Distributed Computation in Networks, Part I
Compute-and-Forward

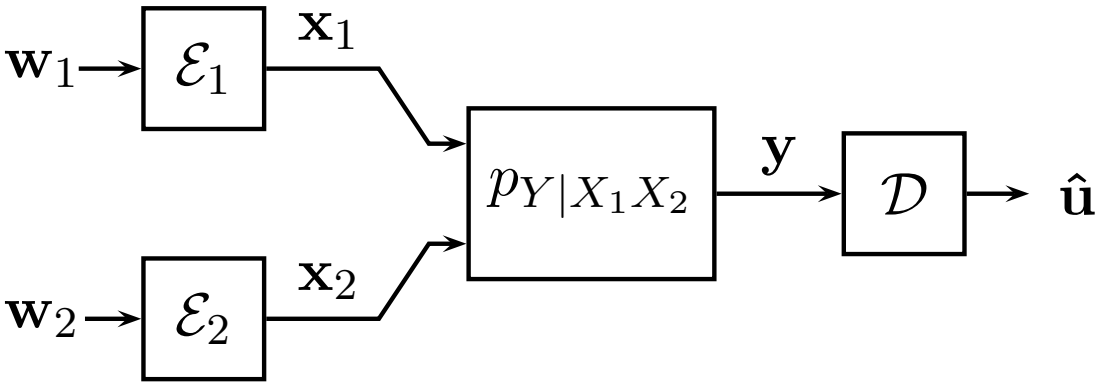
Michael Gastpar

Acknowledgement: Bobak Nazer, Jingge Zhu

Spring School, Paris, May 2014

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

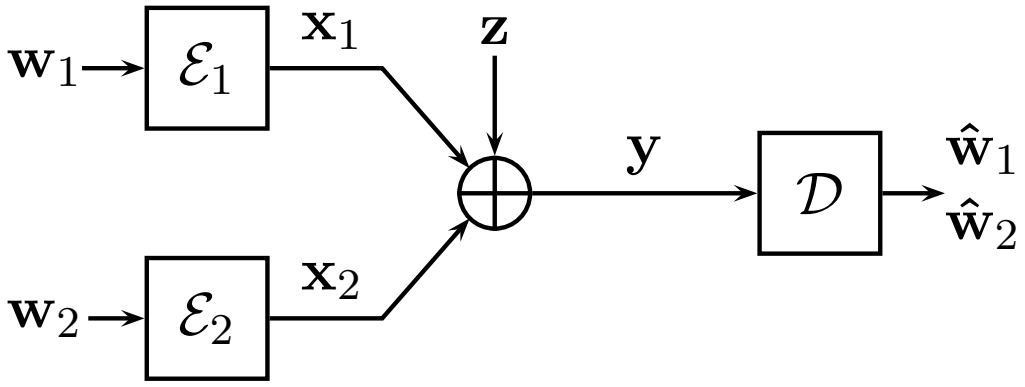
Outline

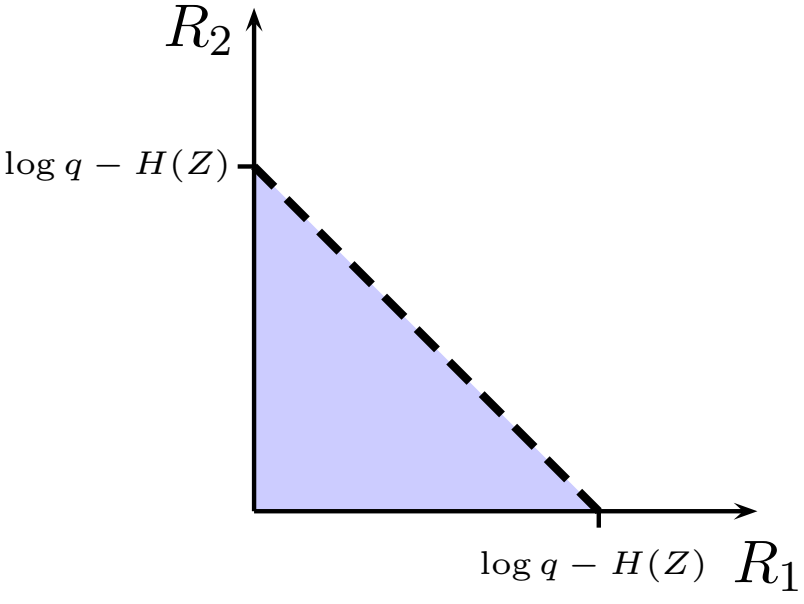


- **Rate Region:** Set of rates (R_1, R_2) such that the decoder can recover $f(\mathbf{w}_1, \mathbf{w}_2)$ with vanishing probability of error

$$\mathbb{P}\{\hat{\mathbf{u}} \neq f(\mathbf{w}_1, \mathbf{w}_2)\} \to 0 \text{ as } m \to \infty$$

Finite-Field Multiple-Access Channels





- Receiver observes noisy modulo sum of codewords $\mathbf{y} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \mathbf{z}$

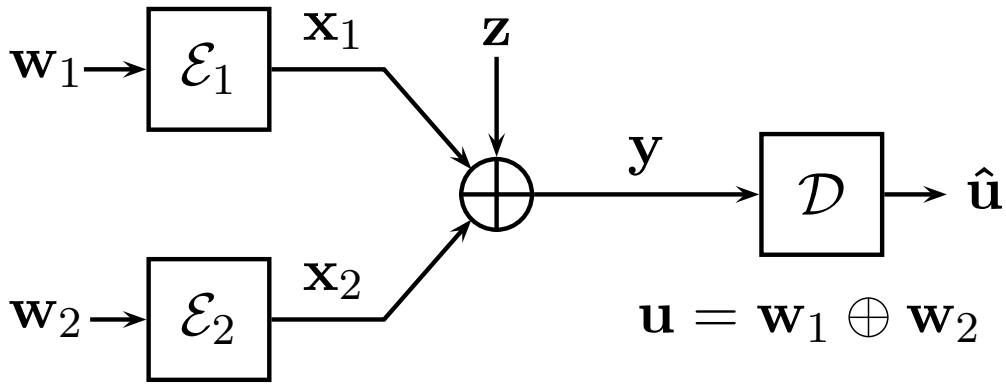
Finite Field MAC Rate Region

All rates (R_1, R_2) satisfying

$$R_1 + R_2 \leq \log q - H(Z)$$

Computation over Finite Field Multiple-Access Channels

- Independent msgs $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_q^k$.
- Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}\{\hat{\mathbf{u}} \neq \mathbf{u}\} \to 0$

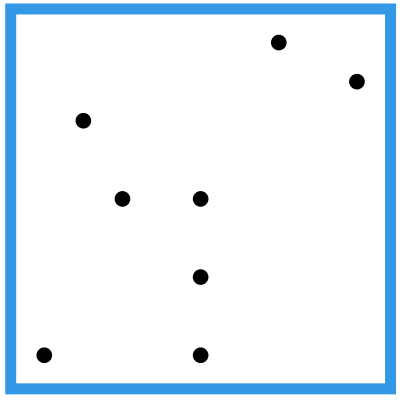


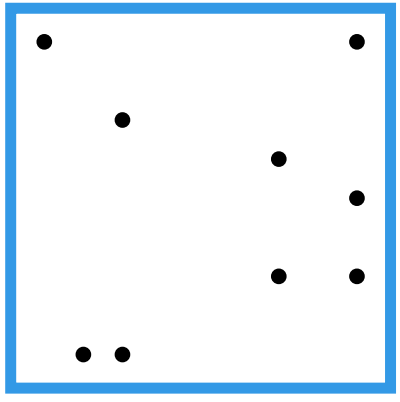
I.I.D. Random Coding

- Generate 2^{nR_1} i.i.d. uniform codewords for user 1.
- Generate 2^{nR_2} i.i.d. uniform codewords for user 2.
- With high probability, (nearly) all sums of codewords are distinct.
- This is ideal for multiple-access but not for computation.
- Need $R_1 + R_2 \leq \log q - H(Z)$

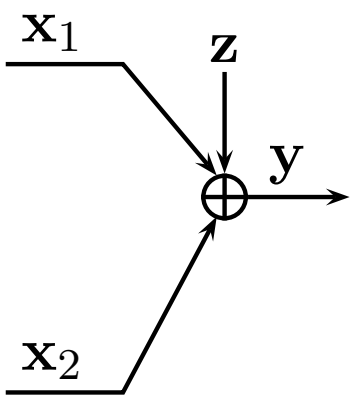
Random i.i.d. codes are not good for computation

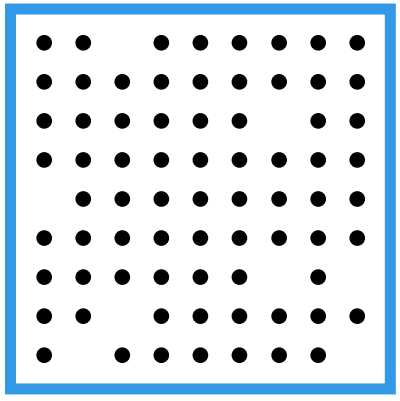
2^{nR_1} codewords





2^{nR_2} codewords

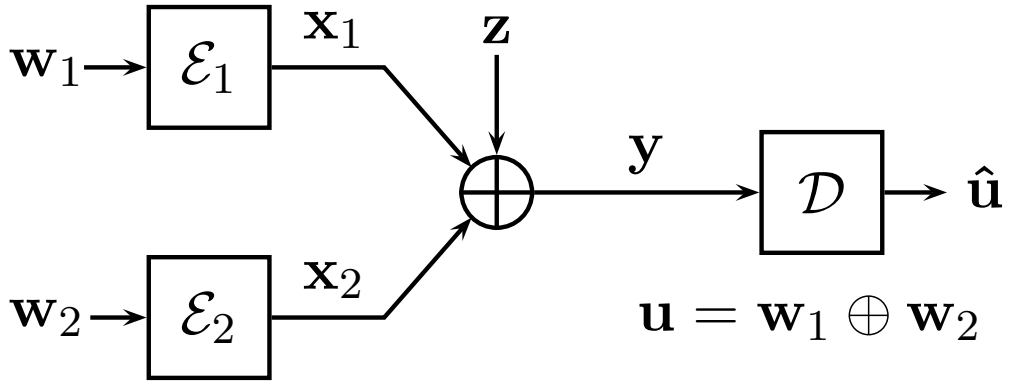




$2^{n(R_1+R_2)}$ modulo sums of codewords

Independent msgs $\mathbf{w}_1, \mathbf{w}_2$.

Want the sum $\mathbf{u} = \mathbf{w}_1 \oplus \mathbf{w}_2$ with vanishing prob. of error $\mathbb{P}\{\hat{\mathbf{u}} \neq \mathbf{u}\} \to 0$



Random Linear Coding

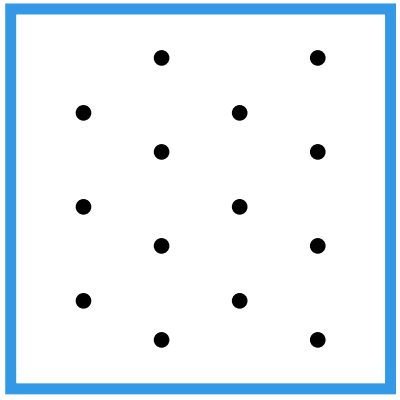
- Same linear code at both transmitters $\mathbf{x}_1 = \mathbf{G}\mathbf{w}_1$, $\mathbf{x}_2 = \mathbf{G}\mathbf{w}_2$.
- Sums of codewords are themselves codewords:

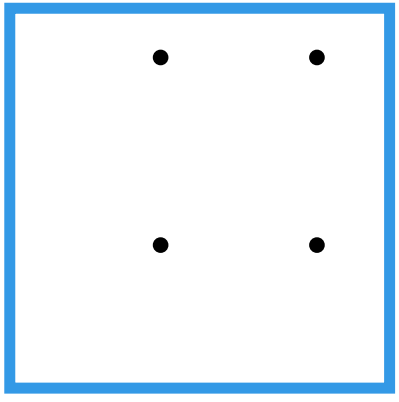
$$\begin{aligned}\mathbf{y} &= \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{w}_1 \oplus \mathbf{G}\mathbf{w}_2 \oplus \mathbf{z} \\ &= \mathbf{G}(\mathbf{w}_1 \oplus \mathbf{w}_2) \oplus \mathbf{z} \\ &= \mathbf{G}\mathbf{u} \oplus \mathbf{z}\end{aligned}$$

- Need $\max(R_1, R_2) \leq \log q - H(Z)$

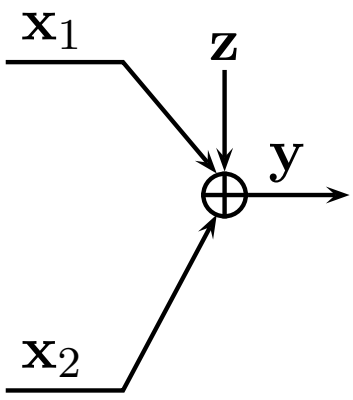
Random linear codes are good for computation

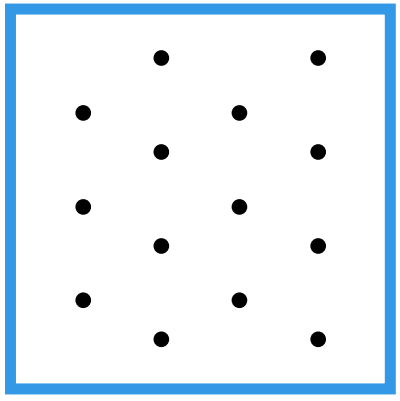
2^{nR_1} codewords



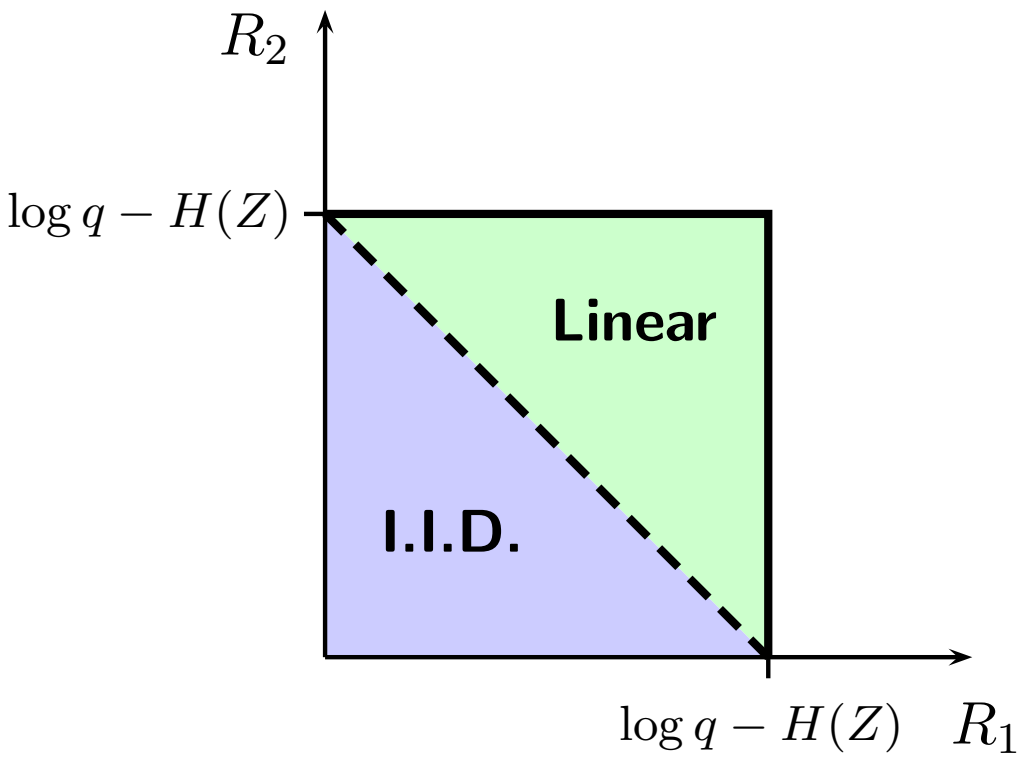


2^{nR_2} codewords





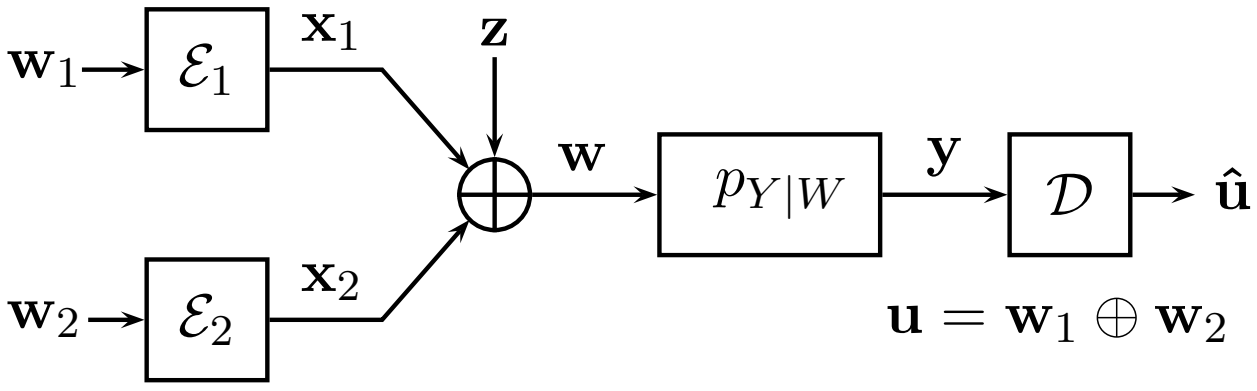
$2^{n \max(R_1, R_2)}$ modulo sums of codewords



- **I.I.D. Random Coding**: $R_1 + R_2 \leq \log q - H(Z)$
- **Random Linear Coding**: $\max(R_1, R_2) \leq \log q - H(Z)$
- Linear codes double the sum rate *without any dependency*.
- Is this useful for *sending messages* (no computation)?

