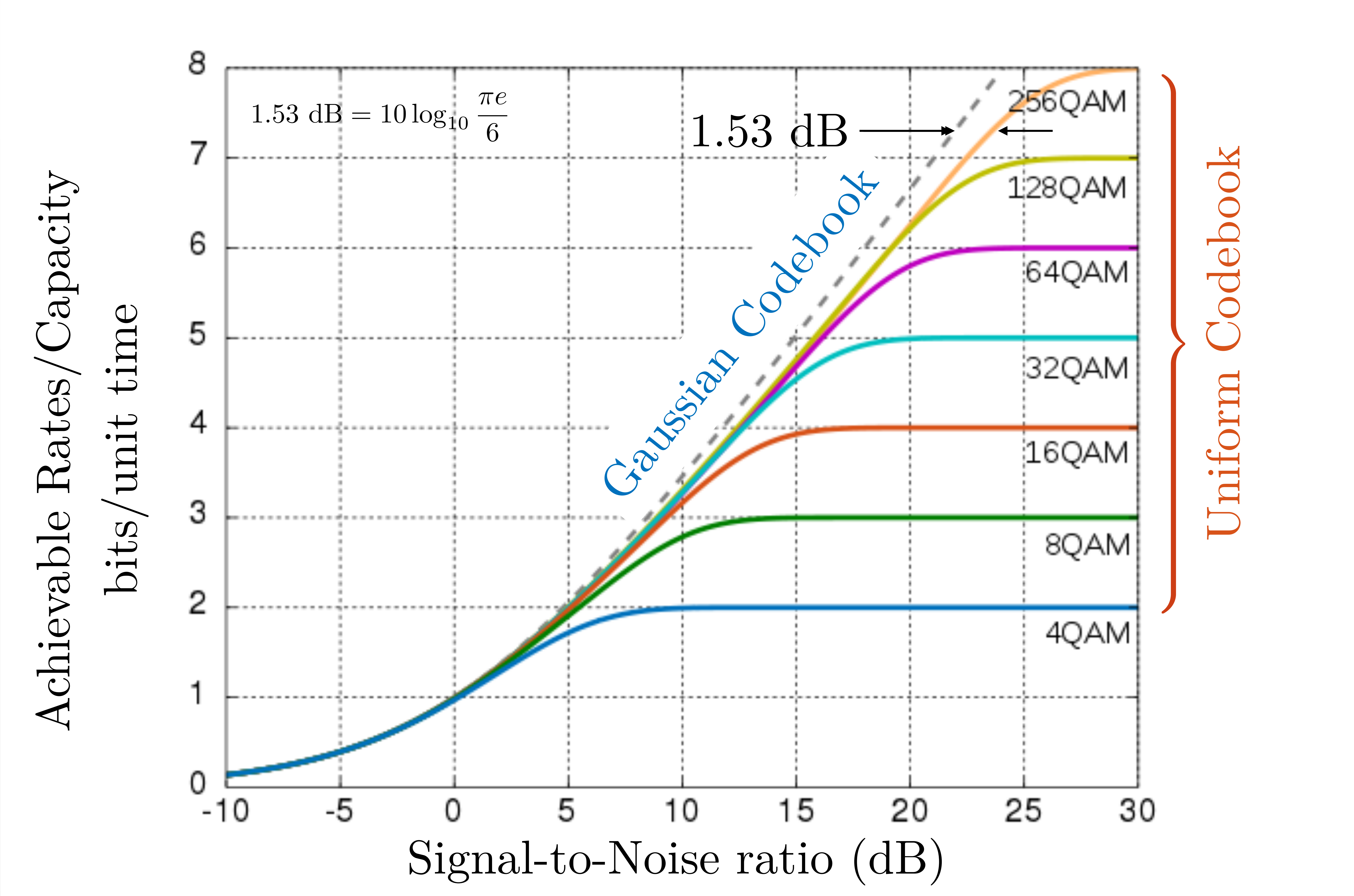


Claude Shannon
father of information theory

Gaussian codebook maximizes capacity, uniform codebook (QAM) cannot



Gaussian Codebook vs QAM (Uniform)



At high SNR, high rates, using a Gaussian codebook (sphere-like) gain 1.53 dB

No special benefit to using Gaussian codebook at low rates/low SNR

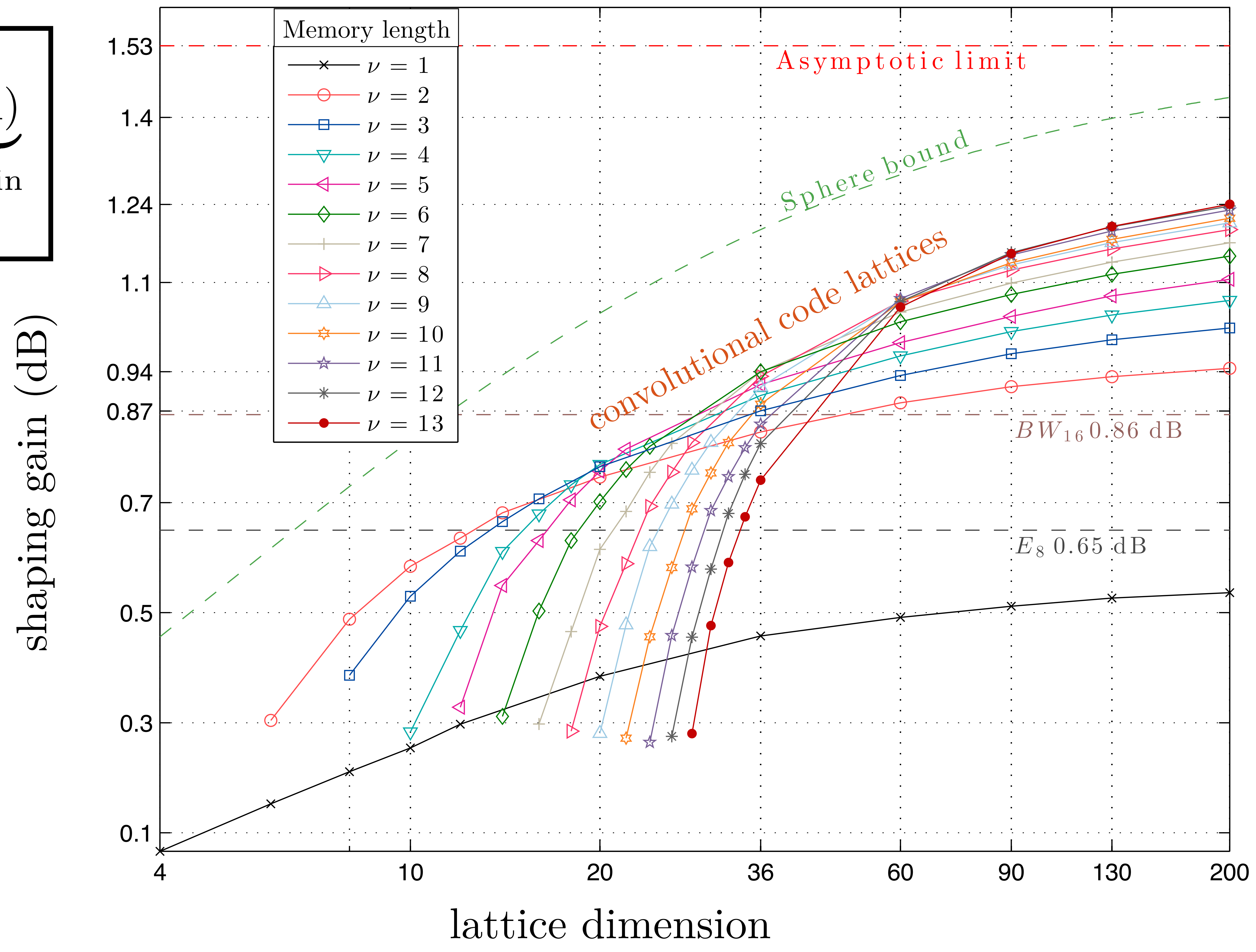
Shaping Gain is Reduction of Transmit Power

$$\text{Average Power} \approx \underbrace{\frac{\int_B ||\mathbf{x}||^2 d\mathbf{x}}{nV(B)^{\frac{2}{n}+1}}}_{\text{shaping gain } G(B)} \cdot \underbrace{M^n V(\Lambda)}_{\text{coding gain}}$$

A spherical codebook has a Gaussian input distribution, as n to infinity.

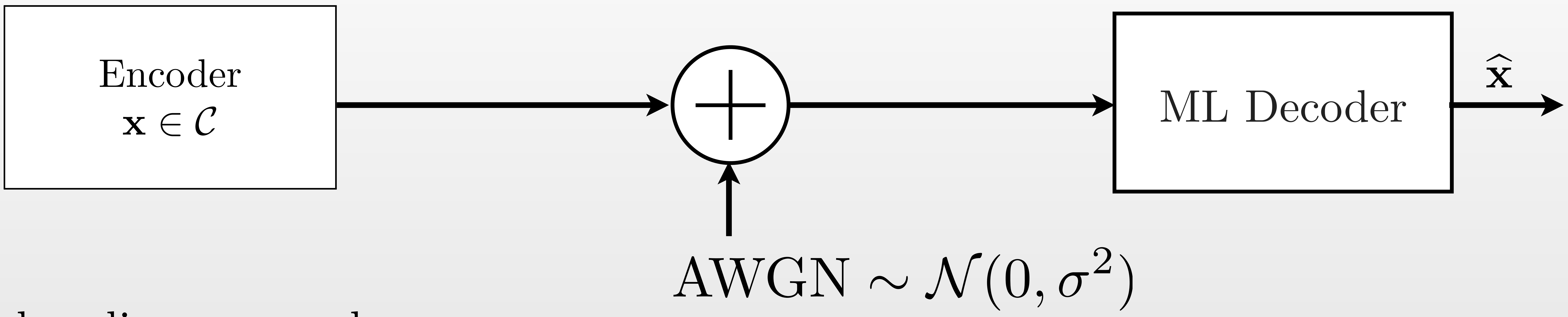
The shaping gain of various lattices is shown at the left

Proposed convolutional code lattices have excellent performance-complexity tradeoff [ZK17]



[ZK17] F. Zhou and B. M. Kurkoski, “Shaping LDLC lattices using convolutional code lattices,” IEEE Communications Letters, pp. 730-733, April 2017.

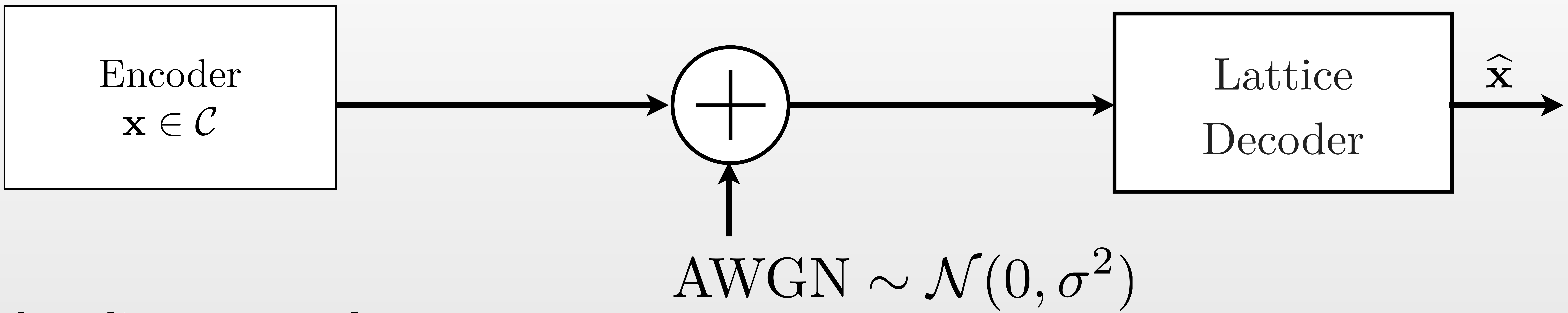
Lattice Code ML Decoding Achieves Capacity



Lattice decoding approaches:

- Maximum likelihood decoding achieves capacity $C = \frac{1}{2}\log(1 + P/\sigma^2)$ [de Buda. Urbanke and Rimoldi]. But this is not practical.

