Towards an Algebraic Network Information Theory

Bobak Nazer
Boston University

Charles River Information Theory Day
April 28, 2014

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pdfs.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pdfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pdfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...
- State-of-the-art elegantly captured in the recent textbook of El Gamal and Kim.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Classical Approach:

- Generate codebooks according to some i.i.d. distributions.
- Powerful generalizations including superposition coding, dirty paper coding, block Markov coding, and many more...
- Rate regions described in terms of (single-letter) information measures optimized over pdfs.
- Many important successes: multiple-access channels, (degraded) broadcast channels, Slepian-Wolf compression, network coding, and many more...
- State-of-the-art elegantly captured in the recent textbook of El Gamal and Kim.
- Codes with algebraic structure are sought after to mimic the performance of i.i.d. random codes.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- No general theory as of yet...

Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- No general theory as of yet... but we are making progress...

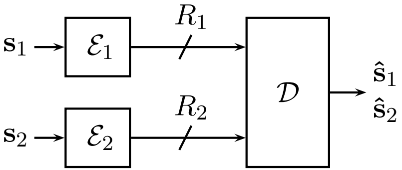
Goal: Roughly speaking, for a given network, determine necessary and sufficient conditions on the rates at which the sources (or some functions thereof) can be communicated to the destinations.

Algebraic Approach:

- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- No general theory as of yet... but we are making progress...
- Most of the initial efforts have focused on Gaussian networks.

- Algebraic Network Source Coding: Classic example of Körner and Marton.
- Algebraic Network Channel Coding: Compute-and-forward and an application to interference alignment.

Slepian-Wolf Problem



- Joint i.i.d. sources $p(\mathbf{s}_1, \mathbf{s}_2) = \prod_{i=1}^{n} p_{S_1 S_2}(s_{1i}, s_{2i})$
- **Rate Region:** Set of rates (R_1, R_2) such that the encoders can send $\mathbf{s}_1$ and $\mathbf{s}_2$ to the decoder with vanishing probability of error

$$\mathbb{P}\{(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) \neq (\mathbf{s}_1, \mathbf{s}_2)\} \to 0 \text{ as } n \to \infty$$

Random Binning

- Codebook 1: Independently and uniformly assign each source sequence $\mathbf{s}_1$ to a label $\{1, 2, \ldots, 2^{nR_1}\}$
- Codebook 2: Independently and uniformly assign each source sequence $\mathbf{s}_2$ to a label $\{1, 2, \ldots, 2^{nR_2}\}$

Random Binning

- Codebook 1: Independently and uniformly assign each source sequence $\mathbf{s}_1$ to a label $\{1, 2, \ldots, 2^{nR_1}\}$
- Codebook 2: Independently and uniformly assign each source sequence $\mathbf{s}_2$ to a label $\{1, 2, \ldots, 2^{nR_2}\}$
- Decoder: Look for jointly typical pair $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$ within the received bin. Union bound:

$$
\begin{aligned}
&\mathbb{P}\Big\{\text{jointly typical } (\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) \neq (\mathbf{s}_1, \mathbf{s}_2) \text{ in bin } (\ell_1, \ell_2)\Big\} \\
&\leq \sum_{\text{jointly typical } (\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2)} 2^{-n(R_1+R_2)} \\
&\leq 2^{n(H(S_1,S_2)+\epsilon)}\, 2^{-n(R_1+R_2)}
\end{aligned}
$$

- Codebook 1: Independently and uniformly assign each source sequence $\mathbf{s}_1$ to a label $\{1, 2, \ldots, 2^{nR_1}\}$
- Codebook 2: Independently and uniformly assign each source sequence $\mathbf{s}_2$ to a label $\{1, 2, \ldots, 2^{nR_2}\}$
- Decoder: Look for jointly typical pair $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$ within the received bin. Union bound:

$$\begin{aligned}
&\mathbb{P}\Big\{\text{jointly typical } (\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) \neq (\mathbf{s}_1, \mathbf{s}_2) \text{ in bin } (\ell_1, \ell_2)\Big\} \\
&\leq \sum_{\text{jointly typical } (\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2)} 2^{-n(R_1+R_2)} \\
&\leq 2^{n(H(S_1,S_2)+\epsilon)}\, 2^{-n(R_1+R_2)}
\end{aligned}$$

- Need $R_1 + R_2 > H(S_1, S_2)$.