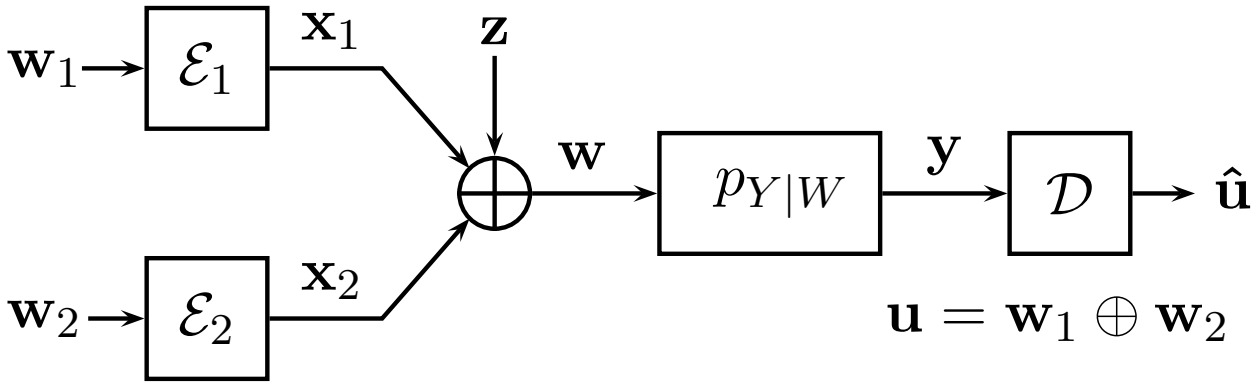
Computation over More General MACs

- Consider the following model:



- Find an achievable computation rate.

Computation over More General MACs



- One possibly interesting rate is attained by using the same binary linear code at both transmitters.
- Then, the resulting computation rate is simply $R = I(W;Y)$, where W is Bernoulli(1/2).

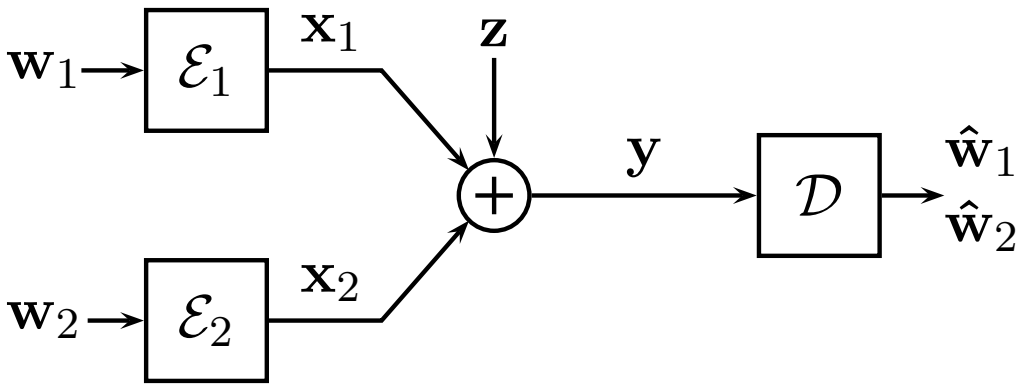
Gaussian Multiple-Access Channel: Capacity

Rate Region

$$R_1 < \frac{1}{2}\log\left(1+\frac{P_1}{N}\right)$$

$$R_2 < \frac{1}{2}\log\left(1+\frac{P_2}{N}\right)$$

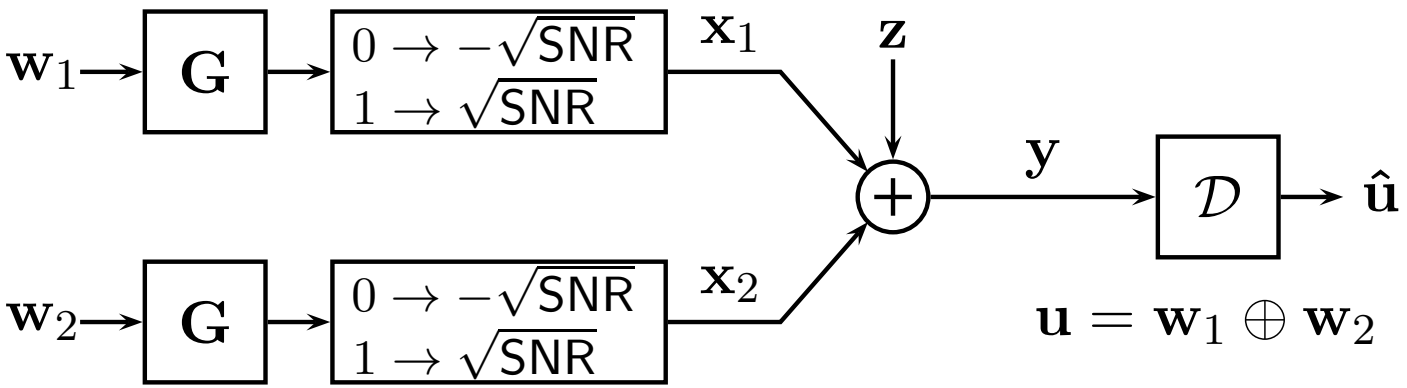
$$R_1+R_2 < \frac{1}{2}\log\left(1+\frac{P_1+P_2}{N}\right)$$



Power constraints P_1, P_2. Noise variance N.

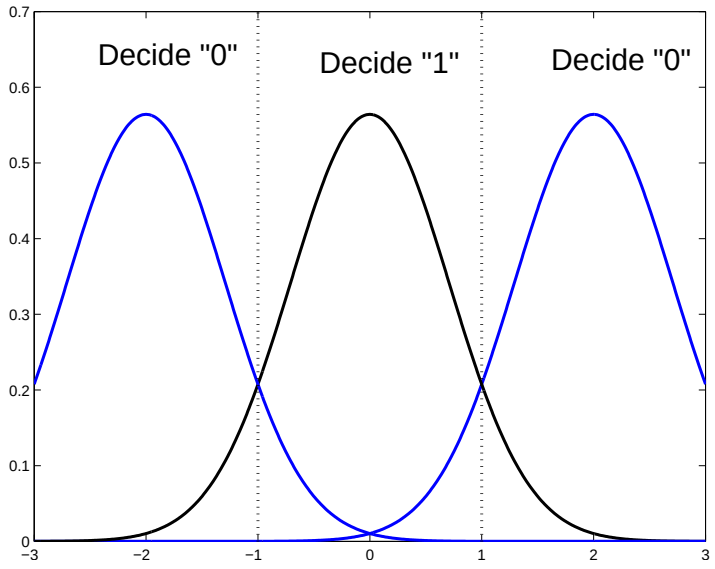
How can we extend this to the Gaussian Multiple-Access Channel?

- Let us assume $P = P_1 = P_2$, and introduce, for simplicity, $\mathsf{SNR} = P/N$.
- Then, consider the following simple approach for the computation problem:



Gaussian Multiple-Access Channels: Mod-2 Sum Computation?

Let us use a simple sub-optimal decoding step:



- Step 1: for each symbol, decide between $-2, 0, 2$.
- Step 2: Map: 0 to 1, and both -2 and 2 to 0. Overall, this leads to a binary asymmetric channel.
- Step 3: ML decoding with respect to the code.

Exercise: Calculate the rate at which we can decode the modulo-2 sum.

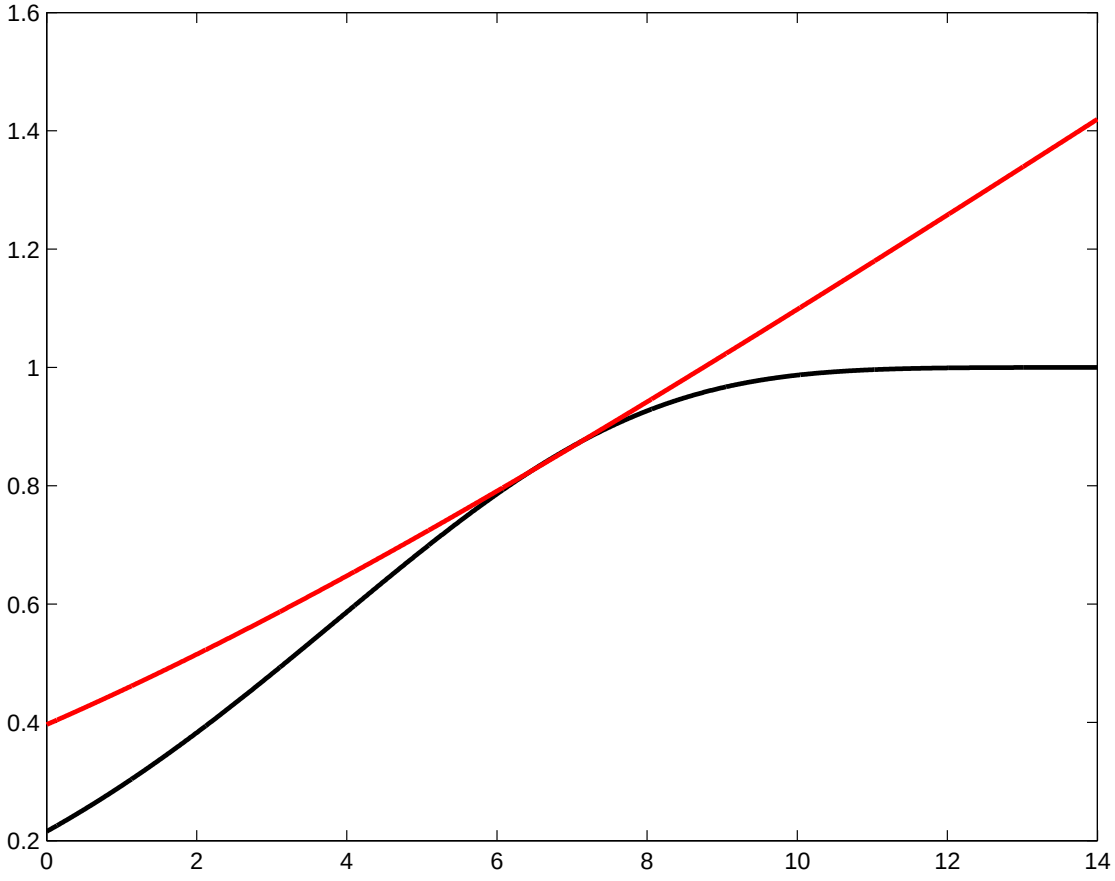
Exercise: Calculate the rate at which we can decode the modulo-2 sum.

Solution:

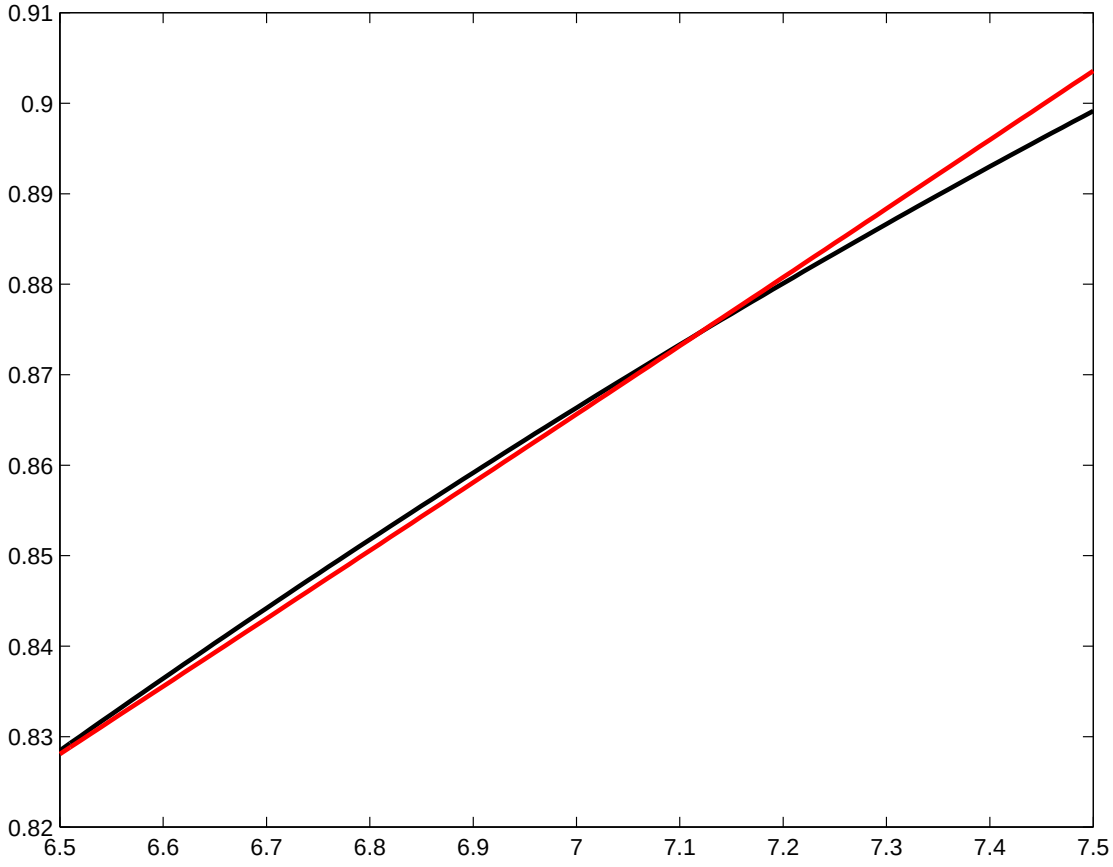
- Since both users use the *same code*, the overall scenario can be thought of as a *point-to-point channel* with a binary input.
- 0 is flipped to 1 with probability $Q(\sqrt{\mathsf{SNR}}) - Q(3\sqrt{\mathsf{SNR}})$.
- 1 is flipped to 0 with probability $2Q(\sqrt{\mathsf{SNR}})$.
- On this channel, we are using a *uniform input distribution.*
- Hence, the rate is equal to the mutual information across this channel, evaluated for uniform inputs.

Gaussian Multiple-Access Channels: Mod-2 Sum Computation?