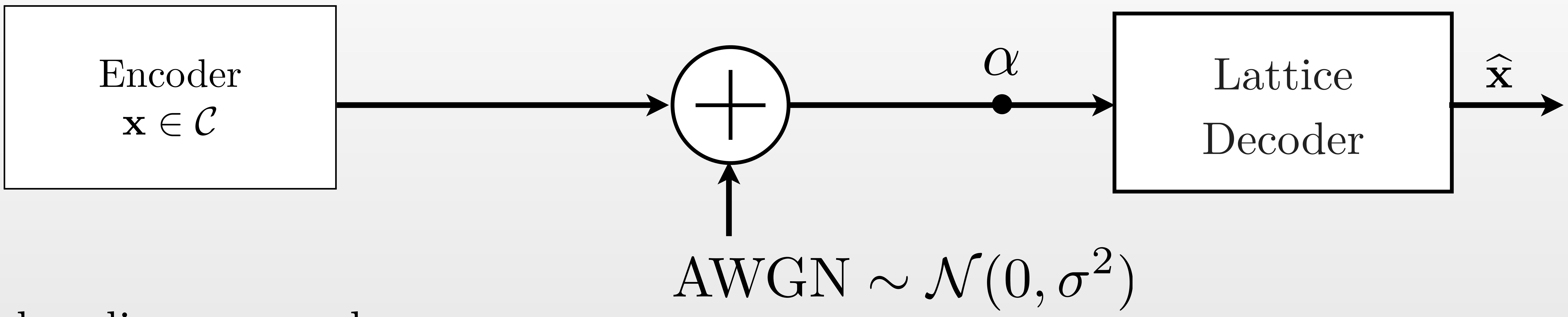
Lattice Codes with Lattice Decoding



Lattice decoding approaches:

- Maximum likelihood decoding achieves capacity $C = \frac{1}{2}\log(1 + P/\sigma^2)$ [de Buda. Urbanke and Rimoldi]. But this is not practical.
- Lattice decoding only achieves $R < \frac{1}{2}\log(P/\sigma^2)$ [Loeliger]. Practical, but "lattice decoding" ignores the codebook boundaries.

Lattice Codes with Inflated Lattice Decoding



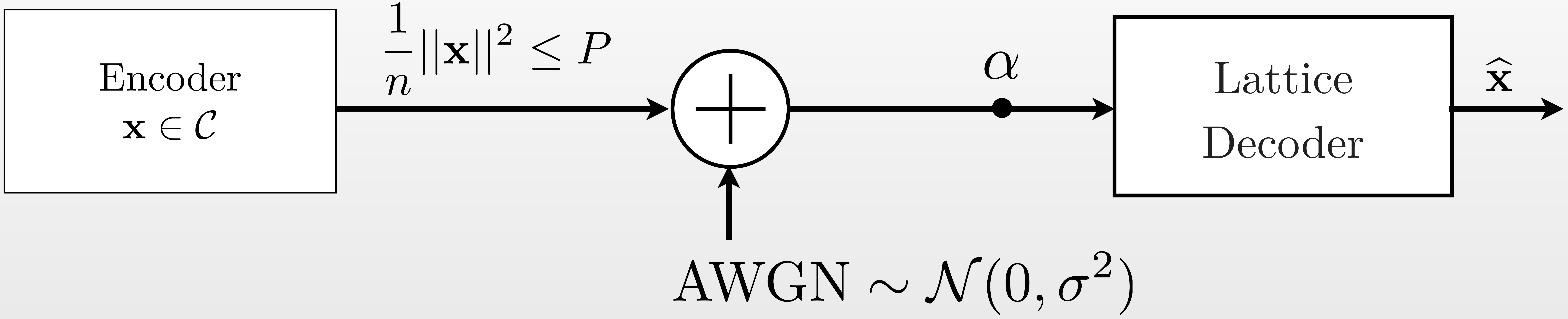
Lattice decoding approaches:

- Maximum likelihood decoding achieves capacity $C = \frac{1}{2}\log(1 + P/\sigma^2)$ [de Buda. Urbanke and Rimoldi]. But this is not practical.

- Lattice decoding only achieves $R < \frac{1}{2}\log(P/\sigma^2)$ [Loeliger]. Practical, but "lattice decoding" ignores the codebook boundaries.

- Lattice decoding with lattice inflation achieves $C = \frac{1}{2}\log(1 + P/\sigma^2)$ [Erez and Zamir] Amazing!

Decoding Nested Lattice Codes



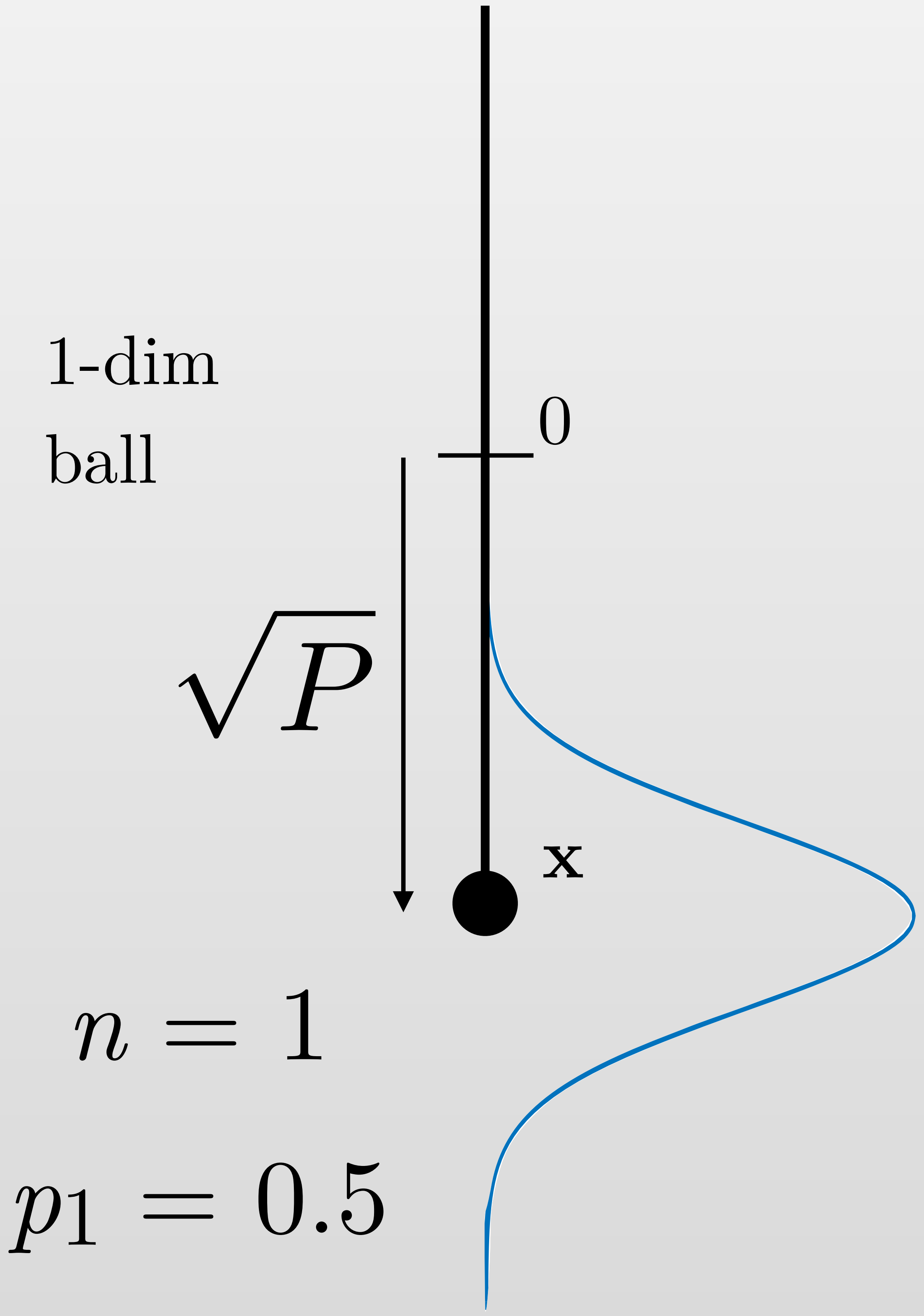
$$\alpha = \frac{P}{P + \sigma^2}$$ MMSE coefficient

"inflates" lattice by α^{-1}

Intuition for Lattice Inflation

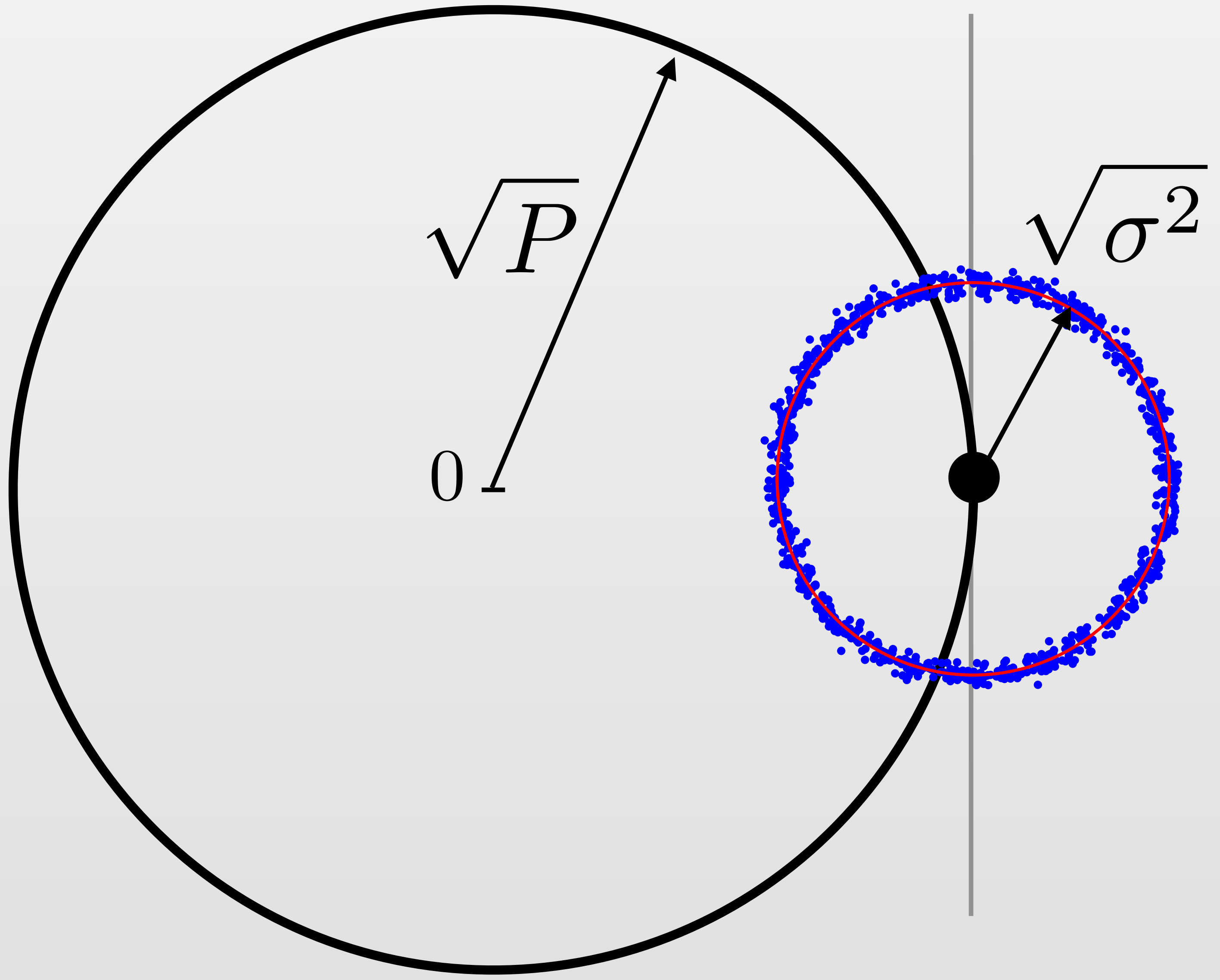
Assume codeword $\mathbf{c}$ is on the surface of n-ball. Noise is added to get $\mathbf{y}$