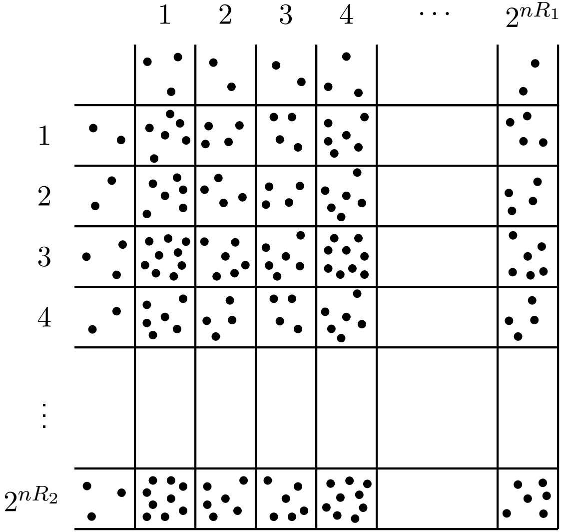
Random Binning

- Codebook 1: Independently and uniformly assign each source sequence $\mathbf{s}_1$ to a label $\{1, 2, \ldots, 2^{nR_1}\}$
- Codebook 2: Independently and uniformly assign each source sequence $\mathbf{s}_2$ to a label $\{1, 2, \ldots, 2^{nR_2}\}$
- Decoder: Look for jointly typical pair $(\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2)$ within the received bin. Union bound:

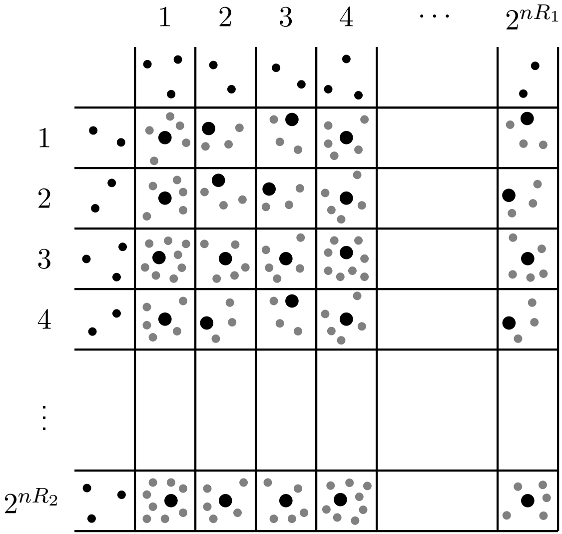
$$
\begin{aligned}
&\mathbb{P}\Big\{\text{jointly typical } (\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2) \neq (\mathbf{s}_1, \mathbf{s}_2) \text{ in bin } (\ell_1, \ell_2)\Big\} \\
&\leq \sum_{\text{jointly typical } (\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2)} 2^{-n(R_1+R_2)} \\
&\leq 2^{n(H(S_1,S_2)+\epsilon)}\, 2^{-n(R_1+R_2)}
\end{aligned}
$$

- Need $R_1 + R_2 > H(S_1, S_2)$.
- Similarly, $R_1 > H(S_1|S_2)$ and $R_2 > H(S_2|S_1)$

Slepian-Wolf Problem: Binning Illustration



Slepian-Wolf Problem: Binning Illustration



Random Linear Binning

- Assume we have chosen an injective mapping from the source alphabets to $\mathbb{F}_p$.
- Codebook 1: Generate matrix $\mathbf{G}_1$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_1$ is binned via matrix multiplication, $\mathbf{w}_1 = \mathbf{G}_1\mathbf{s}_1$.
- Codebook 2: Generate matrix $\mathbf{G}_2$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_2$ is binned via matrix multiplication, $\mathbf{w}_2 = \mathbf{G}_2\mathbf{s}_2$.