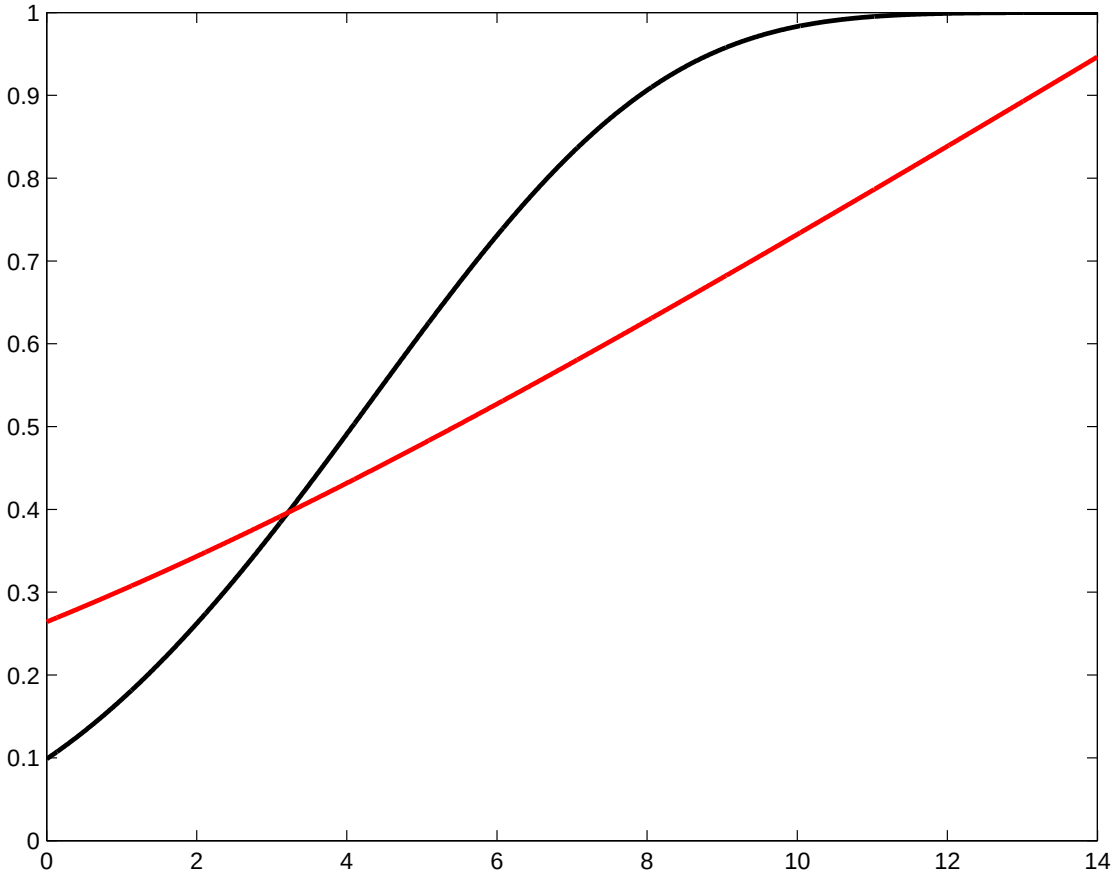
Two users:



Two users, detail:



Three users:



The Role of Alphabet ("field") Size

- In binary, the rate can be at most one.
- This may be acceptable in low-SNR.
- In high SNR, it appears inevitable to consider larger alphabets...

Outline

Lattices

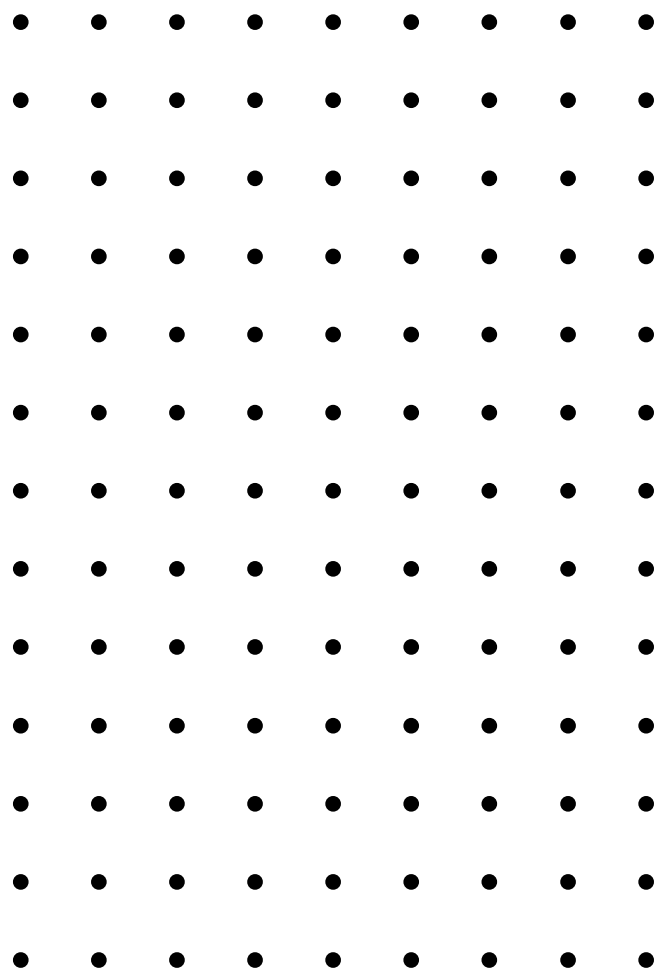
- A lattice Λ is a discrete subgroup of $\mathbb{R}^n$.
- Can write a lattice as a linear transformation of the integer vectors,

$$\Lambda = \mathbf{B}\mathbb{Z}^n \ ,$$

for some $\mathbf{B} \in \mathbb{R}^{n \times n}$.

Lattice Properties

- Closed under addition: $\lambda_1, \lambda_2 \in \Lambda \implies \lambda_1 + \lambda_2 \in \Lambda$.
- Symmetric: $\lambda \in \Lambda \implies -\lambda \in \Lambda$



$\mathbb{Z}^n$ is a simple lattice.

Voronoi Regions

- Nearest neighbor quantizer:

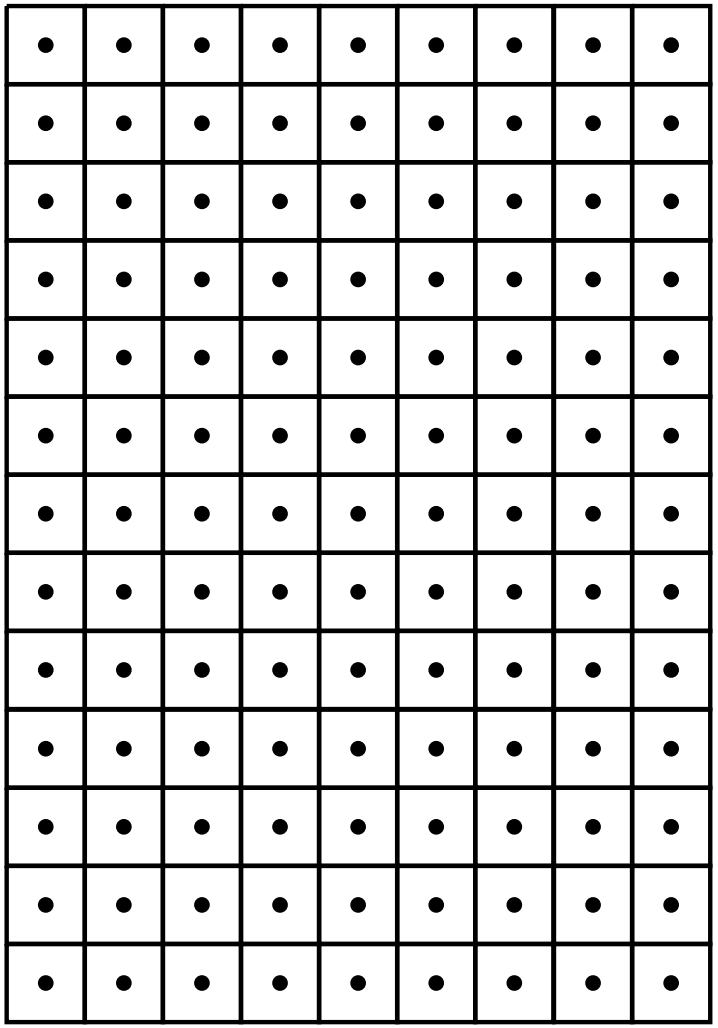
$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|_2$$

- The Voronoi region of a lattice point is the set of all points that quantize to that lattice point.

- Fundamental Voronoi region $\mathcal{V}$: points that quantize to the origin,

$$\mathcal{V} = \{\mathbf{x} : Q_\Lambda(\mathbf{x}) = \mathbf{0}\}$$

- Each Voronoi region is just a shift of the fundamental Voronoi region $\mathcal{V}$



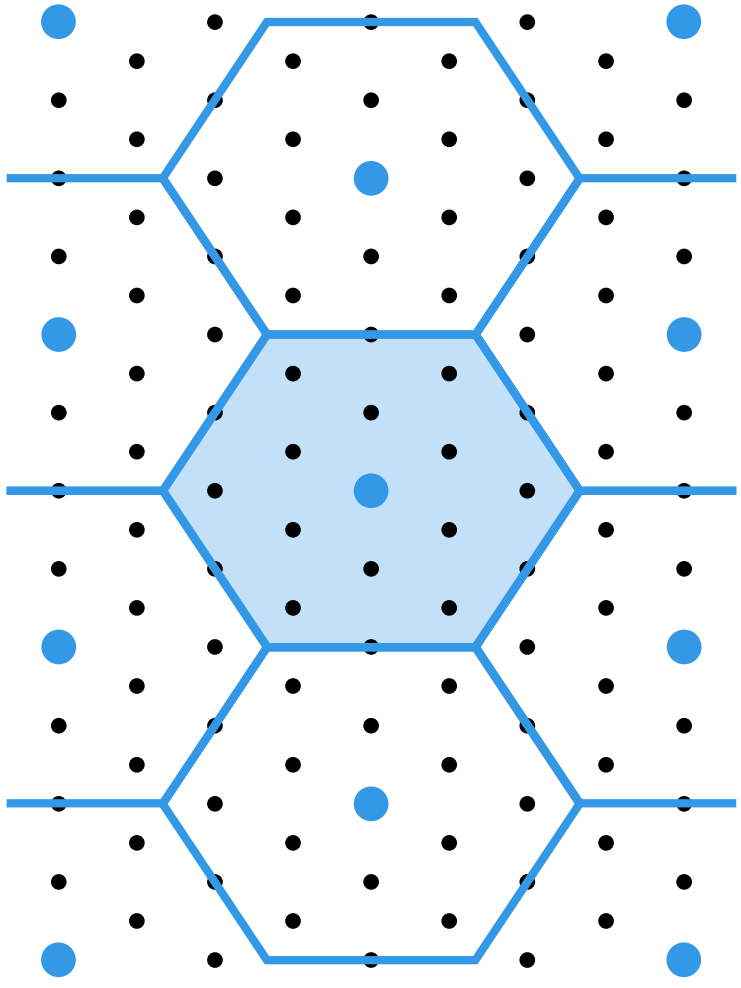
Lattice Tricks

1. Observing the power constraint: *Nested Lattices*
2. Proving achievable rates:
   - *Dithering*
   - *"MMSE Scaling"*

Nested Lattices

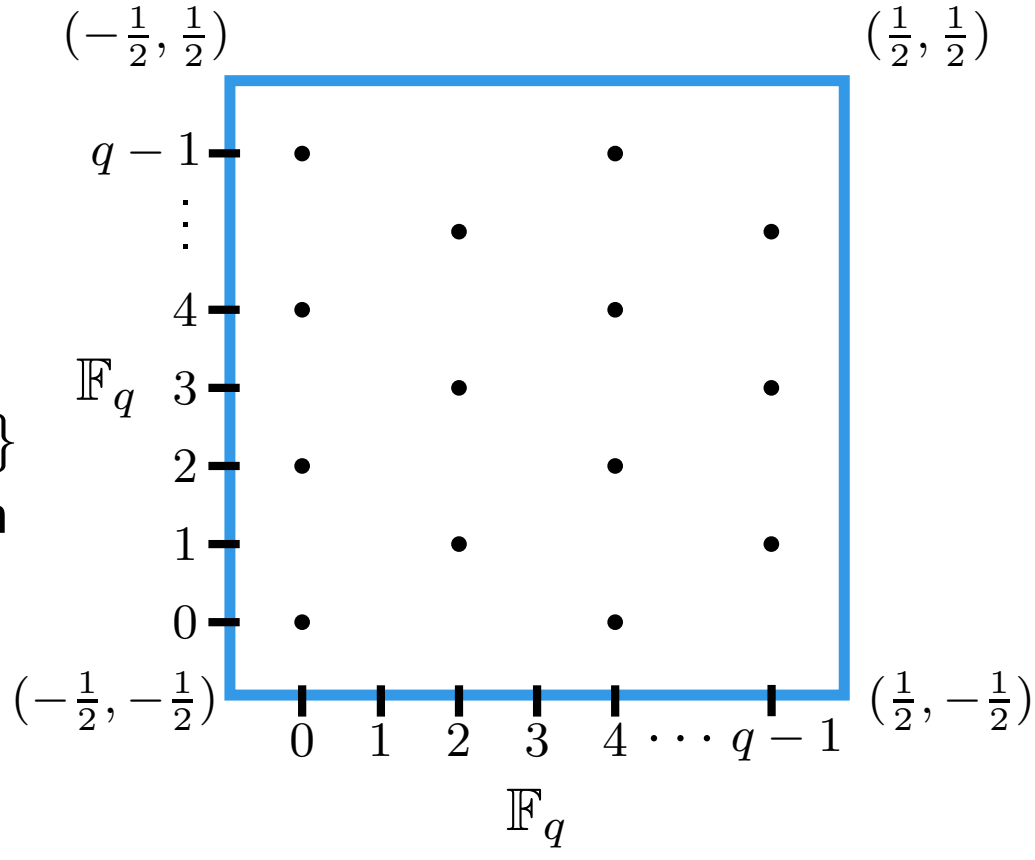
- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$
- **Nested Lattice Code:** All lattice points from Λ_{FINE} that fall in the fundamental Voronoi region $\mathcal{V}$ of Λ.
- $\mathcal{V}$ acts like a power constraint

$$\text{Rate} = \frac{1}{n} \log \left(\frac{\text{Vol}(\mathcal{V})}{\text{Vol}(\mathcal{V}_{\mathsf{FINE}})} \right)$$



Nested Lattice Codes from q-ary Linear Codes

- Choose an $n \times k$ generator matrix $\mathbf{G} \in \mathbb{F}_q^{n \times k}$ for q-ary code.
- Integers serve as coarse lattice, $\Lambda = \mathbb{Z}^n$.
- Map elements $\{0, 1, 2, \ldots, q-1\}$ to equally spaced points between $-1/2$ and $1/2$.
- Place codewords $\mathbf{x} = \mathbf{G}\mathbf{w}$ into the fundamental Voronoi region $\mathcal{V} = [-1/2, 1/2)^n$