What is the probability p_n that $\mathbf{y}$ is outside of the ball?



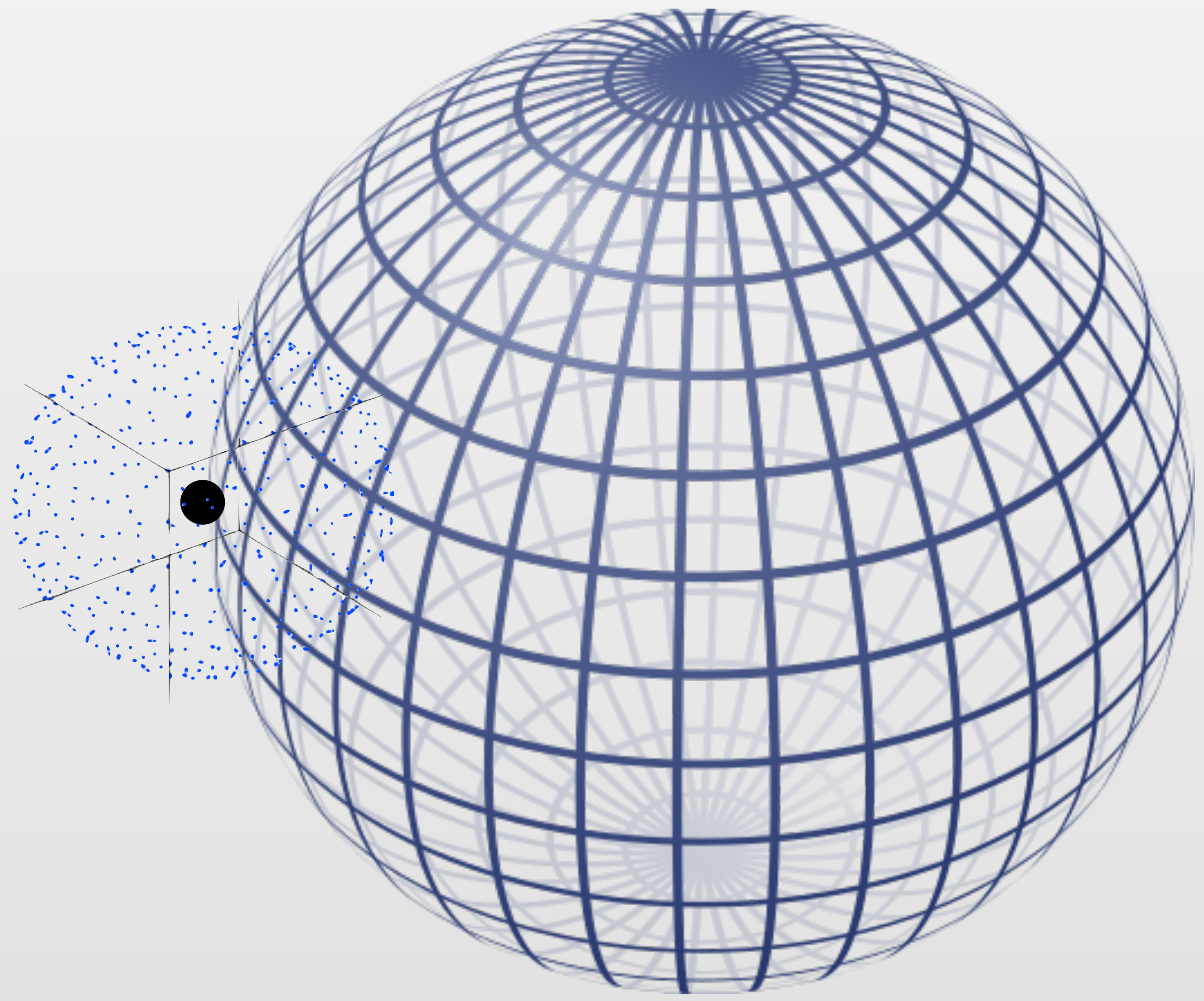
$n = 1$

$p_1 = 0.5$



$n = 2$

$p_2 > 0.5$



$n = 3$

$p_3 > p_2 > 0.5$

As $n \to \infty$ the noise tends to be outside of the ball

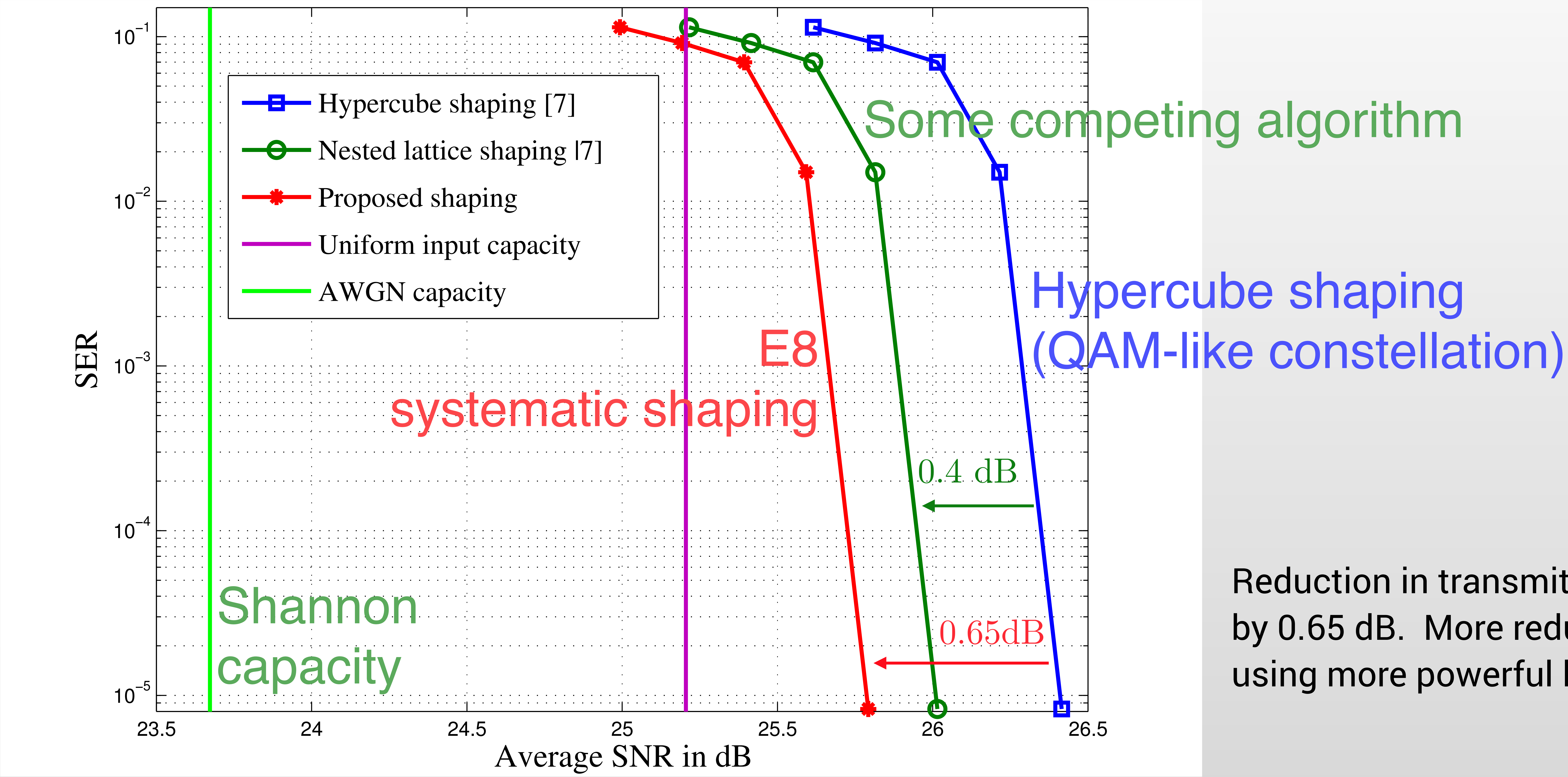

Codebook for transmission
Lattice for decoding


Codebook for transmission
Lattice for decoding

Lattice "inflation" by

$$\alpha = \frac{P}{P+\sigma^2}$$

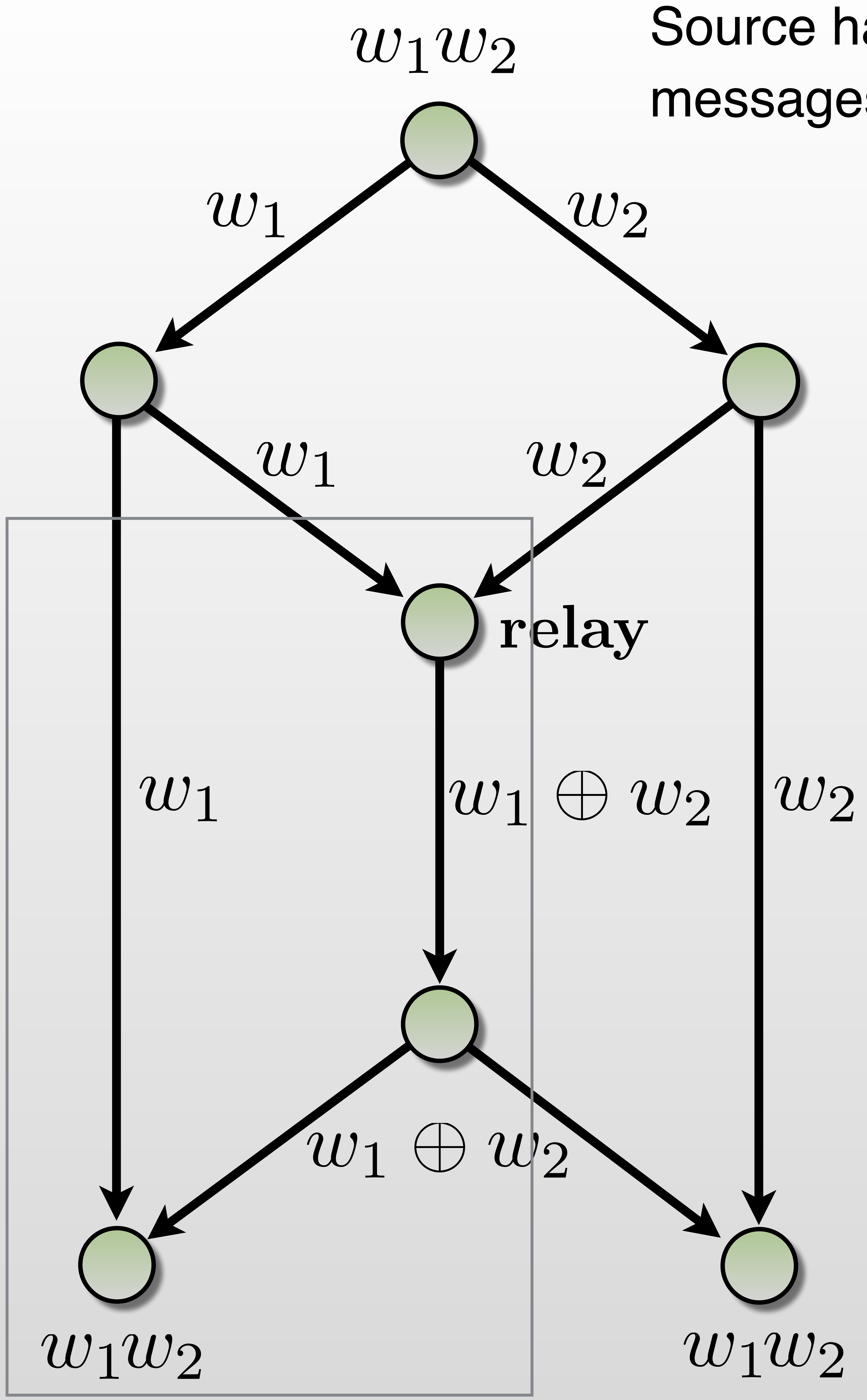
Shaping LDLC using E8 Lattice



Reduction in transmit power by 0.65 dB. More reduction by using more powerful lattices.

Routing vs. Network Coding

Source has messages $w_1 w_2 w_3$



matrix form...

Capacity: max rate from source to destination

Routing

- Internal nodes only forward one incoming packet
- Capacity $= 3/2$