- Assume we have chosen an injective mapping from the source alphabets to $\mathbb{F}_p$.
- Codebook 1: Generate matrix $\mathbf{G}_1$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_1$ is binned via matrix multiplication, $\mathbf{w}_1 = \mathbf{G}_1\mathbf{s}_1$.
- Codebook 2: Generate matrix $\mathbf{G}_2$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_2$ is binned via matrix multiplication, $\mathbf{w}_2 = \mathbf{G}_2\mathbf{s}_2$.
- Bin assignments are uniform and pairwise independent (except for $\mathbf{s}_\ell = \mathbf{0}$)

Random Linear Binning

- Assume we have chosen an injective mapping from the source alphabets to $\mathbb{F}_p$.
- Codebook 1: Generate matrix $\mathbf{G}_1$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_1$ is binned via matrix multiplication, $\mathbf{w}_1 = \mathbf{G}_1\mathbf{s}_1$.
- Codebook 2: Generate matrix $\mathbf{G}_2$ with i.i.d. uniform entries drawn from $\mathbb{F}_p$. Each sequence $\mathbf{s}_2$ is binned via matrix multiplication, $\mathbf{w}_2 = \mathbf{G}_2\mathbf{s}_2$.
- Bin assignments are uniform and pairwise independent (except for $\mathbf{s}_\ell = \mathbf{0}$)
- Can apply the same union bound analysis as random binning.

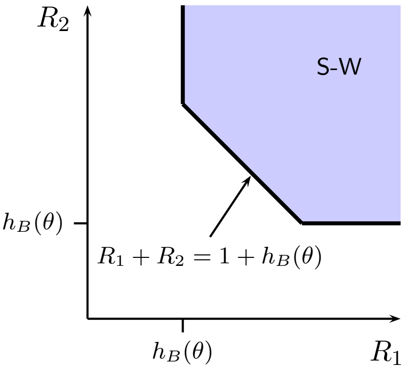
Slepian-Wolf Rate Region

Slepian-Wolf Theorem

Reliable compression possible if and only if:

$$R_1 \geq H(S_1|S_2)$$
$$R_2 \geq H(S_2|S_1)$$
$$R_1 + R_2 \geq H(S_1, S_2)$$

Random linear binning is as good as random i.i.d. binning.

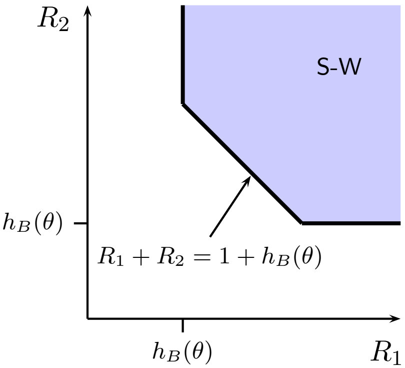


Slepian-Wolf Rate Region

Slepian-Wolf Theorem

Reliable compression possible if and only if:

$$R_1 \geq H(S_1|S_2) = h_B(\theta)$$
$$R_2 \geq H(S_2|S_1) = h_B(\theta)$$
$$R_1 + R_2 \geq H(S_1, S_2) = 1 + h_B(\theta)$$

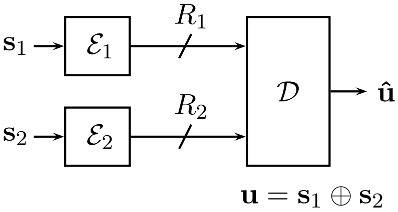


Random linear binning is as good as random i.i.d. binning.

Example: Doubly Symmetric Binary Source

$S_1 \sim \text{Bern}(1/2) \qquad U \sim \text{Bern}(\theta) \qquad S_2 = S_1 \oplus U$

- Binary sources
- $\mathbf{s}_1$ is i.i.d. Bernoulli$(1/2)$
- $\mathbf{s}_2$ is $\mathbf{s}_1$ corrupted by Bernoulli(θ) noise
- Decoder wants the modulo-2 sum .



Rate Region: Set of rates (R_1, R_2) such that there exist encoders and decoders with vanishing probability of error

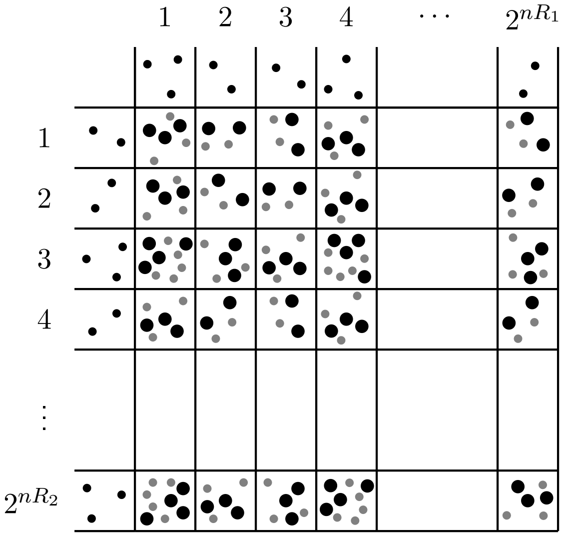
$$\mathbb{P}\{\hat{\mathbf{u}} \neq \mathbf{u}\} \to 0 \text{ as } n \to \infty$$