Modulo Operation

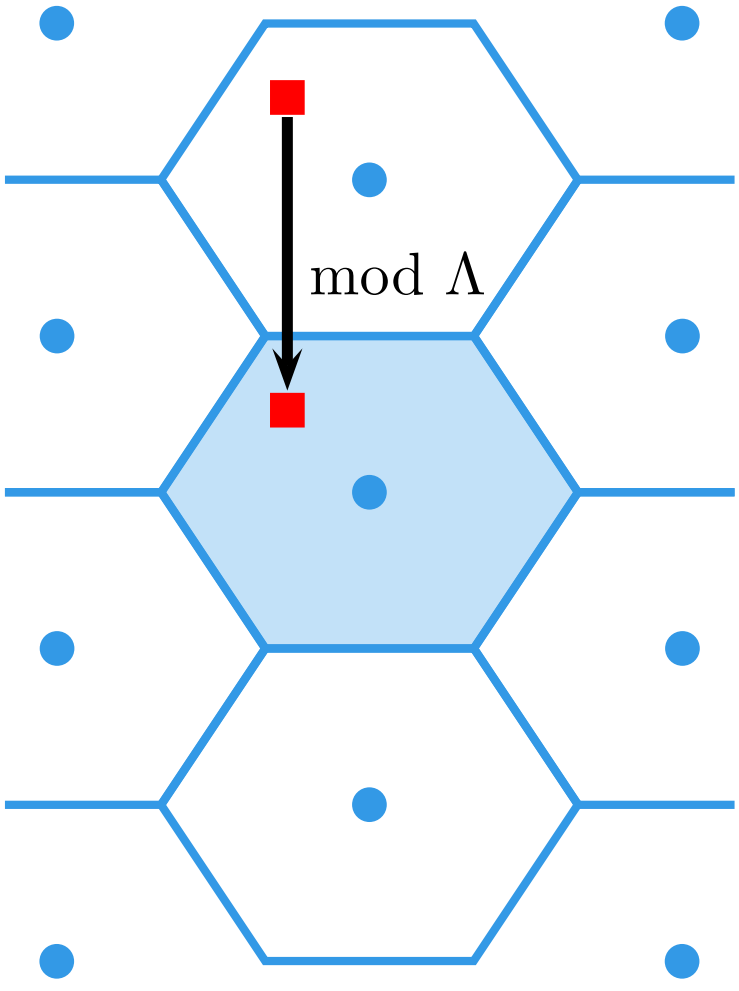
- Modulo operation with respect to lattice Λ is just the residual quantization error,

$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \ .$$

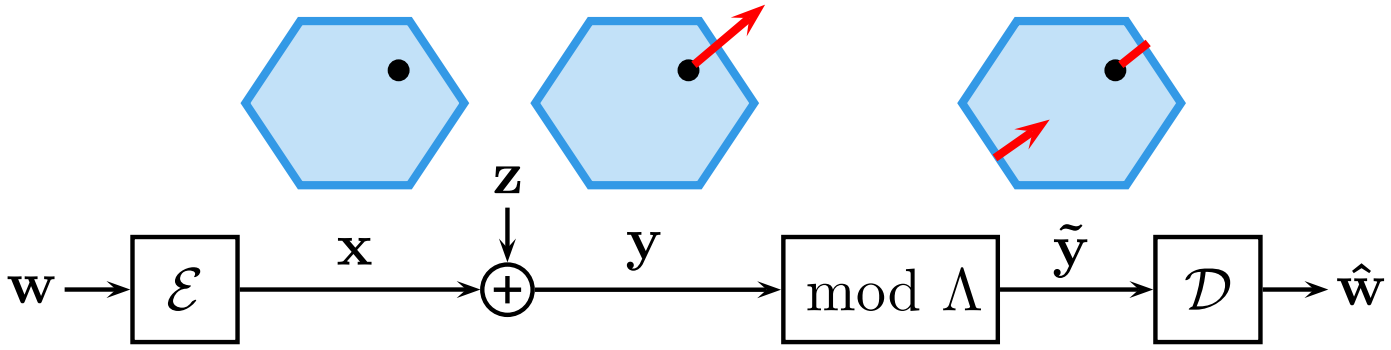
- Mimics the role of $\bmod q$ in q-ary alphabet.

- Distributive Law:

$$\Big[\mathbf{x}_1 + [\mathbf{x}_2] \bmod \Lambda\Big] \bmod \Lambda$$
$$= [\mathbf{x}_1 + \mathbf{x}_2] \bmod \Lambda$$

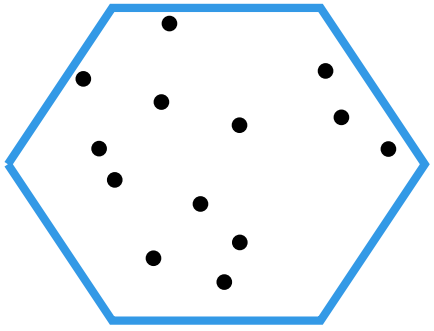


$\mathrm{mod}\ \Lambda$ *AWGN Channel*

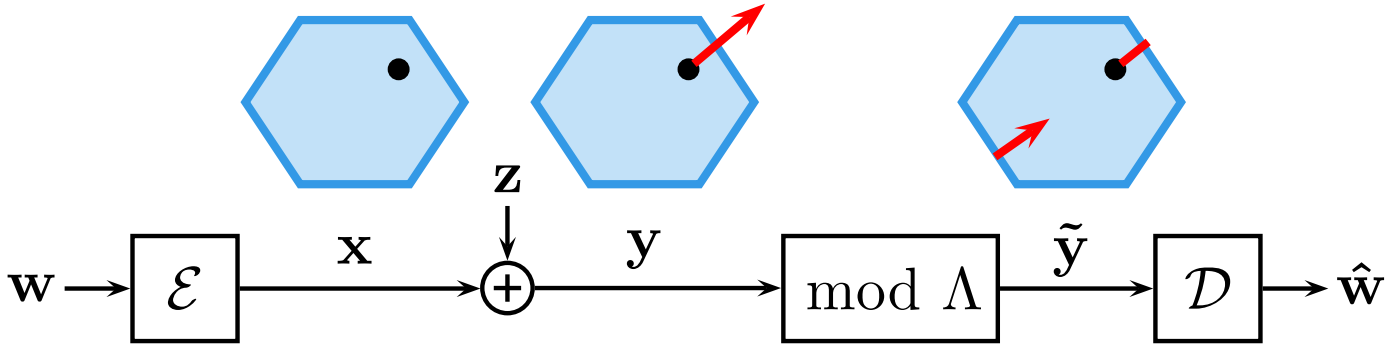


- Codebook lives on Voronoi region $\mathcal{V}$ of coarse lattice Λ.
- Take $\mathrm{mod}\ \Lambda$ of received signal prior to decoding.
- What is the capacity of the $\mathrm{mod}\ \Lambda$ channel?

Using random i.i.d. code drawn over $\mathcal{V}$: $$C = \frac{1}{n} \max_{p(\mathbf{x})} I(\mathbf{x}; \tilde{\mathbf{y}})$$

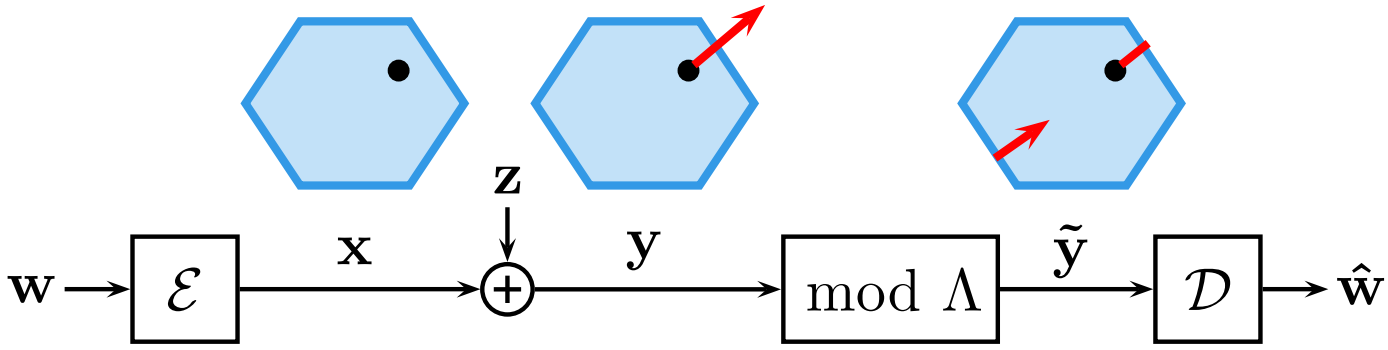


mod Λ *AWGN Channel Capacity*



$$
\begin{aligned}
nC &= \max_{p(\mathbf{x})} I(\mathbf{x};\tilde{\mathbf{y}}) \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h(\tilde{\mathbf{y}}|\mathbf{x}) \Big) \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h\Big([\mathbf{z}] \bmod \Lambda \Big) \Big) \qquad \text{Distributive Law} \\
&\geq \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - h(\mathbf{z}) \Big) \qquad \text{Point Symmetry of Voronoi Region} \\
&= \max_{p(\mathbf{x})} \Big(h(\tilde{\mathbf{y}}) - \frac{n}{2} \log(2\pi e N) \Big) \qquad \text{Entropy of Gaussian Noise}
\end{aligned}
$$

$\mathrm{mod}\ \Lambda$ *AWGN Channel Capacity*



- Channel output entropy is equal to the logarithm of the Voronoi region volume if it is uniform over $\mathcal{V}$:

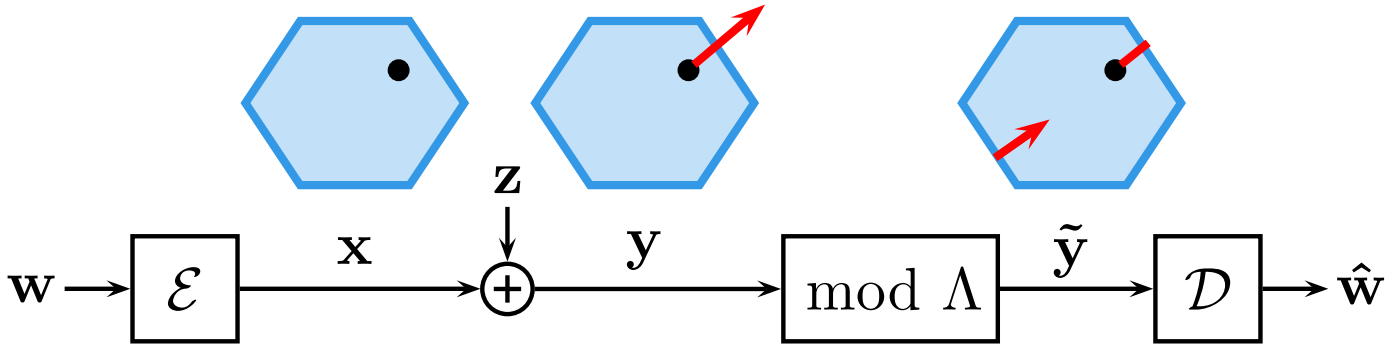
$$h(\tilde{\mathbf{y}}) = \log(\mathrm{Vol}(\mathcal{V})) \qquad \text{if } \tilde{\mathbf{y}} \sim \mathrm{Unif}(\mathcal{V})$$

- $\tilde{\mathbf{y}} = [\mathbf{x} + \mathbf{z}] \bmod \Lambda$ is uniform over $\mathcal{V}$ if $\mathbf{x}$ is uniform over $\mathcal{V}$.

- Random i.i.d. coding over the Voronoi region $\mathcal{V}$ can achieve:

$$R = \frac{1}{n} \log(\mathrm{Vol}(\mathcal{V})) - \frac{1}{2} \log(2\pi e N)$$

Power Constraints and Second Moments

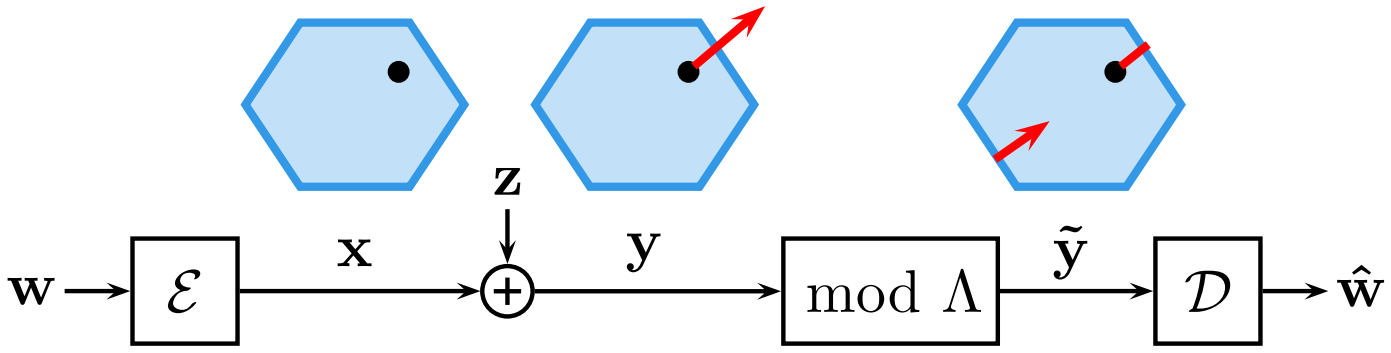


- Must scale lattice Λ so that the uniform distribution over the Voronoi region $\mathcal{V}$ meets the power constraint P.
- Set second moment $\sigma_\Lambda^2 = \dfrac{1}{n\text{Vol}(\mathcal{V})}\displaystyle\int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}$ equal to P.

Normalized Second Moment: $$G(\Lambda) = \frac{\sigma_\Lambda^2}{(\text{Vol}(\mathcal{V}))^{2/n}}$$

$$\implies \frac{1}{n}\log(\text{Vol}(\mathcal{V})) = \frac{1}{2}\log\left(\frac{\sigma_\Lambda^2}{G(\Lambda)}\right) = \frac{1}{2}\log\left(\frac{P}{G(\Lambda)}\right)$$

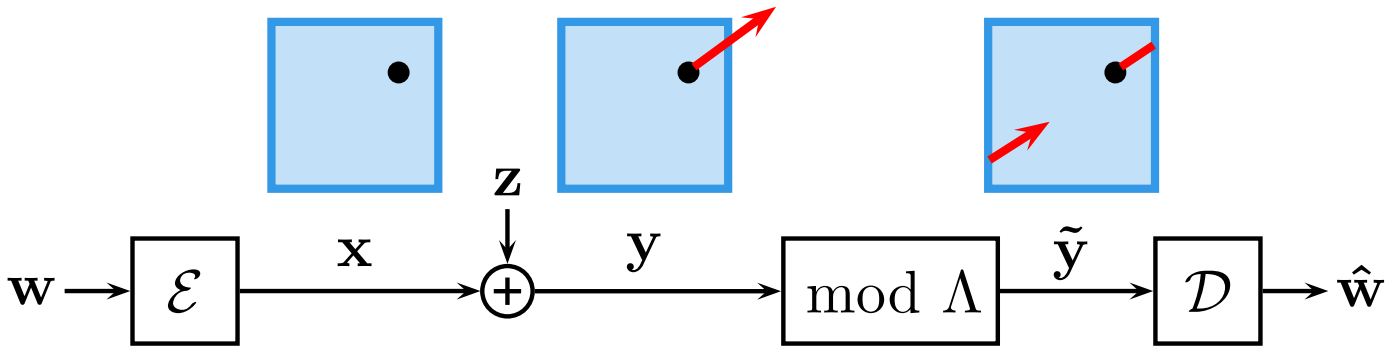
mod Λ *AWGN Channel Capacity*



- Random i.i.d. coding over the Voronoi region $\mathcal{V}$ can achieve:

$$\begin{aligned} C &\geq \frac{1}{n}\log(\mathsf{Vol}(\mathcal{V})) - \frac{1}{2}\log(2\pi eN) \\ &= \frac{1}{2}\log\left(\frac{P}{G(\Lambda)}\right) - \frac{1}{2}\log(2\pi eN) \\ &= \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log(2\pi eG(\Lambda)) \end{aligned}$$

What is $G(\Lambda)$?



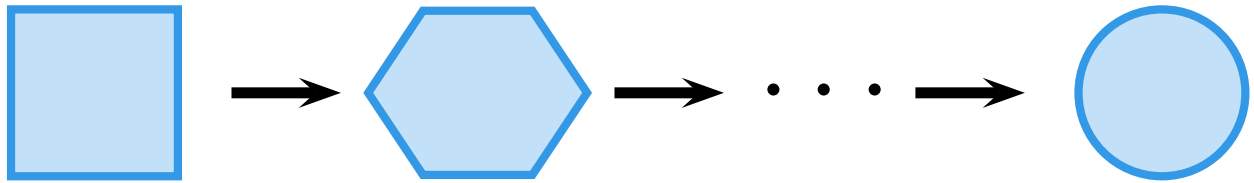
- The normalized second moment $G(\Lambda)$ is a dimensionless quantity that captures the shaping gain.
- Integer lattice is not so bad, $G(\mathbb{Z}^n) = 1/12$.
- Capacity under $\mod \mathbb{Z}^n$ is at least

$$C \geq \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$
$$\approx \frac{1}{2}\log\left(\frac{P}{N}\right) - 0.255$$

Asymptotically Good $G(\Lambda)$

Theorem (Zamir-Feder-Poltyrev '94)

There exists a sequence of lattices $\Lambda^{(n)}$ such that $\lim_{n\to\infty} G(\Lambda^{(n)}) = \dfrac{1}{2\pi e}$.