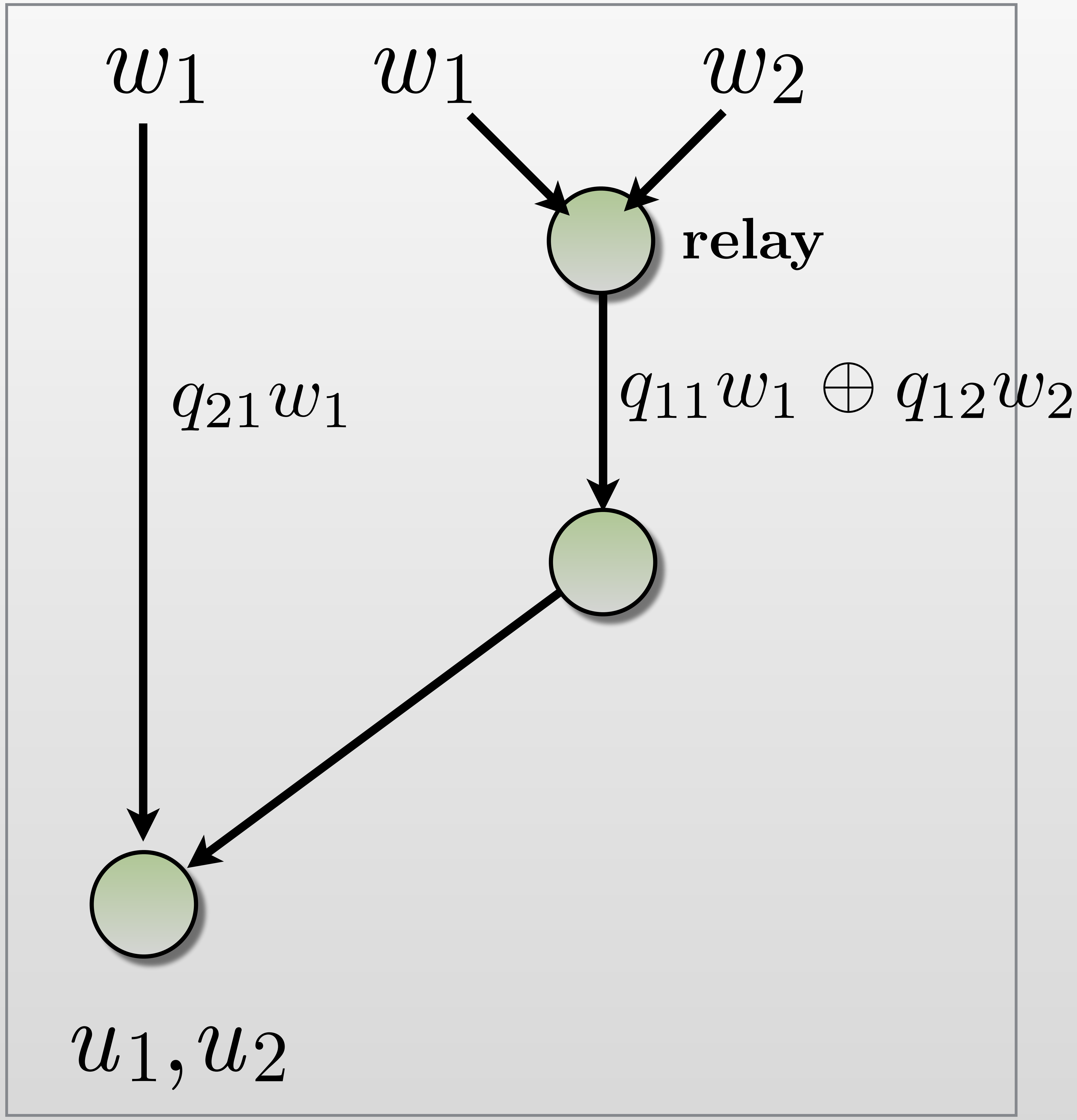
Network Coding

- Internal nodes perform linear operations
- Capacity $= 2$

Forwarding combinations of messages can increase capacity

Destinations wants messages $w_1 w_2 w_3$

Matrix Form Recovery of Messages



w, u, q in a field. Allow relay to multiply by q

2 received messages and 2 desired messages:

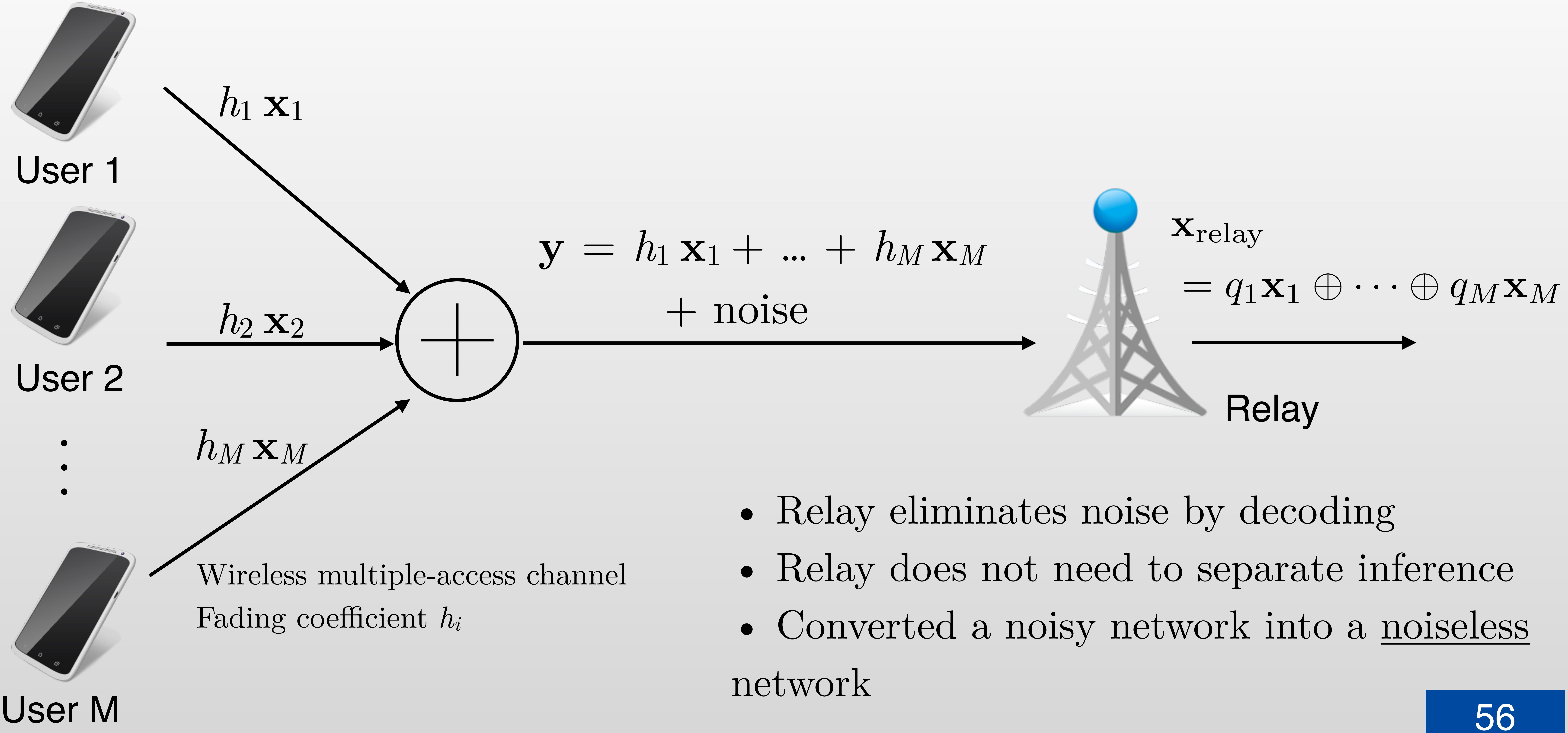
$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}}_{\mathbf{Q}} \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}^{-1} \cdot \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\text{received messages}} = \underbrace{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}}_{\text{desired messages}}$$

Destination should receive sufficient linear combinations such that $\mathbf{Q}$ is invertible