Are any rate savings possible over sending $\mathbf{s}_1$ and $\mathbf{s}_2$ in their entirety?

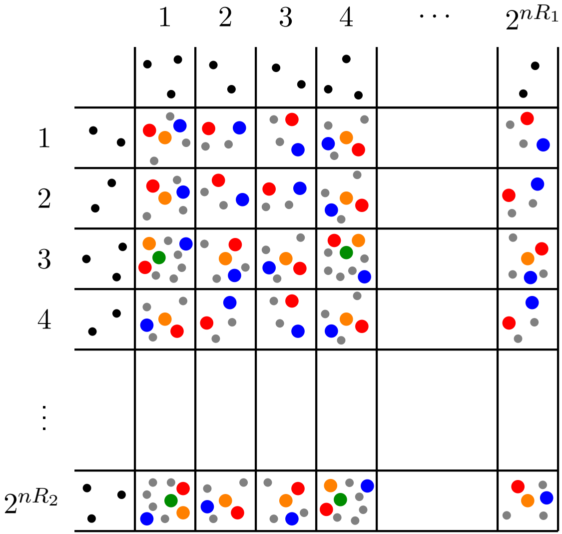
Random Binning

- Sending $\mathbf{s}_1$ and $\mathbf{s}_2$ with random binning requires $R_1 + R_2 > 1 + h_B(\theta)$.
- What happens if we use rates such that $R_1 + R_2 < 1 + h_B(\theta)$?
- There will be exponentially many pairs $(\mathbf{s}_1, \mathbf{s}_2)$ in each bin!
- This would be fine if all pairs in a bin have the same sum, $\mathbf{s}_1 + \mathbf{s}_2$. But this probability goes to zero exponentially fast!

Körner-Marton Problem: Random Binning Illustration



Körner-Marton Problem: Random Binning Illustration



Linear Binning

- Use the same random matrix $\mathbf{G}$ for linear binning at each encoder:

$$\mathbf{w}_1 = \mathbf{G}\mathbf{s}_1 \qquad \mathbf{w}_2 = \mathbf{G}\mathbf{s}_2$$

- Idea from **Körner-Marton '79:** Decoder adds up the bins.

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

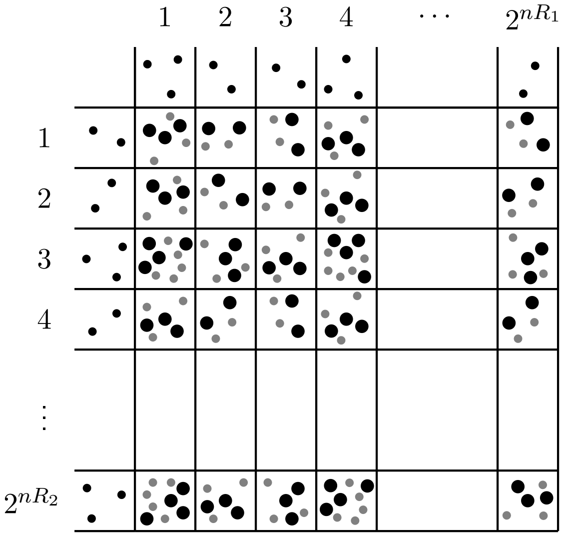
- $\mathbf{G}$ is good for compressing $\mathbf{u}$ if $R > H(U) = h_B(\theta)$.