$n = 1$ $\qquad$ $n = 2$ $\qquad$ $n \to \infty$

- Best possible normalized second moment is that of a sphere.
- Using a sequence $\Lambda^{(n)}$ with an asymptotically good $G(\Lambda^{(N)})$ allows to approach

$$\begin{aligned} R &= \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{2\pi e}\right) \\ &= \frac{1}{2}\log\left(\frac{P}{N}\right) \end{aligned}$$

Asymptotically Good $G(\Lambda)$

- Can actually get this with a linear code tiled over $\mathbb{Z}^n$ (see, for instance, **Erez-Litsyn-Zamir '05**.)
- Many works looking at this from different perspectives.
- We will just assume existence.

Recall the two key properties of random linear codes $\mathbf{G}$ from earlier:

Codeword Properties

1. Marginally uniform over $\mathbb{F}_q^n$. For a given message $\mathbf{w} \neq \mathbf{0}$, the codeword $\mathbf{x} = \mathbf{G}\mathbf{w}$ looks like an i.i.d. uniform sequence.

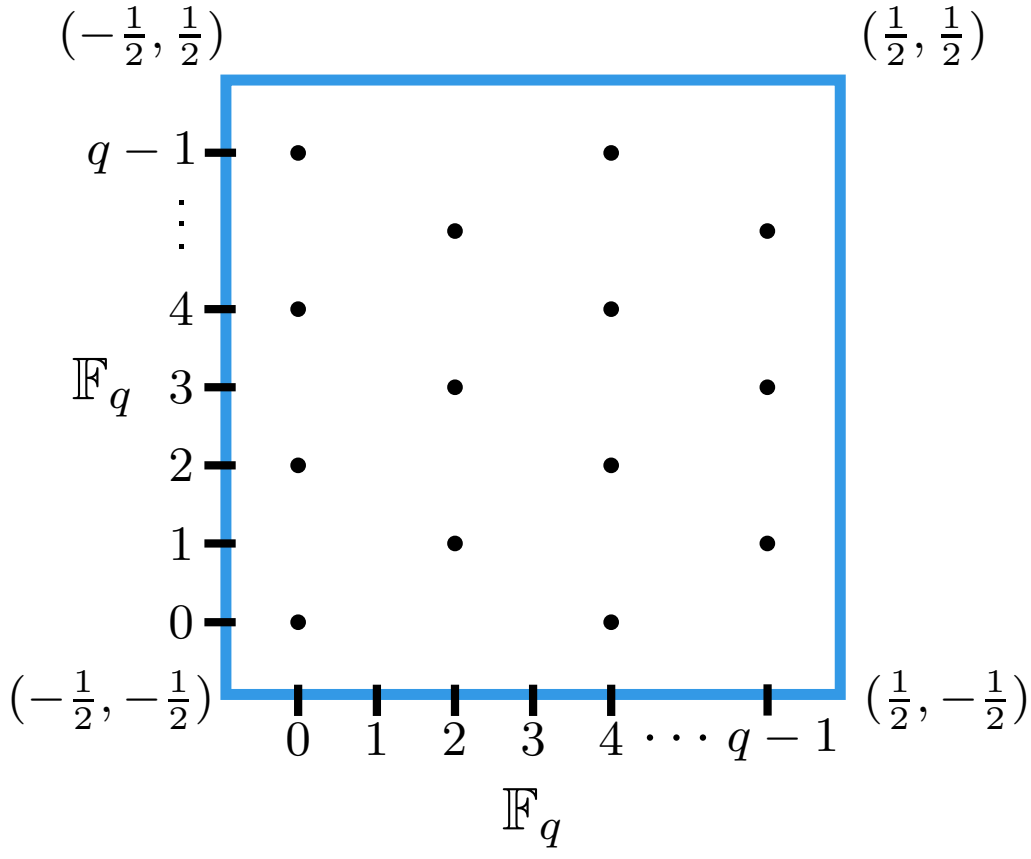
$$\mathbb{P}\{\mathbf{x} = \mathsf{x}\} = \frac{1}{q^n} \quad \text{for all } \mathsf{x} \in \mathbb{F}_q^n$$

2. Pairwise independent. For $\mathbf{w}_1, \mathbf{w}_2 \neq \mathbf{0}$, $\mathbf{w}_1 \neq \mathbf{w}_2$, codewords $\mathbf{x}_1, \mathbf{x}_2$ are independent.

$$\mathbb{P}\{\mathbf{x_1} = \mathsf{x}_1, \mathbf{x_2} = \mathsf{x}_2\} = \frac{1}{q^{2n}} = \mathbb{P}\{\mathbf{x}_1 = \mathsf{x}_1\}\mathbb{P}\{\mathbf{x}_2 = \mathsf{x}_2\}$$

Linear Codes for $\mathrm{mod}\ \Lambda$ *Channels*

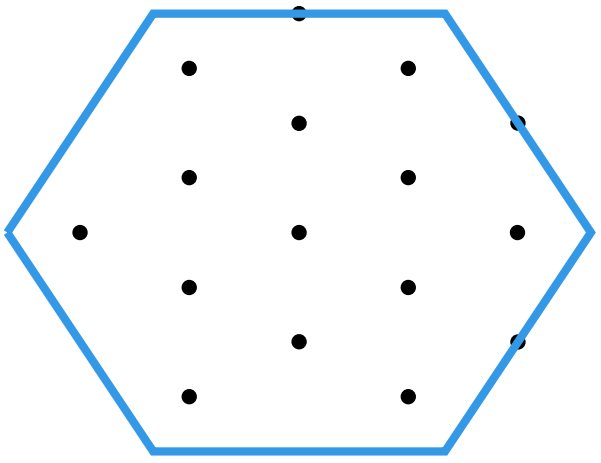
- Instead of an "inner" random codes, we can use a q-ary linear code.
- This is exactly a nested lattice.
- Each codeword has a uniform marginal distribution over the grid.
- Rate loss due to finite constellation which goes to 0 as $q \to \infty$.
- Codewords are pairwise independent so we can apply the union bound.



$$\mathbf{x} = [\gamma \mathbf{G}\mathbf{w}] \mod \mathbb{Z}^n$$

Linear Codes for $\mod \Lambda$ *Channels*

- General coarse lattice $\Lambda = \mathbf{B}\mathbb{Z}^n$.
- First, apply generator matrix for linear code $\mathbf{Gw}$. Then scale down by γ and tile over $\mathbb{Z}^n$.
- Multiply by $\mathbf{B}$ and apply $\mod \Lambda$ to get codebook.
- As q gets large, each codeword's marginal distribution looks uniform over $\mathcal{V}$.
- Codewords are pairwise independent so we can apply the union bound.



$$\mathbf{x} = [\mathbf{B}\gamma\mathbf{Gw}] \mod \Lambda$$

MMSE Scaling

- **Erez-Zamir '04:** Prior to taking $\bmod \Lambda$, scale by α.

$$\begin{aligned}\tilde{\mathbf{y}} &= [\alpha \mathbf{y}] \bmod \Lambda \\ &= [\alpha \mathbf{x} + \alpha \mathbf{z}] \bmod \Lambda \\ &= [\mathbf{x} + \underbrace{\alpha \mathbf{z} - (1-\alpha)\mathbf{x}}_{\text{Effective Noise}}] \bmod \Lambda\end{aligned}$$

- For now, ignore that the effective noise is not independent of the codeword. Effective noise variance $N_{\mathsf{EFFEC}} = \alpha^2 N + (1-\alpha)^2 P$.
- Optimal choice of α is the MMSE coefficient $\alpha_{\mathsf{MMSE}} = \dfrac{P}{N+P}$.

$$N_{\mathsf{EFFEC}} = \alpha_{\mathsf{MMSE}}^2 N + (1-\alpha_{\mathsf{MMSE}})^2 P = \frac{PN}{N+P}$$

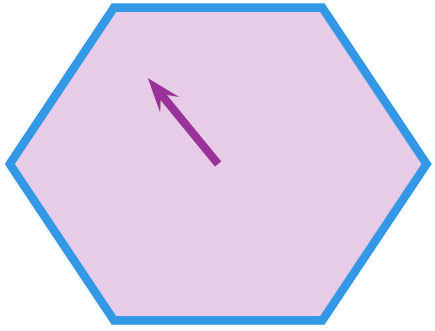
$$C = \frac{1}{2}\log\left(\frac{P}{N_{\mathsf{EFFEC}}}\right) = \frac{1}{2}\log\left(1+\frac{P}{N}\right)$$

Dithering

- Now the noise is dependent on the codeword.
- Dithering can solve this problem (just as in the discrete case).
- Map message $\mathbf{w}$ to a lattice codeword $\mathbf{t}$.
- Generate a random dither vector $\mathbf{d}$ uniformly over $\mathcal{V}$.
- Transmitter sends a dithered codeword:

$$\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda$$

- $\mathbf{x}$ is now independent of the codeword $\mathbf{t}$.



Decoding – Remove Dither First

- Transmitter sends dithered codeword $\mathbf{x} = [\mathbf{t} + \mathbf{d}] \bmod \Lambda$.
- After scaling the channel output $\mathbf{y}$ by α, the decoder subtracts the dither $\mathbf{d}$.

$$
\begin{aligned}
\tilde{\mathbf{y}} &= [\alpha\mathbf{y} - \mathbf{d}] \bmod \Lambda \\
&= [\alpha\mathbf{x} + \alpha\mathbf{z} - \mathbf{d}] \bmod \Lambda \\
&= [\mathbf{x} - \mathbf{d} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \\
&= \Big[[\mathbf{t} + \mathbf{d}] \bmod \Lambda - \mathbf{d} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}\Big] \bmod \Lambda \\
&= [\mathbf{t} + \alpha\mathbf{z} - (1-\alpha)\mathbf{x}] \bmod \Lambda \qquad \text{Distributive Law}
\end{aligned}
$$

- Effective noise is now independent from the codeword $\mathbf{t}$.
- By the probabilistic method, (at least) one good fixed dither exists. No common randomness necessary.

Summary

- Linear code embedded in the integer lattice:

$$R = \frac{1}{2}\log\left(\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$

- Linear code embedded in the integer lattice, MMSE scaling:

$$R = \frac{1}{2}\log\left(1+\frac{P}{N}\right) - \frac{1}{2}\log\left(\frac{2\pi e}{12}\right)$$

- Linear code embedded in a good shaping lattice, MMSE scaling:

$$R = \frac{1}{2}\log\left(1+\frac{P}{N}\right)$$

Theorem (Erez-Zamir '04)

Nested lattice codes can achieve the AWGN capacity.

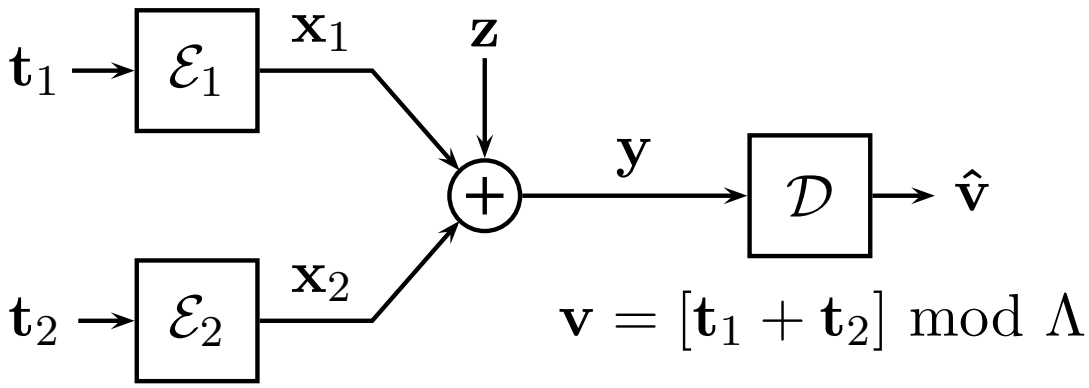
Outline

Decoding the Sum of Lattice Codewords

Encoders use the same nested lattice codebook.

Transmit lattice codewords:

$$\mathbf{x}_1 = \mathbf{t}_1$$
$$\mathbf{x}_2 = \mathbf{t}_2$$



Decoder recovers modulo sum.

$$\begin{aligned}
&[\mathbf{y}] \bmod \Lambda \\
&= [\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}] \bmod \Lambda \\
&= [\mathbf{t}_1 + \mathbf{t}_2 + \mathbf{z}] \bmod \Lambda \\
&= \Big[[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda + \mathbf{z}\Big] \bmod \Lambda \qquad \text{Distributive Law} \\
&= [\mathbf{v} + \mathbf{z}] \bmod \Lambda
\end{aligned}$$

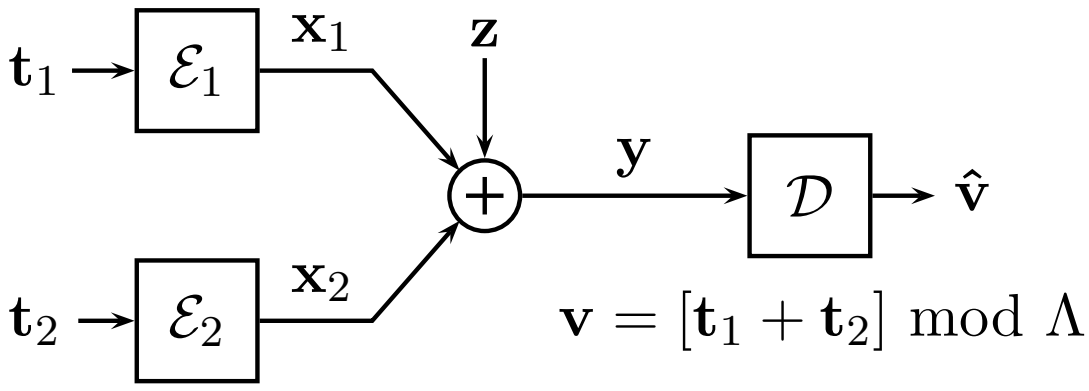
$$R = \frac{1}{2} \log \left(\frac{P}{N} \right)$$

Decoding the Sum of Lattice Codewords – MMSE Scaling

Encoders use the same nested lattice codebook.

Transmit dithered codewords:

$$\mathbf{x}_1 = [\mathbf{t}_1 + \mathbf{d}_1] \bmod \Lambda$$
$$\mathbf{x}_2 = [\mathbf{t}_2 + \mathbf{d}_2] \bmod \Lambda$$



Decoder scales by α, removes dithers, recovers modulo sum.

$$\begin{aligned}
&[\alpha\mathbf{y} - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= [\alpha(\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}) - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= [\mathbf{x}_1 + \mathbf{x}_2 - (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z} - \mathbf{d}_1 - \mathbf{d}_2] \bmod \Lambda \\
&= \Big[[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda - (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z}\Big] \bmod \Lambda \\
&= [\mathbf{v} - (1-\alpha)(\mathbf{x}_1 + \mathbf{x}_2) + \alpha\mathbf{z}] \bmod \Lambda
\end{aligned}$$

Effective Noise $\quad N_{\mathsf{EFFEC}} = (1-\alpha)^2 2P + \alpha^2 N$

Decoding the Sum of Lattice Codewords – MMSE Scaling

- Effective noise after scaling is $N_{\text{EFFEC}} = (1-\alpha)^2 2P + \alpha^2 N$.
- Minimized by setting α to be the MMSE coefficient:

$$\alpha_{\text{MMSE}} = \frac{2P}{N+2P}$$

- Plugging in, we get

$$N_{\text{EFFEC}} = \frac{2NP}{N+2P}$$

- Resulting rate is

$$R = \frac{1}{2}\log\left(\frac{P}{N_{\text{EFFEC}}}\right) = \frac{1}{2}\log\left(\frac{1}{2}+\frac{P}{N}\right)$$

- Getting the full "one plus" term is an open challenge. Does not seem possible with nested lattices.