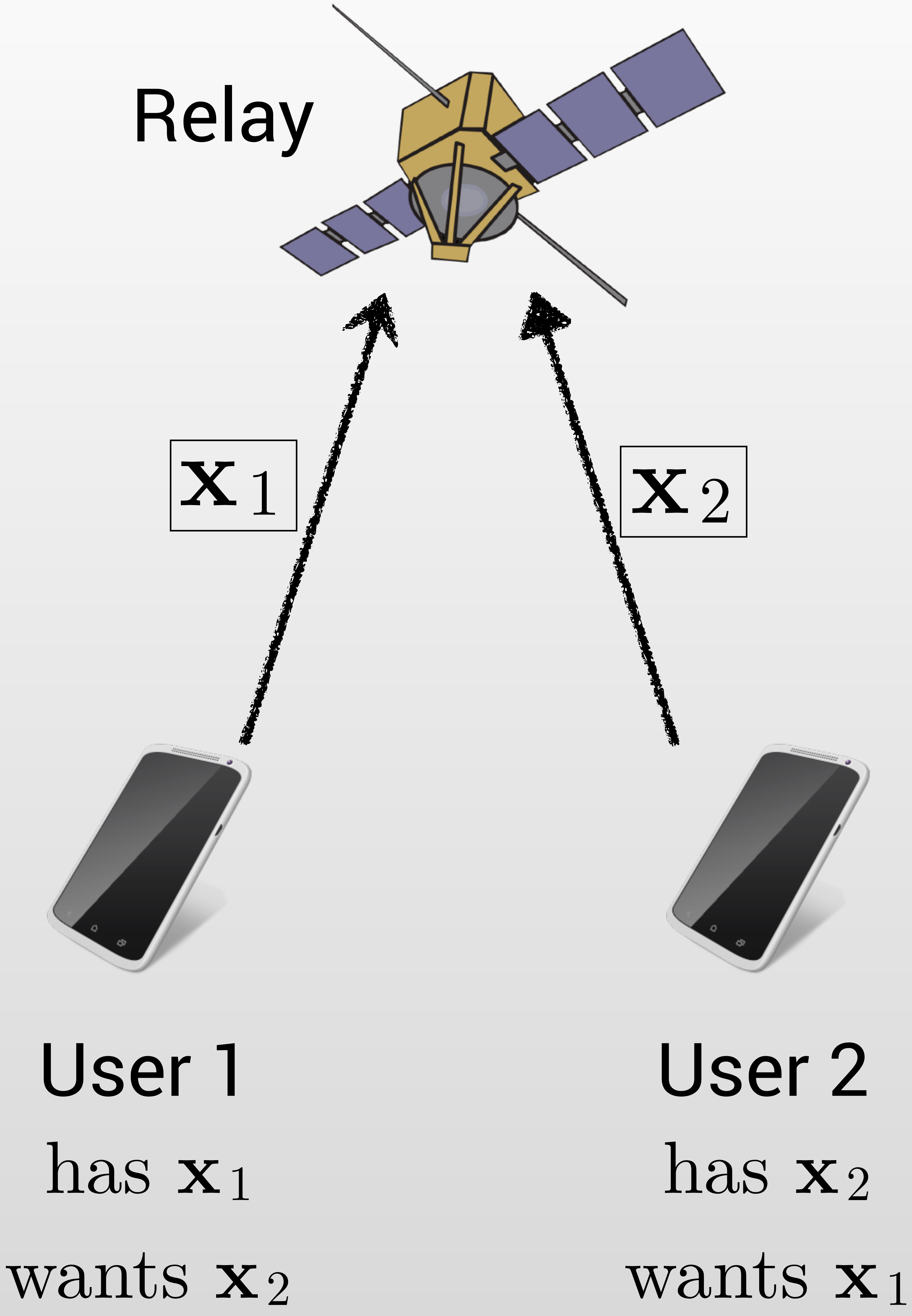
PLNC = Physical Layer Network Coding

Addition occurs over the air



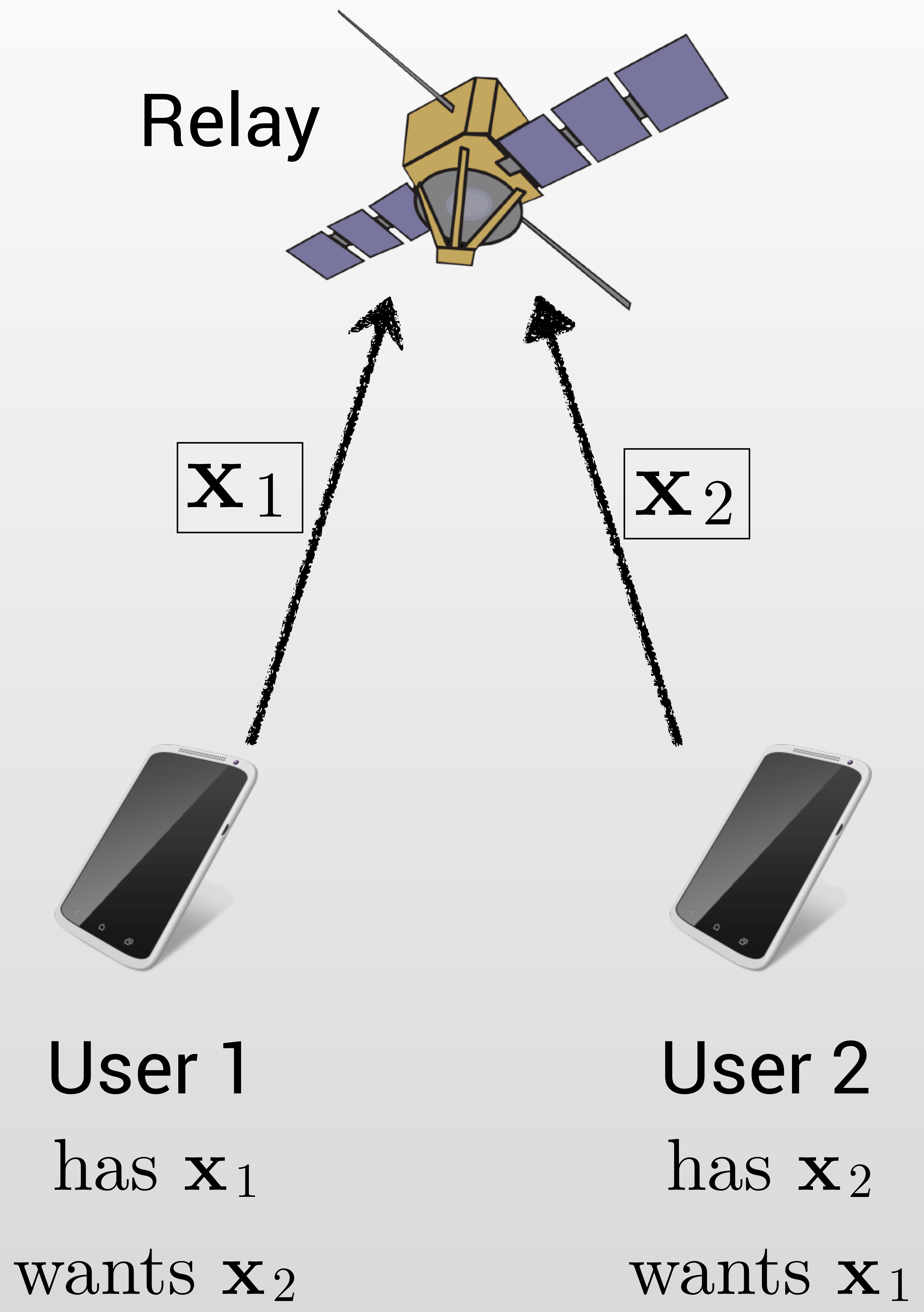
- Relay eliminates noise by decoding
- Relay does not need to separate inference
- Converted a noisy network into a <u>noiseless</u> network

Bidirectional Relay Channel



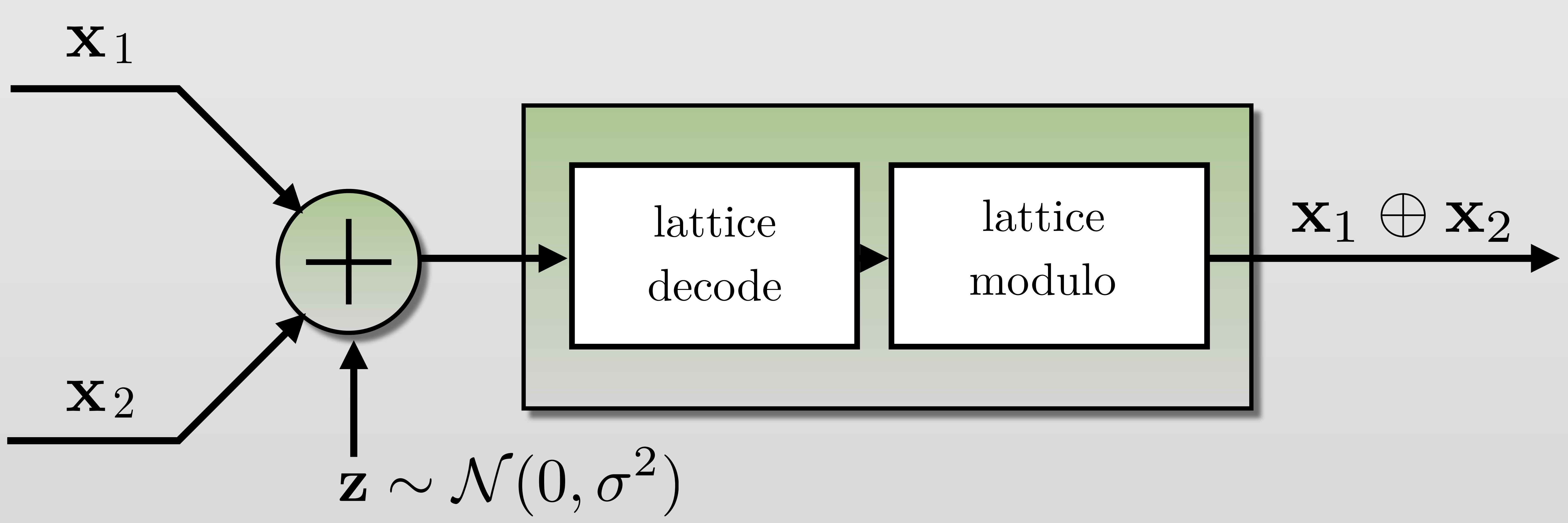
- Orthogonal: uses 4 time slots
- Network coding: uses 3 time slots
- Physical layer network coding (PLNC): 2 time slots