Körner-Marton Theorem

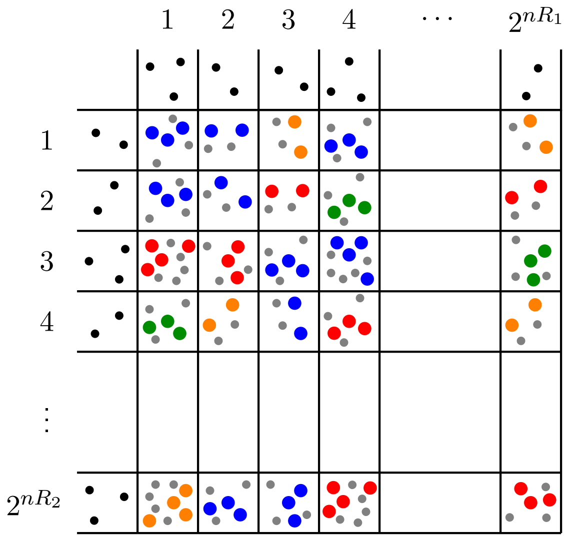
Reliable compression of the sum is possible if and only if:

$$R_1 \geq h_B(\theta) \qquad R_2 \geq h_B(\theta)\ .$$

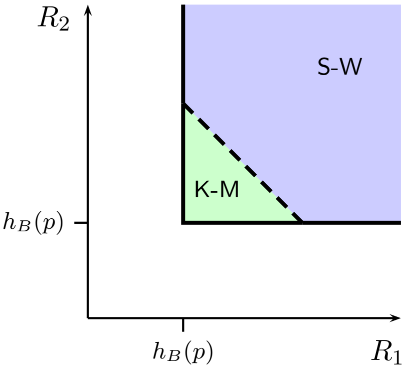
Körner-Marton Problem: Linear Binning Illustration



Körner-Marton Problem: Linear Illustration



Körner-Marton Rate Region



Linear codes can improve performance!

(for distributed computation of dependent sources)

(Algebraic) Network Source Coding

- **Krithivasan-Pradhan '09:** Nested lattice coding framework for distributed Gaussian source coding.
- **Krithivasan-Pradhan '11:** Nested group coding framework for distributed source coding for discrete memoryless sources.
- Can show that these rate regions sometimes outperform the Berger-Tung region (best known performance via i.i.d. ensembles).

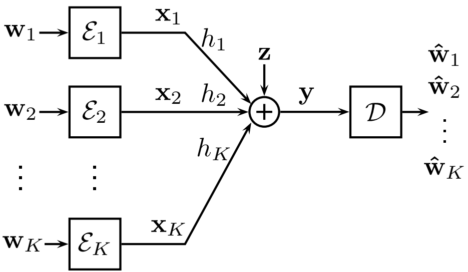
(Algebraic) Network Source Coding

- **Krithivasan-Pradhan '09:** Nested lattice coding framework for distributed Gaussian source coding.
- **Krithivasan-Pradhan '11:** Nested group coding framework for distributed source coding for discrete memoryless sources.
- Can show that these rate regions sometimes outperform the Berger-Tung region (best known performance via i.i.d. ensembles).
- Now let's take a look at an algebraic framework for network channel coding.

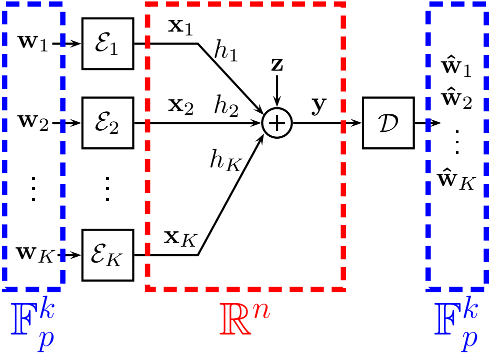
Goal: Convert noisy Gaussian networks into noiseless finite field ones.

Compute-and-Forward

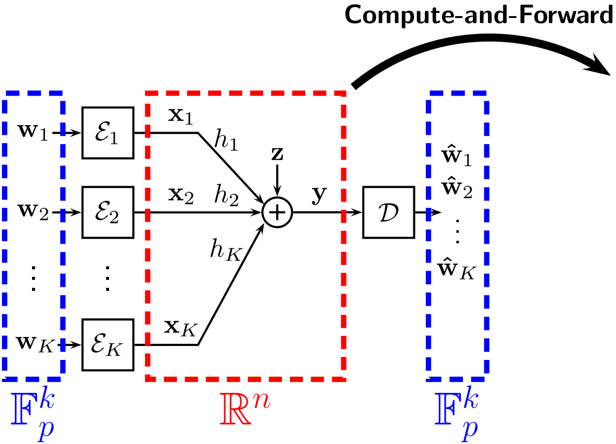
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