From Messages to Lattice Points and Back

- Map messages to lattice points

$$\mathbf{t}_1 = \phi(\mathbf{w}_1) = [\mathbf{B}\gamma\mathbf{G}\mathbf{w}_1] \bmod \Lambda$$
$$\mathbf{t}_2 = \phi(\mathbf{w}_2) = [\mathbf{B}\gamma\mathbf{G}\mathbf{w}_2] \bmod \Lambda$$

- Mapping between finite field messages and lattice codewords preserves linearity:

$$\phi^{-1}\Big([\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda\Big) = \mathbf{w}_1 \oplus \mathbf{w}_2$$

- This means that after decoding a $\bmod \Lambda$ equation of lattice points we can immediately recover the finite field equation of the messages. See **Nazer-Gastpar '11** for more details.

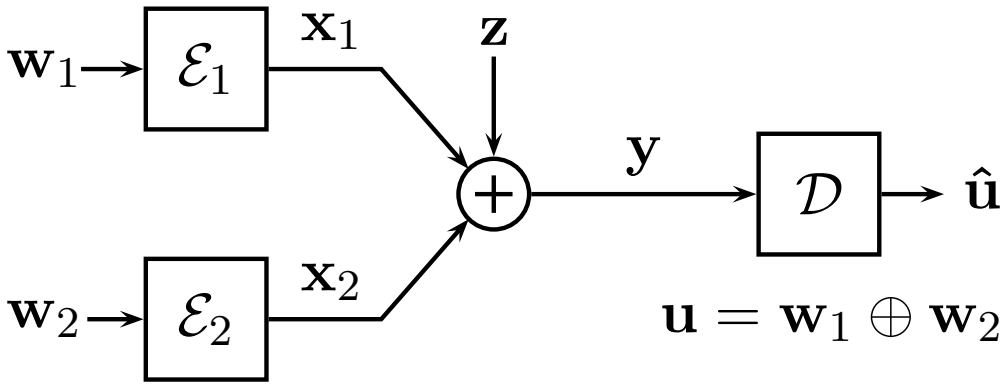
Summary: Finite Field Computation over a Gaussian MAC

Map messages to lattice points:

$$\mathbf{t}_1 = \phi(\mathbf{w}_1)$$
$$\mathbf{t}_2 = \phi(\mathbf{w}_2)$$

Transmit dithered codewords:

$$\mathbf{x}_1 = [\mathbf{t}_1 + \mathbf{d}_1] \bmod \Lambda$$
$$\mathbf{x}_2 = [\mathbf{t}_2 + \mathbf{d}_2] \bmod \Lambda$$



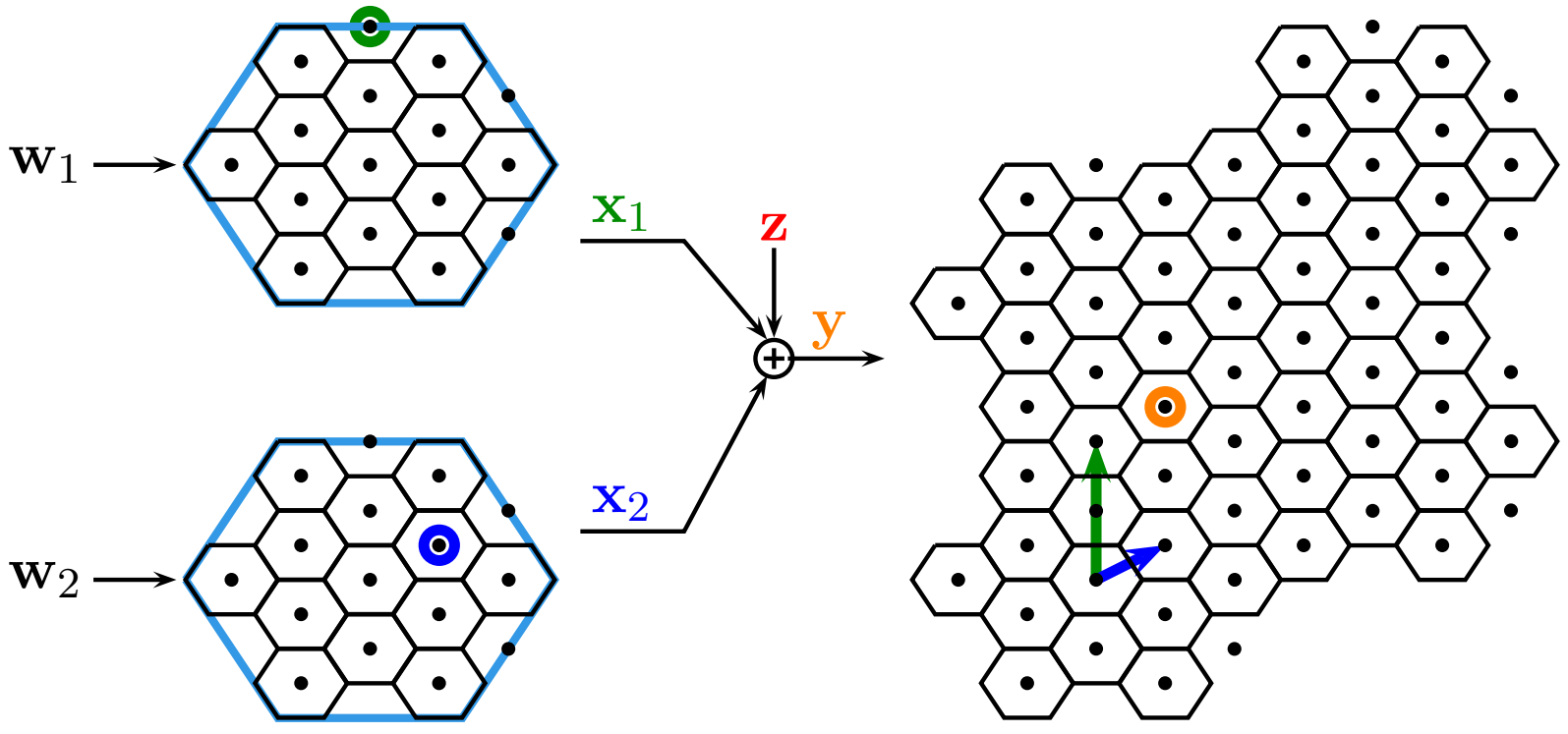
- If decoder can recover $[\mathbf{t}_1 + \mathbf{t}_2] \bmod \Lambda$, it also can get the sum of the messages

$$\mathbf{w}_1 \oplus \mathbf{w}_2 = \phi^{-1}\Big([\mathbf{t}_1 + \mathbf{t}_2] \mod \Lambda\Big) \ .$$

- Achievable rate $R = \frac{1}{2} \log \left(\frac{1}{2} + \frac{P}{N} \right)$.

Lattice Codes for Computation

All users pick the same nested lattice code: Choose messages over field $\mathbf{w}_i \in \mathbb{F}_p^k$: Map $\mathbf{w}_i$ to lattice point in Λ_{FINE} mod Λ_{COARSE}: Transmit lattice points over the channel: Decode the sum:



Decoding is successful whenever $R \leq \frac{1}{2}\log_2\left(\frac{1}{2} + \text{SNR}\right)$

Theorem

For the K-user Gaussian MAC with unit gains, a receiver can decode $\sum \mathbf{w}_i$ at rate:

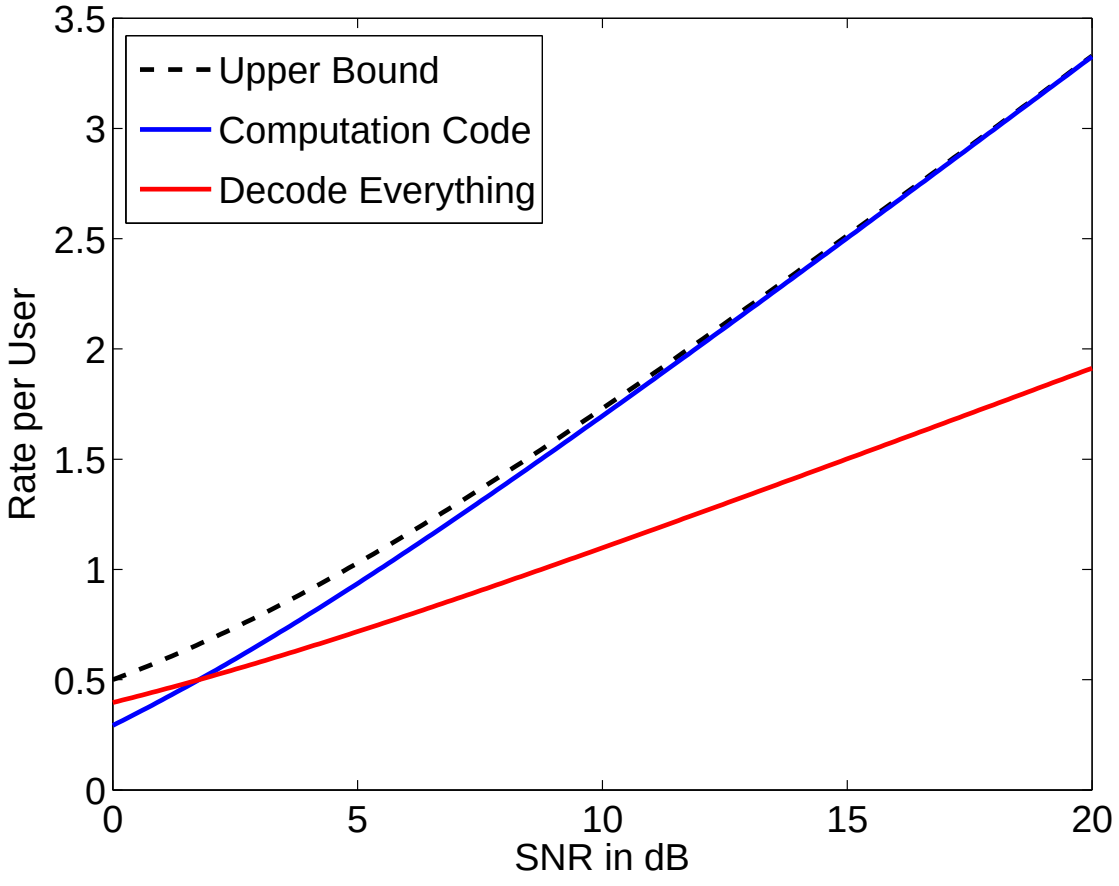
$$R = \frac{1}{2} \log \left(\frac{1}{K} + \frac{P}{N} \right)$$

Note: Constructive proof requires lattices generated from q-ary codes, where q is generally arbitrarily large.

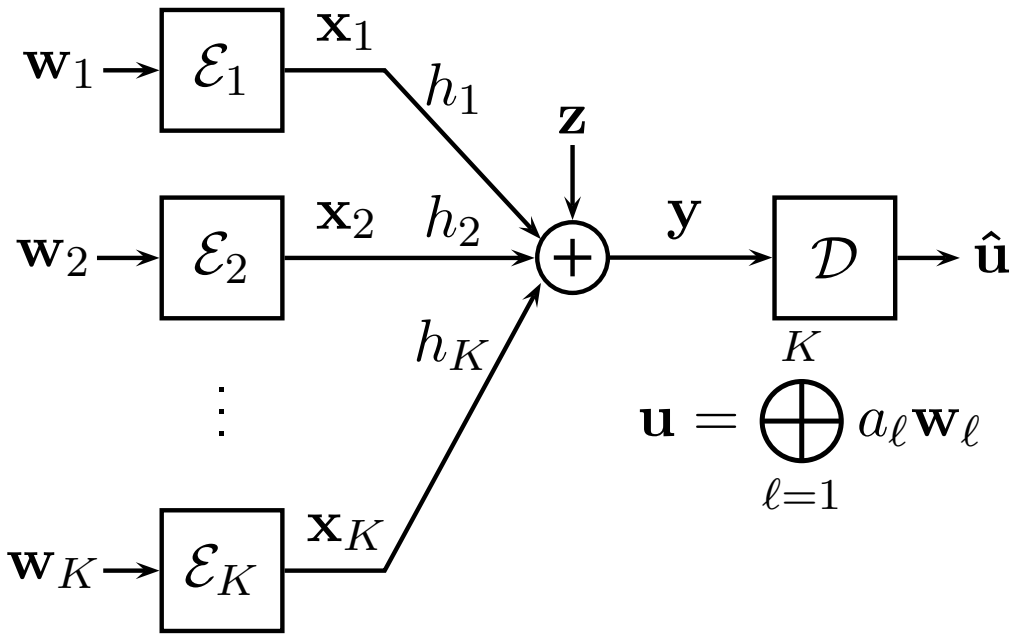
Lattice Codes for Compute-and-Forward: Direct Sum

- Want sum of messages $\sum_{\ell=1}^{K} \mathbf{w}_\ell$
- Channel is perfectly matched $\mathbf{y} = \sum_{\ell=1}^{K} \mathbf{x}_\ell + \mathbf{z}$

$$K = 2$$



Computation over Fading Channels



Computation over Fading Channels

- Map messages to lattice points $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$.
- Transmit dithered codewords $\mathbf{x}_\ell = [\mathbf{t}_\ell + \mathbf{d}_\ell] \bmod \Lambda$
- Receiver removes dithers and decodes an integer combination which can be mapped back to $\mathbb{F}_q$ to recover $\bigoplus_\ell a_\ell \mathbf{w}_\ell$.

$$\left[\mathbf{y} - \sum_{\ell=1}^{K} a_\ell \mathbf{d}_\ell\right] \bmod \Lambda$$

$$= \left[\sum_{\ell=1}^{K} h_\ell \mathbf{x}_\ell + \mathbf{z} - \sum_{\ell=1}^{K} a_\ell \mathbf{d}_\ell\right] \bmod \Lambda$$

$$= \left[\sum_{\ell=1}^{K} a_\ell (\mathbf{x}_\ell - \mathbf{d}_\ell) + \sum_{\ell=1}^{K} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\right] \bmod \Lambda$$

$$= \left[\left[\sum_{\ell=1}^{K} a_\ell \mathbf{t}_\ell\right] \bmod \Lambda + \sum_{\ell=1}^{K} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\right] \bmod \Lambda$$

Distributive Law

Effective Noise

- Effective noise due to mismatch between channel coefficients $\mathbf{h} = [h_1 \cdots h_K]^T$ and equation coefficients $\mathbf{a} = [a_1 \cdots a_K]^T$.

$$N_{\text{EFFEC}} = 1 + \text{SNR}\|\mathbf{h} - \mathbf{a}\|^2$$

$$R = \frac{1}{2} \log \left(\frac{\text{SNR}}{1 + \text{SNR}\|\mathbf{h} - \mathbf{a}\|^2} \right)$$

- Can do better with MMSE scaling.

$$\alpha \mathbf{y} = \sum_{\ell=1}^{K} a_\ell \mathbf{x}_\ell + \sum_{\ell=1}^{K} (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha \mathbf{z}$$

$$R = \max_{\alpha} \frac{1}{2} \log \left(\frac{\text{SNR}}{\alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \mathbf{a}\|^2} \right)$$

$$= \frac{1}{2} \log \left(\frac{1 + \text{SNR}\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + \text{SNR}(\|\mathbf{h}\|^2\|\mathbf{a}\|^2 - (\mathbf{h}^T\mathbf{a})^2)} \right)$$

- Practical codes and constellations: **Feng-Silva-Kschischang '10, Hern and Narayanan '11, Ordentlich and Erez '10, Osmane and Belfiore '11**

Theorem (Nazer-Gastpar 2009, IT Trans 2011)