Bidirectional Relay Channel

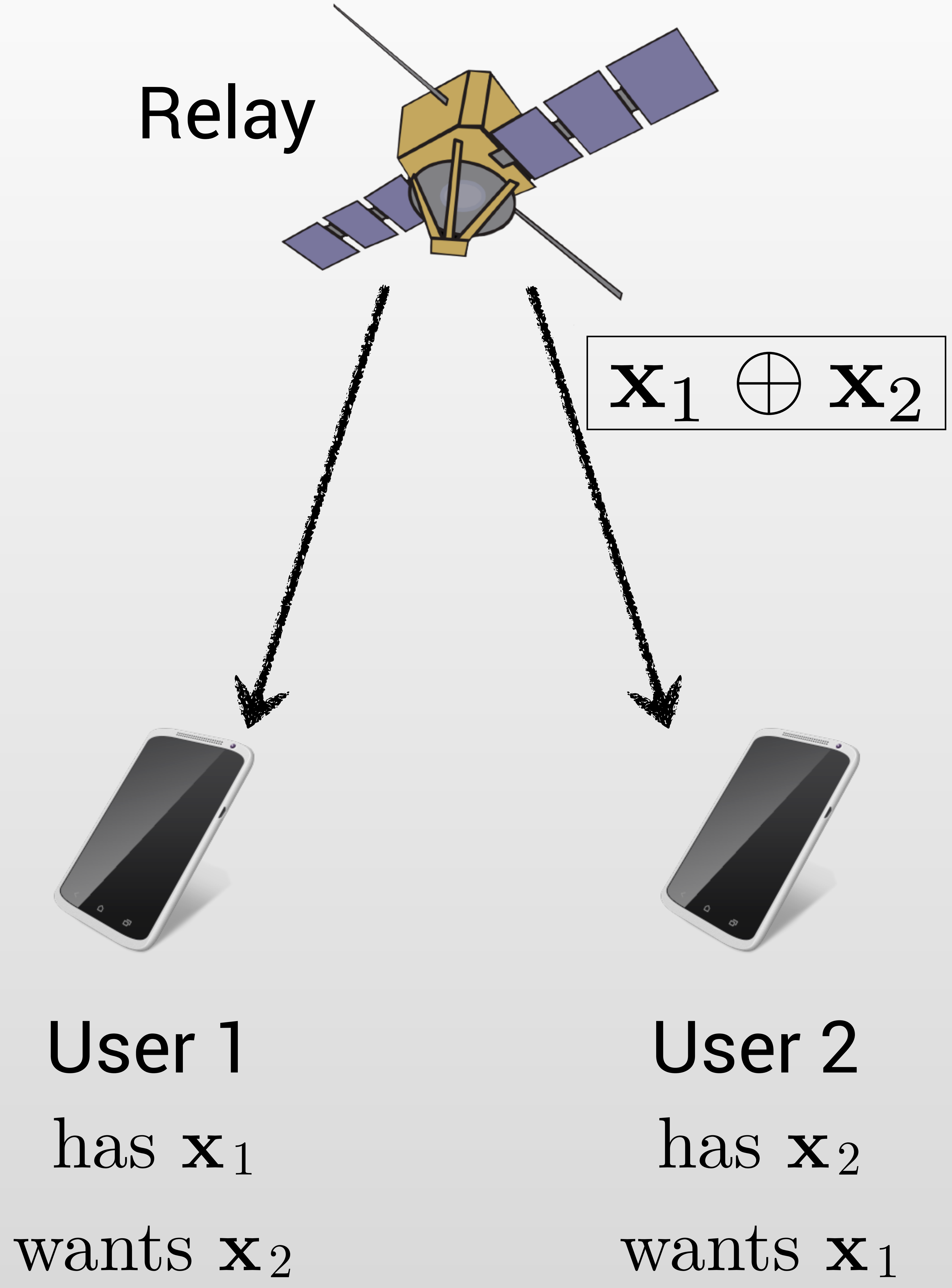


- Orthogonal: uses 4 time slots
- Network coding: uses 3 time slots
- Physical layer network coding (PLNC): 2 time slots

Relay Using PLNC

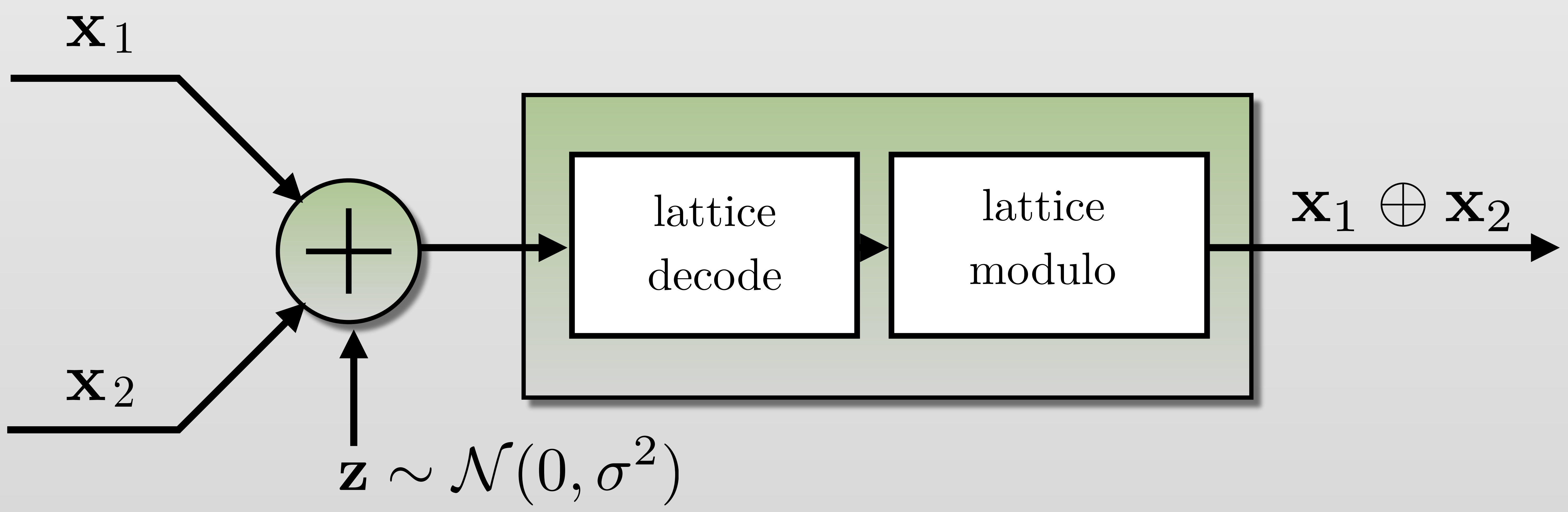


Bidirectional Relay Channel

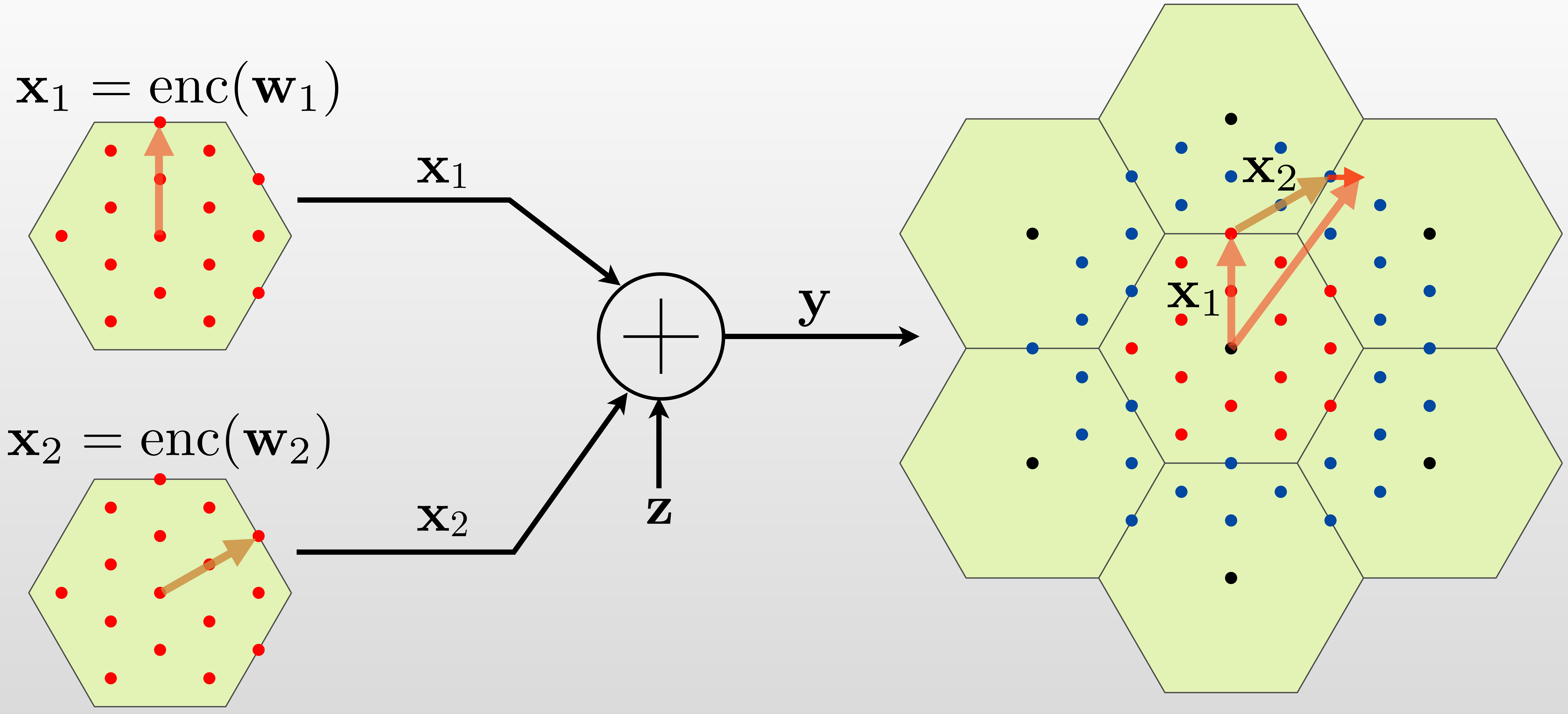


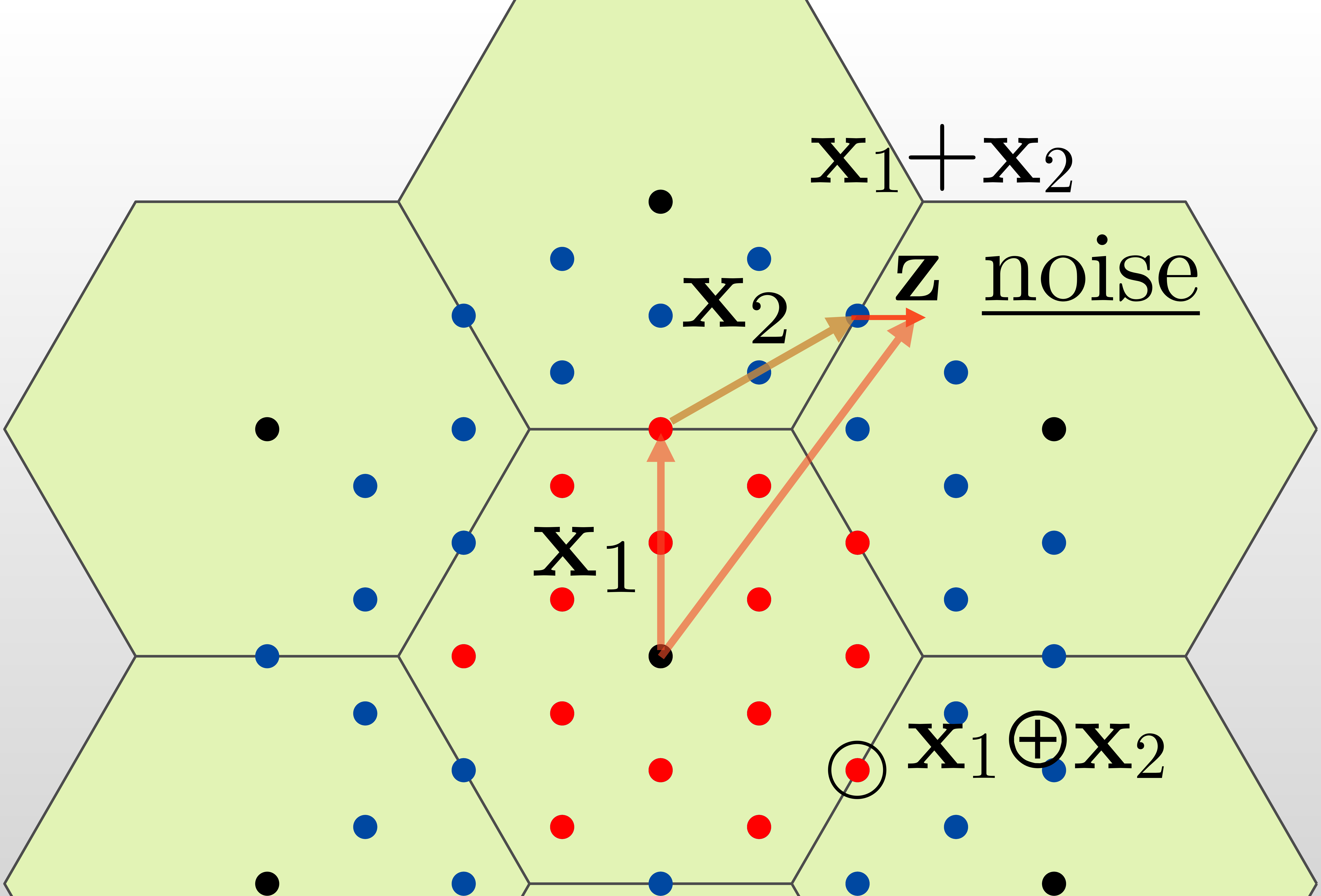
- Orthogonal: uses 4 time slots
- Network coding: uses 3 time slots
- Physical layer network coding (PLNC): 2 time slots

Relay Using PLNC



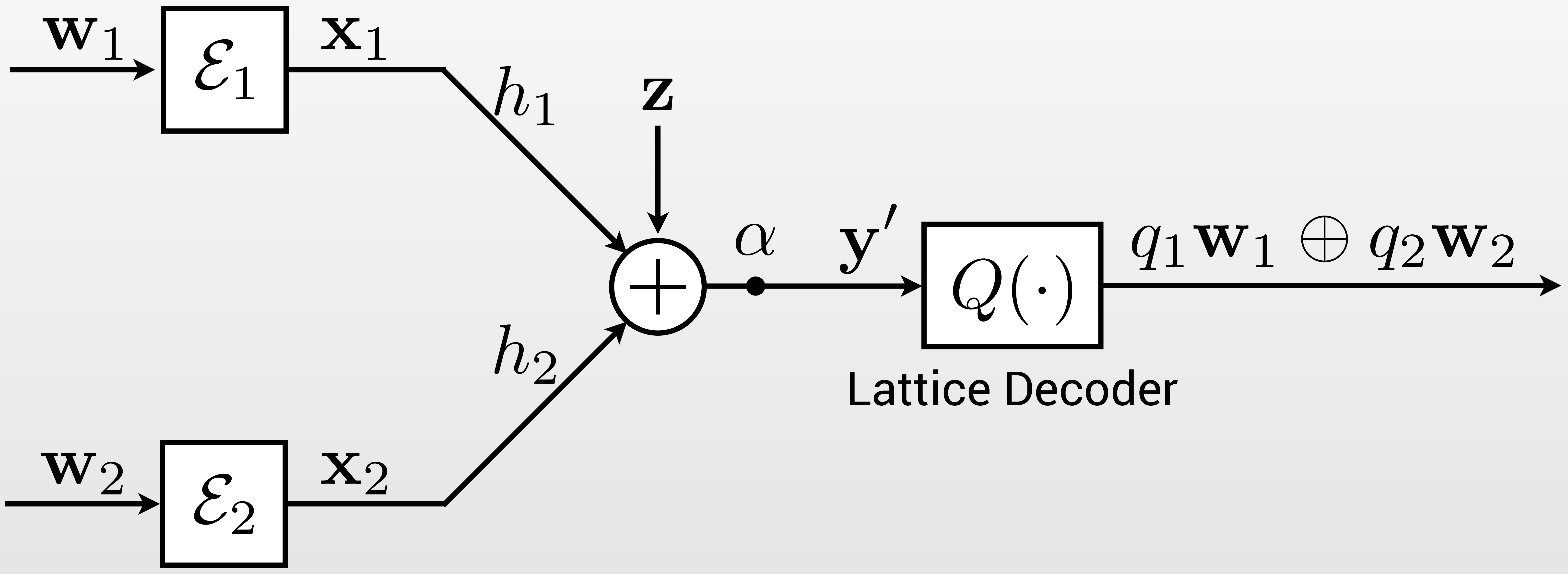
Relay Using PLNC





$\mathbf{x}_1 + \mathbf{x}_2$
$\mathbf{z}$ noise
$\mathbf{x}_2$
$\mathbf{x}_1$
$\mathbf{x}_1 \oplus \mathbf{x}_2$

What if channel coefficients are not integers? Compute-and-Forward



In practice, fading coefficients h are arbitrary values, not integers.

PLNC can still work. This is "compute and forward"

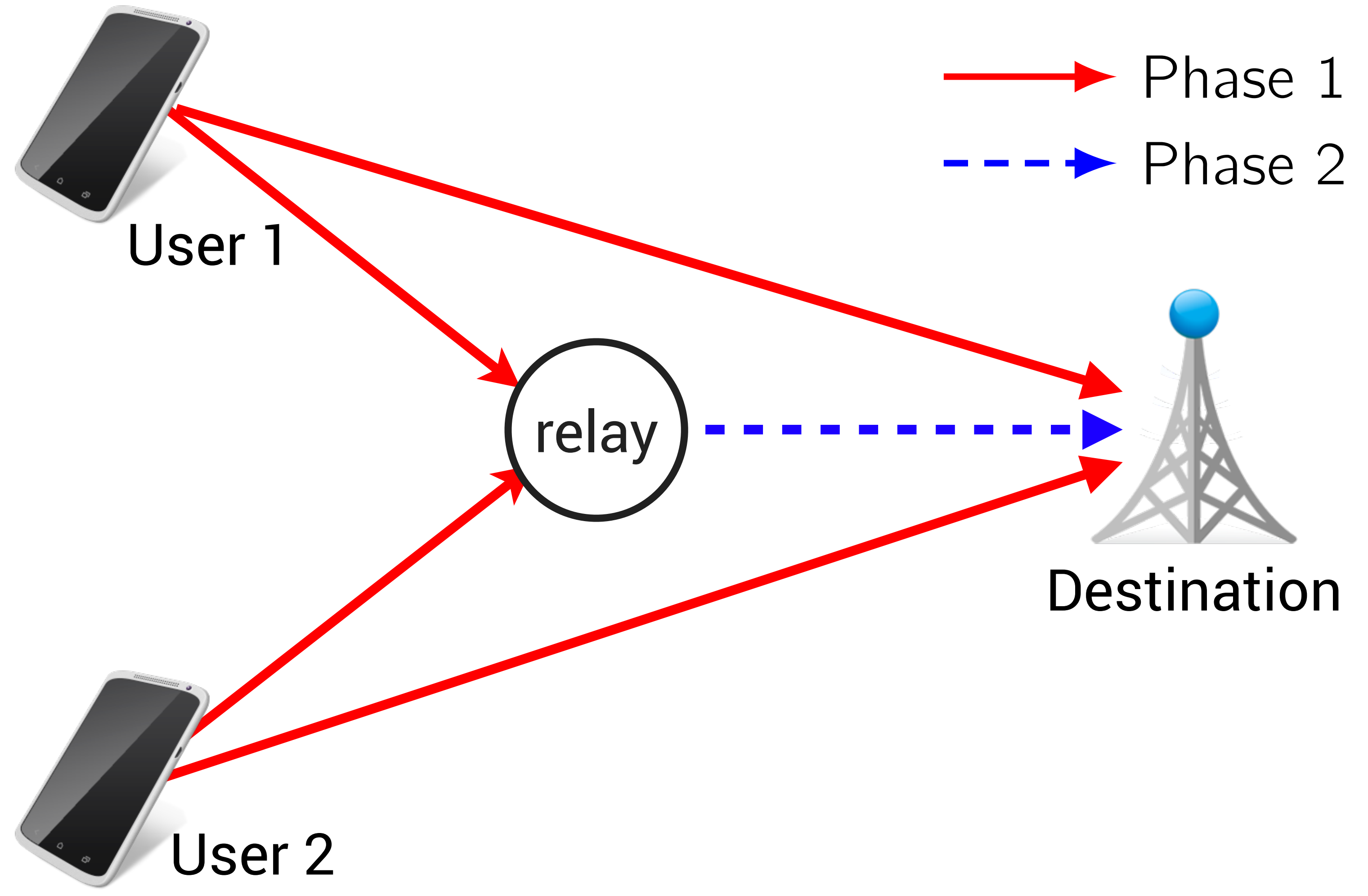
$$\mathbf{y}' = \alpha h_1 \mathbf{x}_1 + \alpha h_2 \mathbf{x}_2 + \alpha \mathbf{z} \qquad \text{fading coefficients } h \in \mathbb{R}$$

$$\mathbf{y}' = a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2 + \mathbf{z}_{\text{eff}} \qquad \text{integer approximation } a \in \mathbb{Z}$$

$$Q(\mathbf{y}') = q_1 \mathbf{w}_1 \oplus q_2 \mathbf{w}_2 \qquad \text{conversion to finite field } q, \mathbf{w} \in \mathbb{F}^n$$

Finding a_1, a_2 is an optimization problem

Compute-Forward for Multiple Access Relay Channel



MARC — Multiple Access Relay Channel
User 1 and User 2 each send one message to Destination with the help of Relay

Naive application of CF to MARC

Relay and Destination independently choose coefficient vectors

destination gets two independent vectors

$$\begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}^{-1} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