For the Gaussian MAC with coefficients $\mathbf{h} = [h_1 \cdots h_K]^T$, unknown *to the transmitters, it is possible to decode the finite-field sum of the messages with coefficients* $\mathbf{a} = [a_1 \cdots a_K]^T$ *at rate*

$$\begin{aligned} R &= \max_{\alpha} \frac{1}{2} \log \left(\frac{\mathsf{SNR}}{\alpha^2 + \mathsf{SNR}\|\alpha \mathbf{h} - \mathbf{a}\|^2} \right) \\ &= \frac{1}{2} \log \left(\frac{1 + \mathsf{SNR}\|\mathbf{h}\|^2}{\|\mathbf{a}\|^2 + \mathsf{SNR}(\|\mathbf{h}\|^2 \|\mathbf{a}\|^2 - (\mathbf{h}^T \mathbf{a})^2)} \right) \end{aligned}$$

- What is the best equation $\mathbf{a}$ to decode?
- We can express the Compute-and-Forward Theorem equivalently as

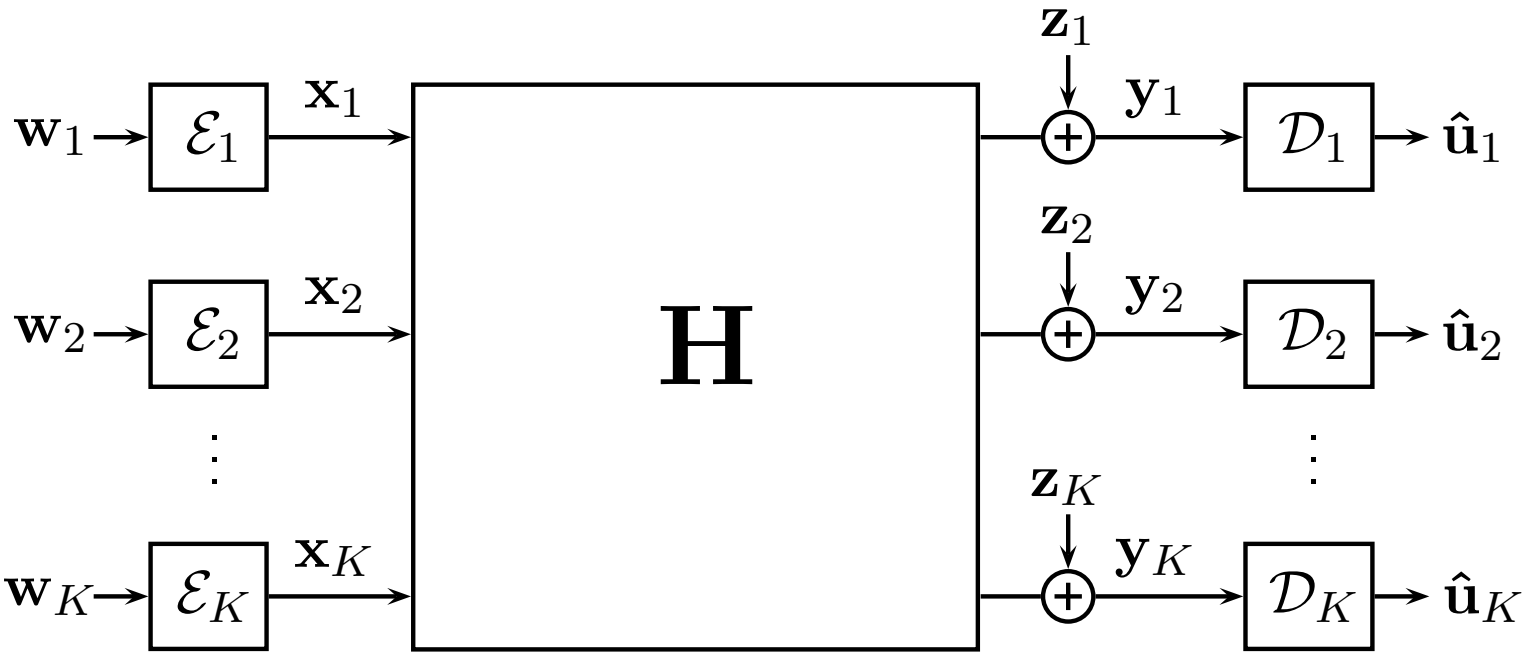
$$R = \frac{1}{2}\log\left(\mathbf{a}^T\left(I_K - \frac{\mathsf{SNR}}{1+\mathsf{SNR}\|\mathbf{h}\|^2}\mathbf{h}\mathbf{h}^T\right)\mathbf{a}\right)^{-1}$$

- Hence, finding the best $\mathbf{a}$ is a shortest lattice vector problem for the lattice with Gram matrix

$$(1+\mathsf{SNR}\|\mathbf{h}\|^2)I_K - \mathsf{SNR}\mathbf{h}\mathbf{h}^T.$$

- There is an elegant numerically efficient solution. Sahraei and Gastpar, arXiv:1404.0564 (cs.DS), April 2014.

Compute-and-Forward – Multiple Receivers



- No channel state information (CSI) at transmitters.
- Receivers use CSI to select coefficients, decode linear equation

$$\mathbf{u}_k = \bigoplus_{\ell=1}^{K} a_{k\ell}\mathbf{w}_\ell$$

- Reliable decoding possible if

$$R < \min_{k:a_{k\ell}\neq 0} \frac{1}{2}\log\left(\frac{N + P\|\mathbf{h}_k\|^2}{N\|\mathbf{a}_k\|^2 + P(\|\mathbf{h}_k\|^2\|\mathbf{a}_k\|^2 - (\mathbf{h}_k^T\mathbf{a}_k)^2)}\right)$$

Outline

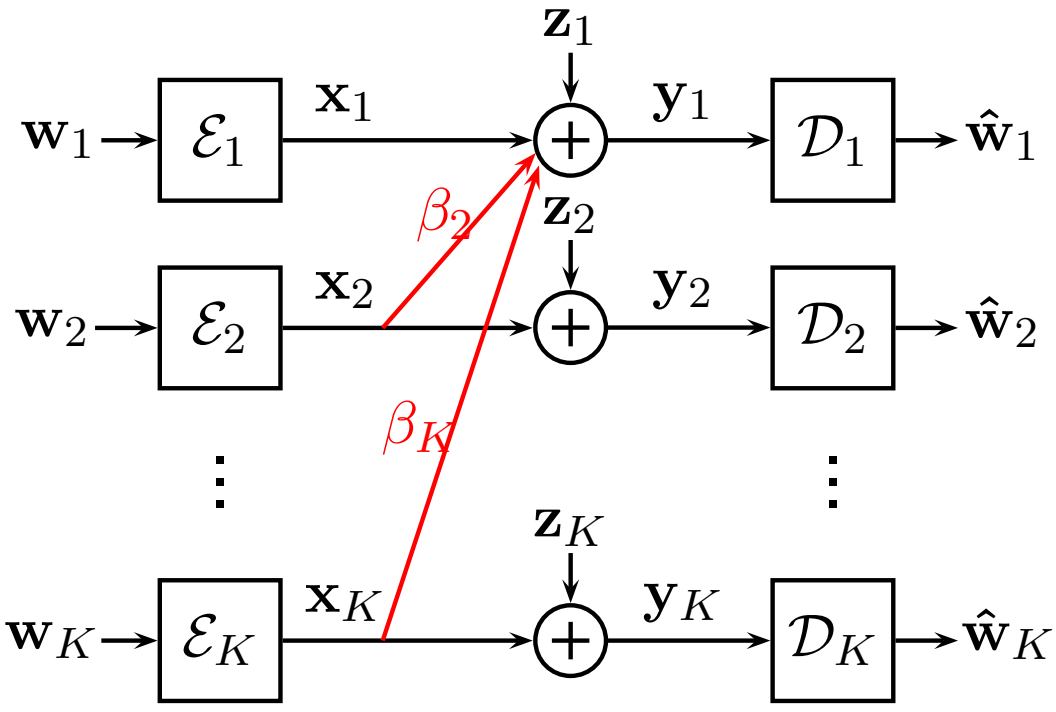
Decoding Multiple Sums of the Codewords

- We can decode one equation...
- ... but what we can do once, we can do again.
  Hence, let us consider decoding multiple equations.

Many-to-One Interference Channel

- Only receiver 1 sees interference:

$$\mathbf{y}_1 = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1$$



- "Compute-and-Forward" Approach: Encoders use the same nested lattice codebook.
- Decoder $\mathcal{D}_1$ decodes linear equations of the messages.
- Additional twist: After decoding an equation, we can (partially) remove it from the received signal.

- Only receiver 1 sees interference:

$$\mathbf{y}_1 = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1$$

  Let us denote $\mathbf{b} = (1, \beta_2, \ldots, \beta_K)$.

- It first decodes the equation

$$q_1^{(1)} \mathbf{w}_1 + q_2^{(1)} \mathbf{w}_2 + \ldots + q_K^{(1)} \mathbf{w}_K$$

  where we collect the integer coefficients into the vector $\mathbf{q}^{(1)}$.

- As we have seen, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha \mathbf{b} - \mathbf{q}^{(1)}\|^2 P + \alpha^2 N} \right)$$

- Next, we form

$$\mathbf{y}_1^{(2)} = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1 - \alpha_1' \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(1)} \mathbf{x}_\ell \right)$$

- From this, we decode

$$q_1^{(2)} \mathbf{w}_1 + q_2^{(2)} \mathbf{w}_2 + \ldots + q_K^{(2)} \mathbf{w}_K$$

  where we collect the integer coefficients into the vector $\mathbf{q}^{(2)}$.

- Again, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha_1', \alpha_2} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_2(\mathbf{b} - \alpha_1' \mathbf{q}^{(1)}) - \mathbf{q}^{(2)}\|^2 P + \alpha_2^2 N} \right)$$

  which we prefer to trivially rewrite as

$$R \leq \max_{\alpha_1, \alpha_2} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_2 \mathbf{b} - \alpha_1 \mathbf{q}^{(1)} - \mathbf{q}^{(2)}\|^2 P + \alpha_2^2 N} \right)$$

- Next, we form

$$\mathbf{y}_1^{(3)} = \mathbf{x}_1 + \sum_{\ell=2}^{K} \beta_\ell \mathbf{x}_\ell + \mathbf{z}_1 - \alpha_1' \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(1)} \mathbf{x}_\ell \right) - \alpha_2' \left(\sum_{\ell=1}^{K} \mathbf{q}_\ell^{(2)} \mathbf{x}_\ell \right)$$

- From this, we decode

$$q_1^{(3)} \mathbf{w}_1 + q_2^{(3)} \mathbf{w}_2 + \ldots + q_K^{(3)} \mathbf{w}_K$$

  where we collect the integer coefficients into the vector $\mathbf{q}^{(3)}$.

- Again, this works if the rate R is chosen to satisfy

$$R \leq \max_{\alpha_1', \alpha_2', \alpha_3} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_3(\mathbf{b} - \alpha_2' \mathbf{q}^{(2)} - \alpha_1' \mathbf{q}^{(1)}) - \mathbf{q}^{(3)}\|^2 P + \alpha_3^2 N} \right)$$

  which we prefer to trivially rewrite as

$$R \leq \max_{\alpha_1, \alpha_2, \alpha_3} \frac{1}{2} \log^+ \left(\frac{P}{\|\alpha_3 \mathbf{b} - \alpha_2 \mathbf{q}^{(2)} - \alpha_1 \mathbf{q}^{(1)} - \mathbf{q}^{(3)}\|^2 P + \alpha_3^2 N} \right)$$

Many-to-One Interference Channel

- At this point, we have decoded three equations, with coeffcients $\mathbf{q}^{(1)}, \mathbf{q}^{(2)}$, and $\mathbf{q}^{(3)}$, respectively.
- Of course, this is only useful if we can now use these to recover the message $\mathbf{w}_1$.
- Suppose we have

$$\begin{aligned} \mathbf{q}^{(1)} &= (1, 1, 2, -3) \\ \mathbf{q}^{(2)} &= (1, -5, 1, 6) \\ \mathbf{q}^{(3)} &= (-1, 3, -5, 0) \end{aligned}$$

  Can we recover $\mathbf{w}_1$?
- Construct the $3 \times K$ matrix

$$\mathbf{Q} = \begin{pmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \mathbf{q}^{(3)} \end{pmatrix}$$

  Let us denote the set of those matrices for which one can recover the first component (i.e., $\mathbf{w}_1$) by $\mathcal{Q}_1$.

- Construct the $3 \times K$ matrix

$$\mathbf{Q} \;=\; \begin{pmatrix} \mathbf{q}^{(1)} \\ \mathbf{q}^{(2)} \\ \mathbf{q}^{(3)} \end{pmatrix}$$

Definition

Let $\mathcal{Q}_1$ be the set of those matrices for which one can recover the first component (i.e., $\mathbf{w}_1$).

- **Exercise:** Give an explicit characterization of $\mathcal{Q}_1$. (For the $3 \times K$ case, and then for the general $L \times K$ case.)
- *Hint:* Consider the matrix $\mathbf{Q}'$, obtained from $\mathbf{Q}$ by removing the first column.

Theorem (Zhu-Gastpar, ISIT'13 and arXiv:1404.0273 [cs.IT])

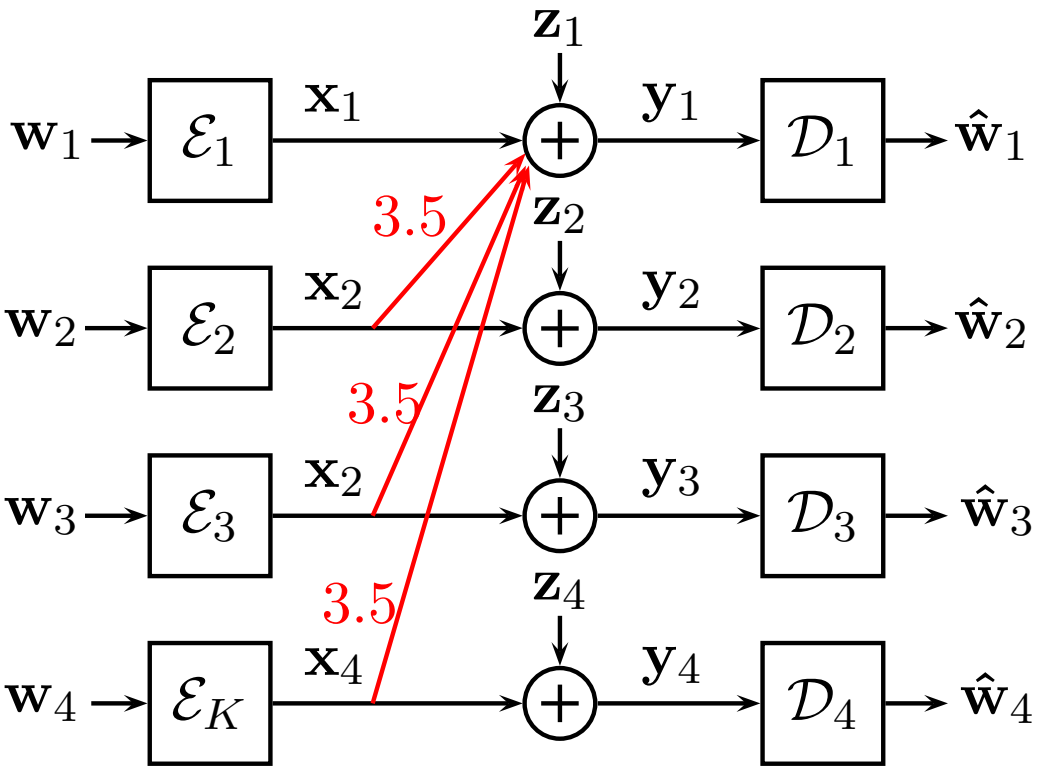
The following rates are achievable for the many-to-one interference networks

$$\begin{aligned} R_1 &\leq \max_{\substack{L\in[1:K]\\ \mathbf{Q}\in\mathcal{Q}_1}} \min_{l\in[1:L]} R_{comp}(\mathbf{q}^{(\ell)},\mathbf{q}^{(\ell-1)},\ldots,\mathbf{q}^{(1)}) \\ R_k &\leq \min\left\{\frac{1}{2}\log\left(1+P\right), R_1\right\} \text{ for } k\in[2:K] \end{aligned}$$

where

$$\begin{aligned} &R_{comp}(\mathbf{q}^{(\ell)},\mathbf{q}^{(\ell-1)},\ldots,\mathbf{q}^{(1)}) \\ &= \max_{\alpha_1,\ldots,\alpha_\ell} \frac{1}{2}\log^+\left(\frac{P}{\left\|\alpha_\ell\mathbf{b}-\sum_{j=1}^{\ell-1}\alpha_j\mathbf{q}^{(j)}-\mathbf{q}^{(\ell)}\right\|^2 P+\alpha_\ell^2 N}\right). \end{aligned}$$

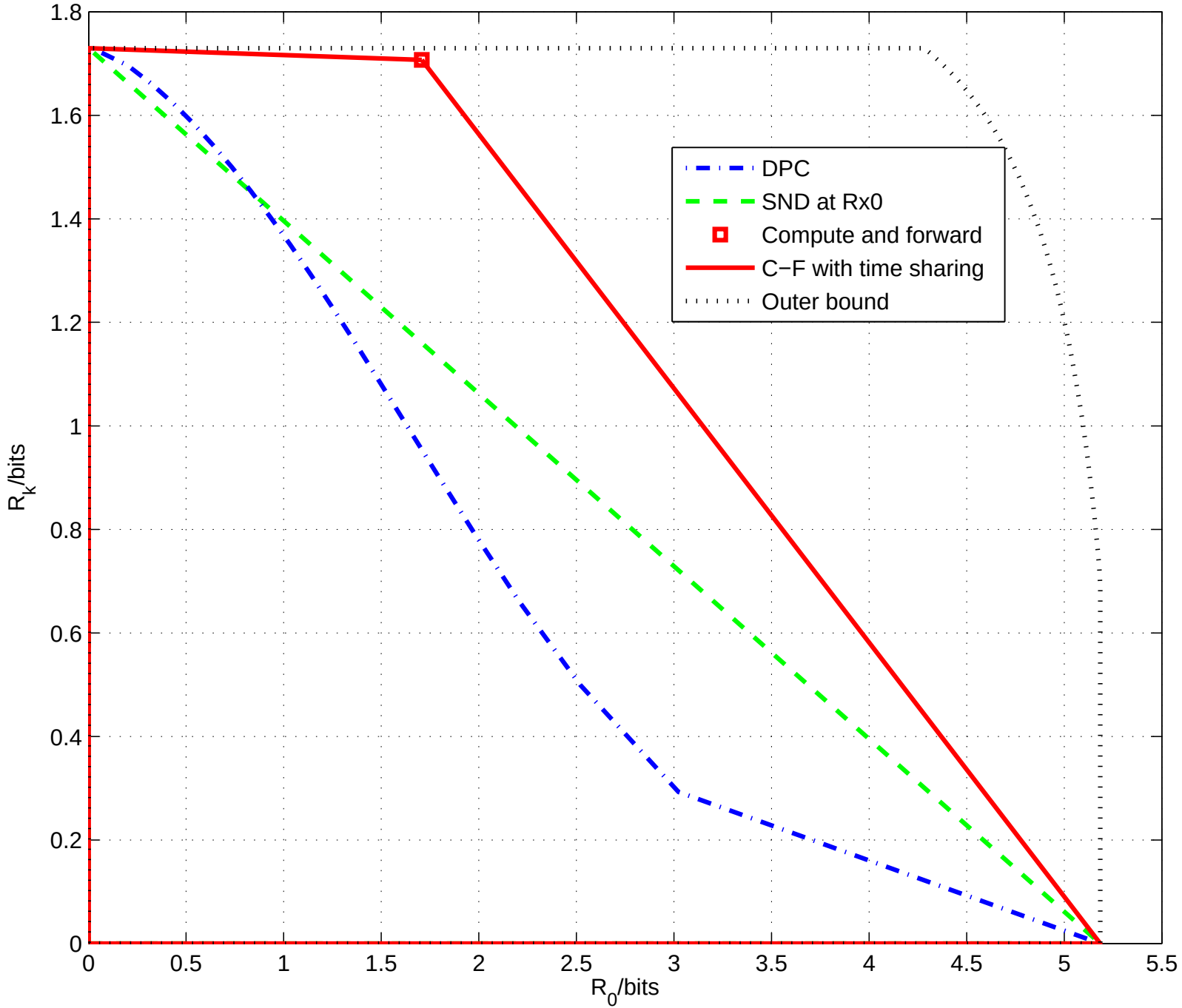
Many-to-One Interference Channel



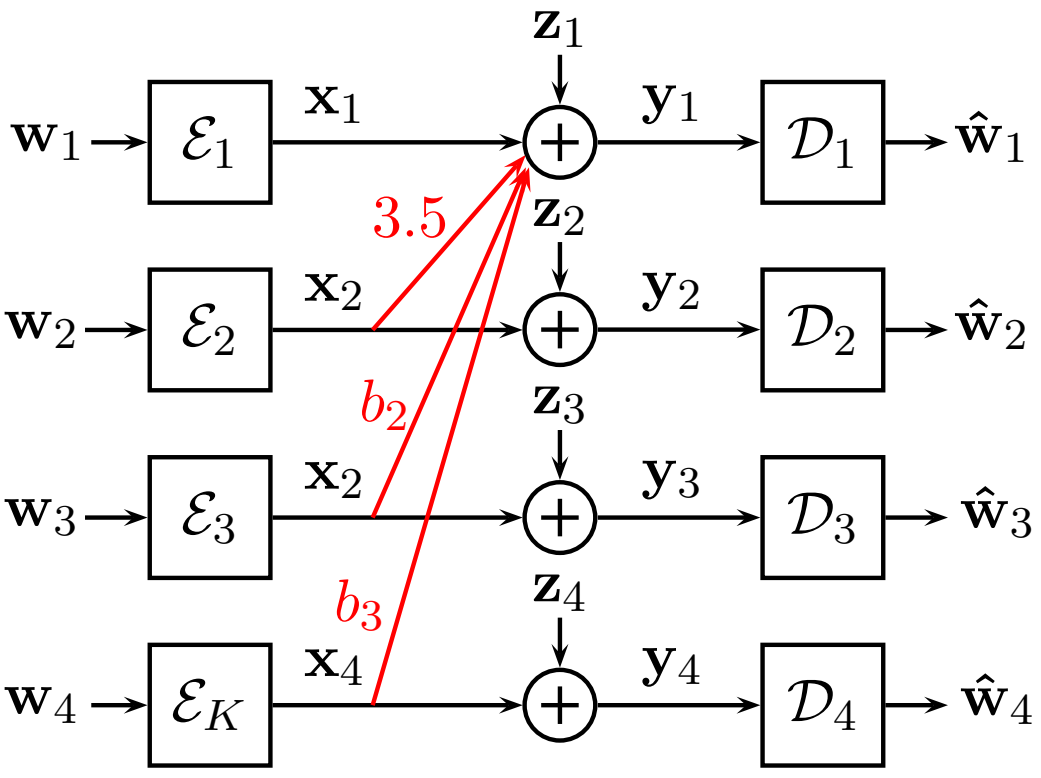
- Now, let $P = 10$.
- Then, the best equations turn out to be (in this order!)

$$\mathbf{q}^{(1)} = (1, 3, 3, 3), \quad \text{leading to} \quad R_{comp} = 1.707$$
$$\mathbf{q}^{(2)} = (3, 10, 10, 10), \quad \text{leading to} \quad R_{comp} = 1.782.$$

Many-to-One Interference Channel

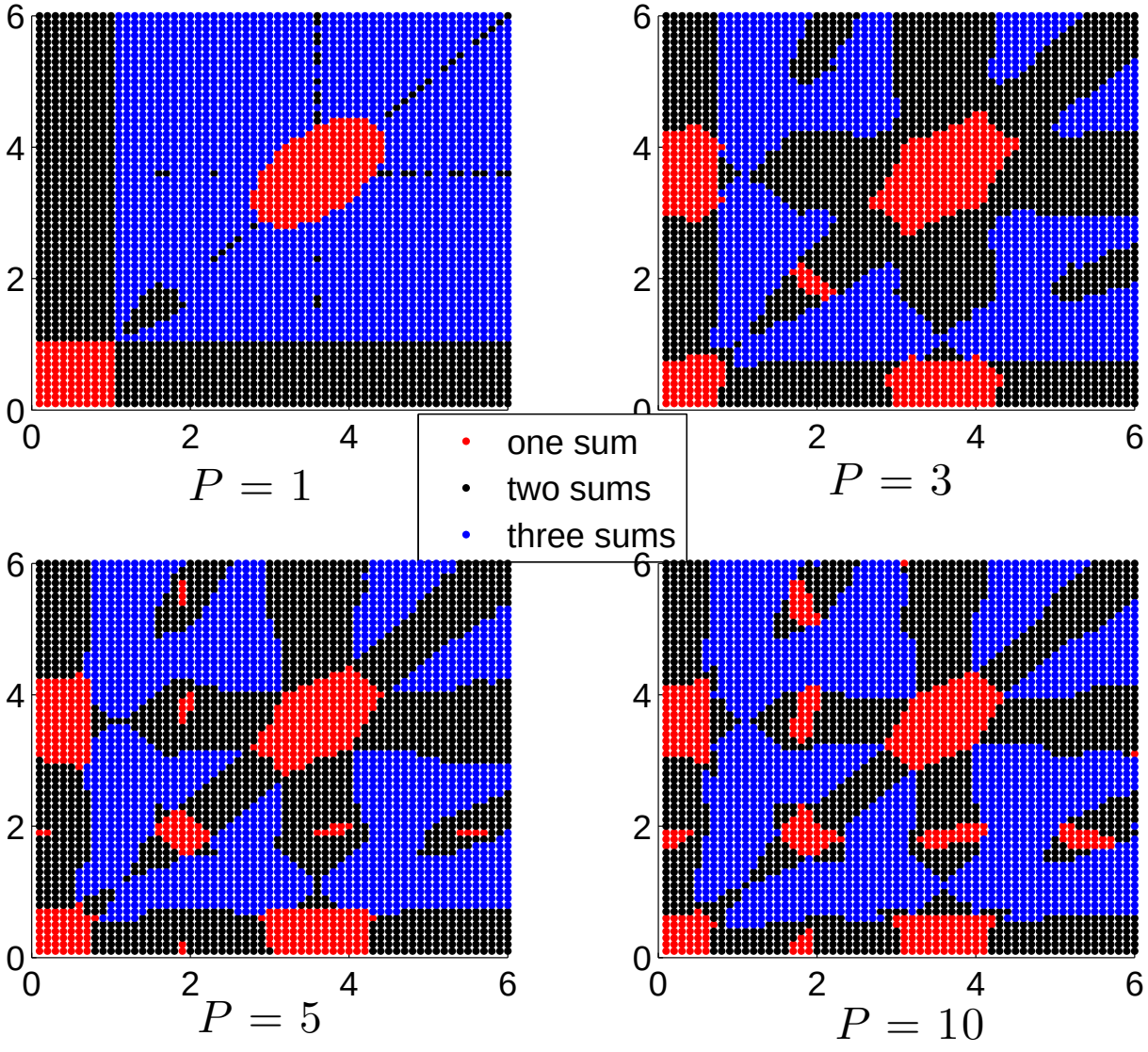


Many-to-One Interference Channel



- How many equations should we decode?
- *Example:*
  - We vary the values of b_2 and b_3.
  - We numerically optimize to maximize the **sum rate.**
  - This gives the following pictures (for various SNRs):

Many-to-One Interference Channel



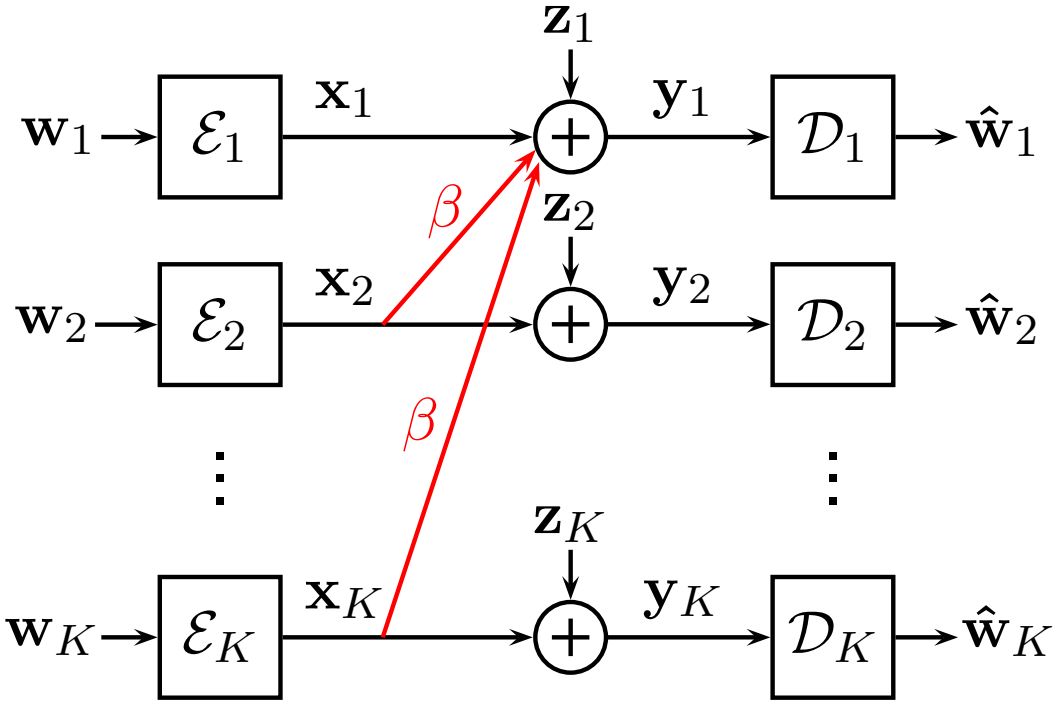
Another Example:

- Equal rates R.
- Good equations:

$$\mathbf{q}^{(1)} = (0, 1, 1, \ldots, 1),$$

$$\mathbf{q}^{(2)} = (1, 0, 0, \ldots, 0).$$

From the theorem, we find...



Many-to-One Interference Channel – Symmetric Very Strong Case

- How big does β have to be to achieve $R = \frac{1}{2}\log\left(1+\frac{P}{N}\right)$? (i.e. "very strong" case)
- Baseline scheme: Decode $\mathbf{w}_2, \ldots, \mathbf{w}_K$ at receiver 1 and remove prior to decoding $\mathbf{w}_1$.

$$R \leq \frac{1}{2(K-1)} \log\left(1 + \frac{\beta^2(K-1)P}{N+P}\right)$$

Hence,

$$\beta^2 \geq \frac{\left((1+\frac{P}{N})^{K-1}-1\right)(N+P)}{(K-1)P}$$

- By contrast, for the "compute-and-forward" scheme:

$$\beta^2 \geq \frac{(P+N)^2}{PN}$$

- Originally shown in **Sridharan-Jafarian-Vishwanath-Jafar '08** using spherical shaping region. Nested lattice scheme: **Nazer-Gastpar '11**.

Compute-and-Forward: Some Concluding Remarks

- Framework for efficient modulo-computation over linearly interfering links (with "Gaussian" noise).
- Many more extensions have been studied.
  - Example: With channel state information at the transmitter, we can *scale* the transmit lattices. This alters the problem and gives rise to new results. (E.g., J. Zhu and M.G., *2014 IEEE International Symposium on Information Theory.*)

General reference:

- B. Nazer and M. Gastpar, "Compute-and-Forward: Harnessing Interference Through Structured Codes," *IEEE Transactions on Information Theory,* 57(10):6463-6486, 2011.

Here, we also specifically discussed the results of the following paper:

- J. Zhu and M. Gastpar, "Lattice Codes for Many-to-One Interference Channels With and Without Cognitive Messages," arXiv:1404.0273 (cs.IT). Also at *2013 IEEE International Symposium on Information Theory.*