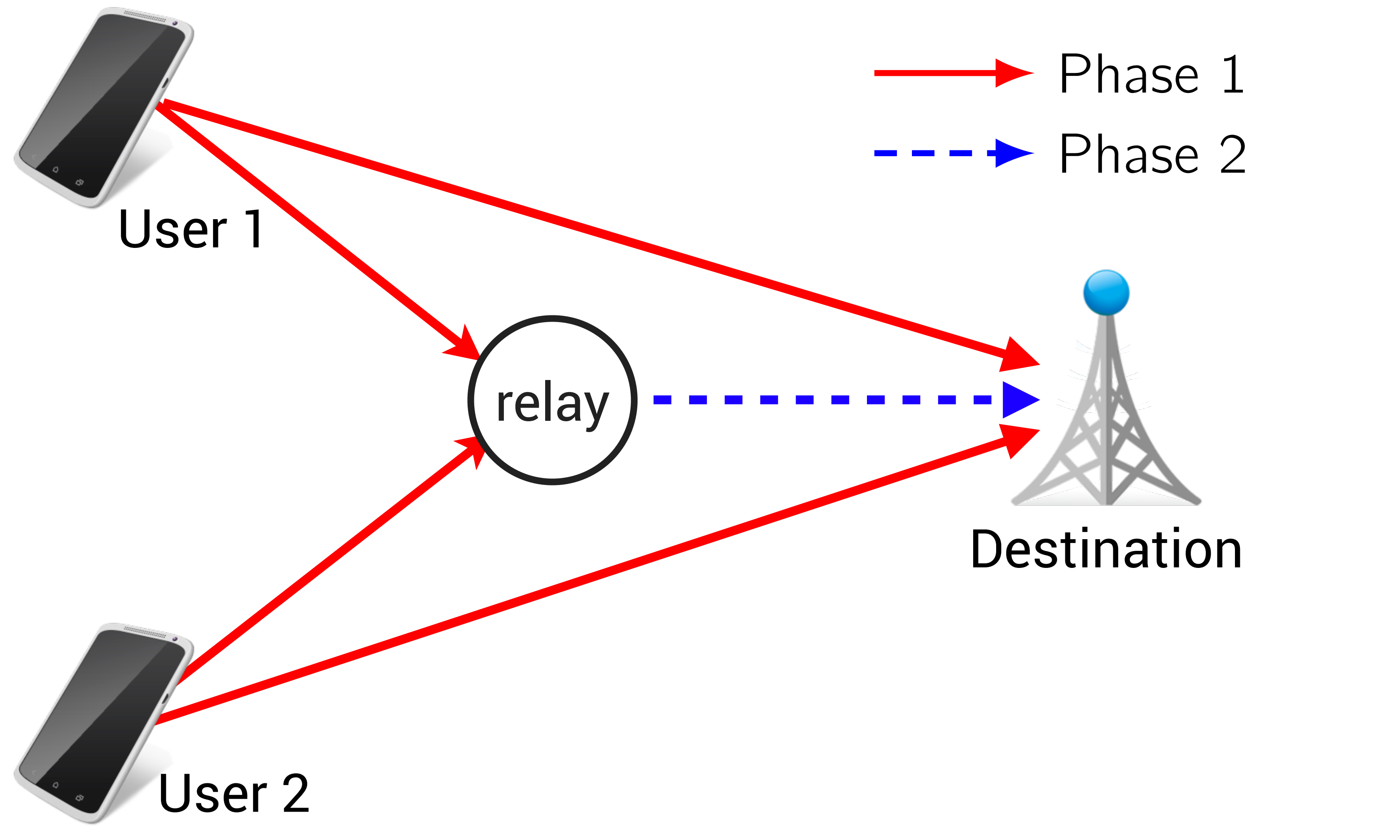
But $\mathbf{Q}$ may not be invertible, with significant probability.

Full cooperation protocol

Destination sends $\mathbf{q}$ vector to relay

Relay selects linearly independent q, to guarantee $\mathbf{Q}$ is full rank

Compute-Forward for Multiple Access Relay Channel



MARC — Multiple Access Relay Channel
User 1 and User 2 each send one message to Destination with the help of Relay

A list Ficke-Pohst algorithm finds L best rates:

$$R(\mathbf{a}^*) \geq R(\mathbf{a}_2) \geq \cdots \geq R(\mathbf{a}_L)$$

and the corresponding coefficient vectors:

$$\mathbf{a}^*, \mathbf{a}_2, \ldots, \mathbf{a}_L$$

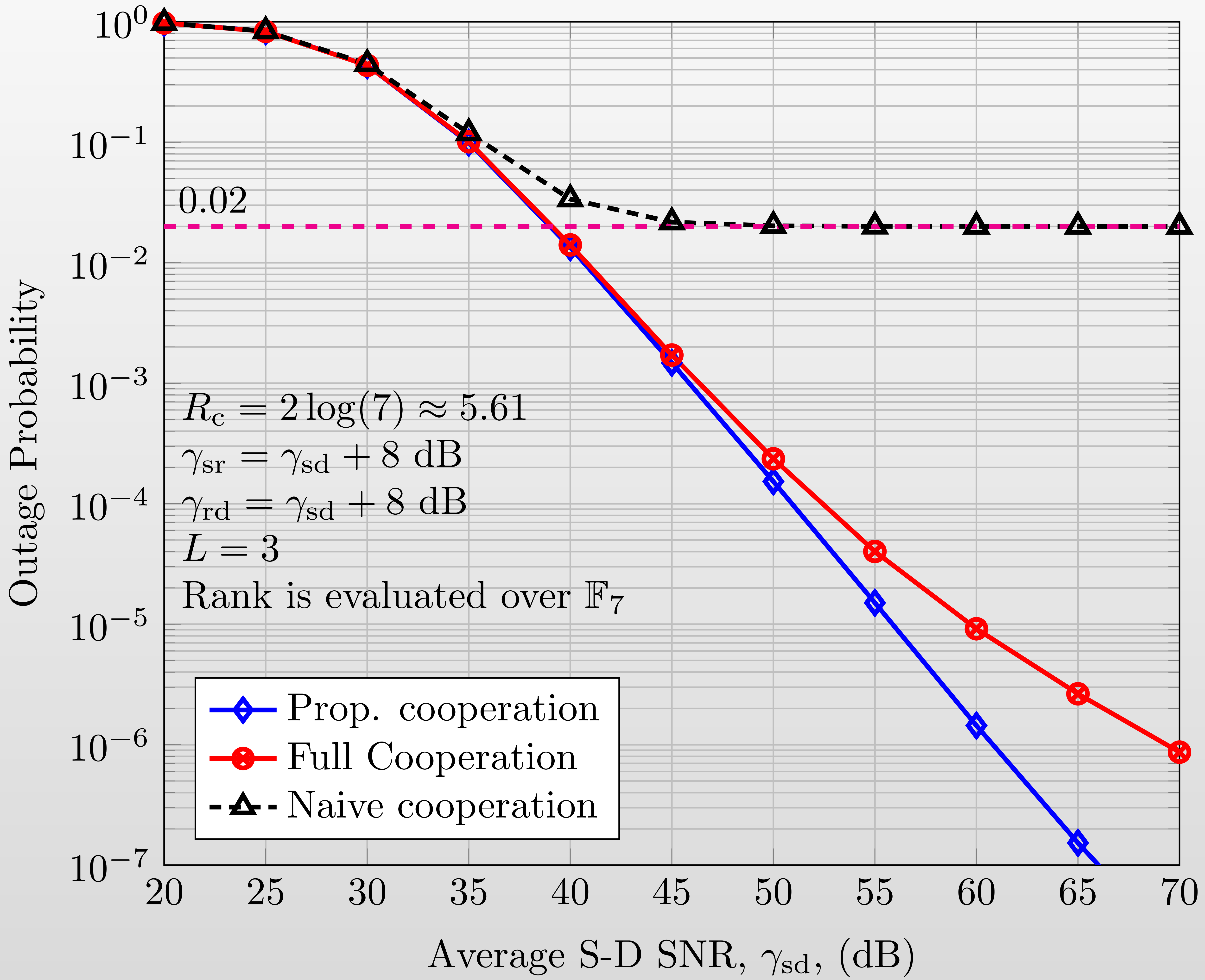
The destination attempts to decode using the two best $\mathbf{a}$'s

Proposal: Form Multiple Linear Combinations at Destination

1. Destination attempts to decode both u and u by forming linearly independent combinations. Relay does nothing.
2. If this fails, destination sends a* to relay. Relay chooses its best linearly independent combination. Using this, data is transmitted from relay to destination.

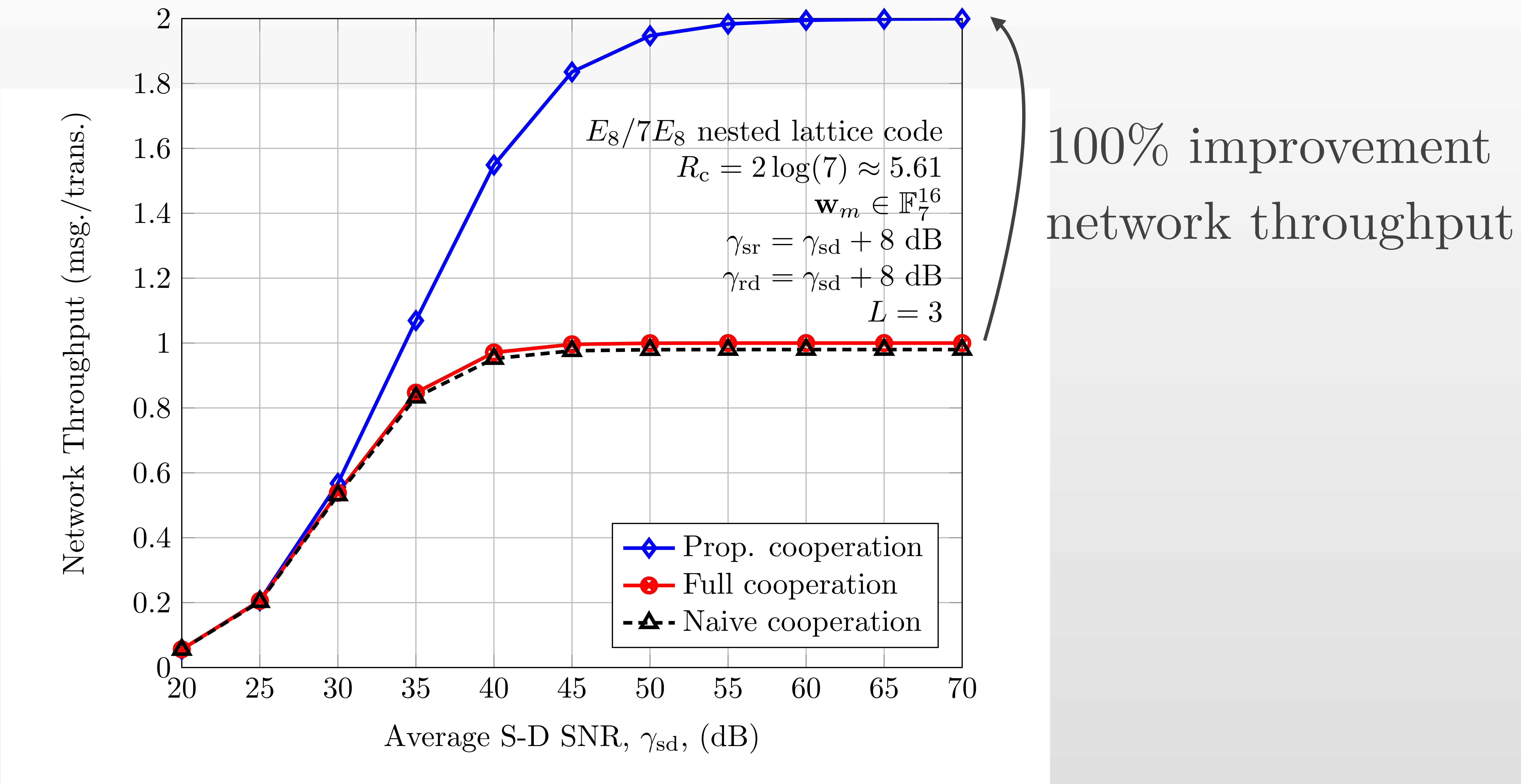
Proposed Method has Lower Outage Probability

- Maximum diversity order of 2 (competing systems have diversity order less than 2)



[HK17] M. N. Hasan and B. M. Kurkoski, “Practical Compute-and-Forward approaches for the multiple access relay channel,” IEEE ICC, (Paris, France), May 2017

100% increase in throughput



- Network throughput increases 100% [HK18]

[HK18] M. N. Hasan and B. M. Kurkoski, “Cooperation protocols for multiple access relay channel with compute-and-forward,” Submitted to IEEE Trans. Comm.

Conclusion

Central question: How might lattices effectively be used in wireless communication systems?

Lattices with practical encoding and decoding are needed — Construction D′ using QC-LDPC codes is a strong candidate

Lattices can provide shaping gain which is difficult otherwise — Convolutional code lattices provide > 1.0 dB of shaping gain

Physical layer network coding provides significant throughput benefit — lattices enable PLNC