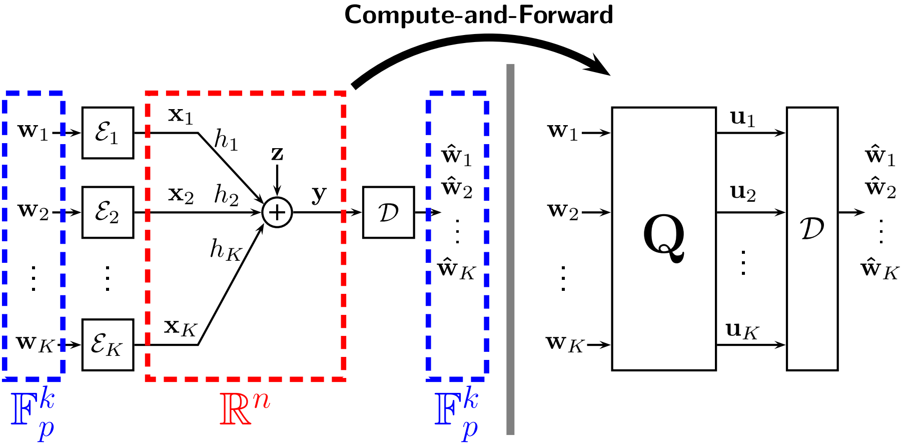
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



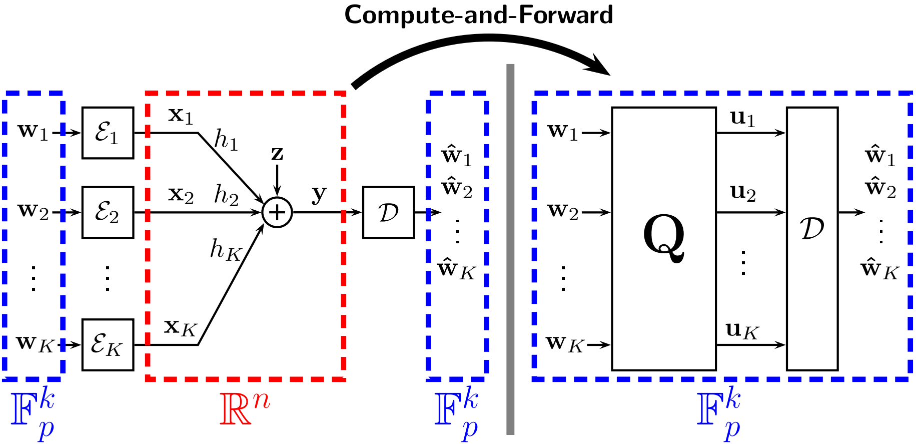
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



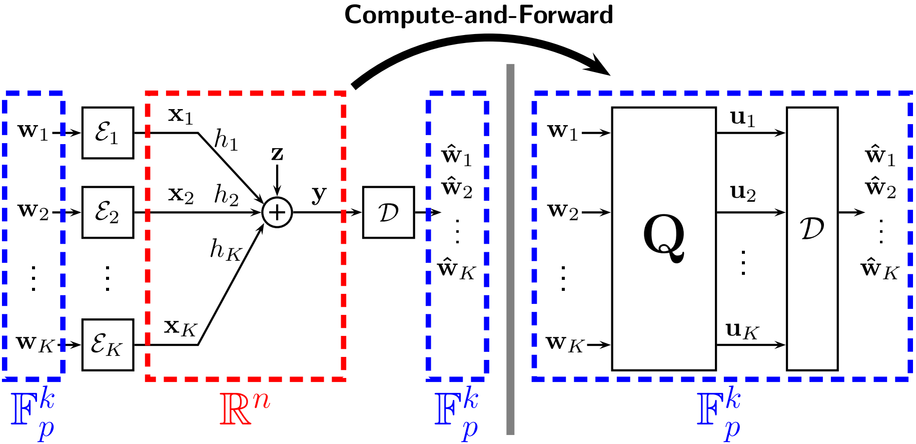
Goal: Convert noisy Gaussian networks into noiseless finite field ones.



Goal: Convert noisy Gaussian networks into noiseless finite field ones.

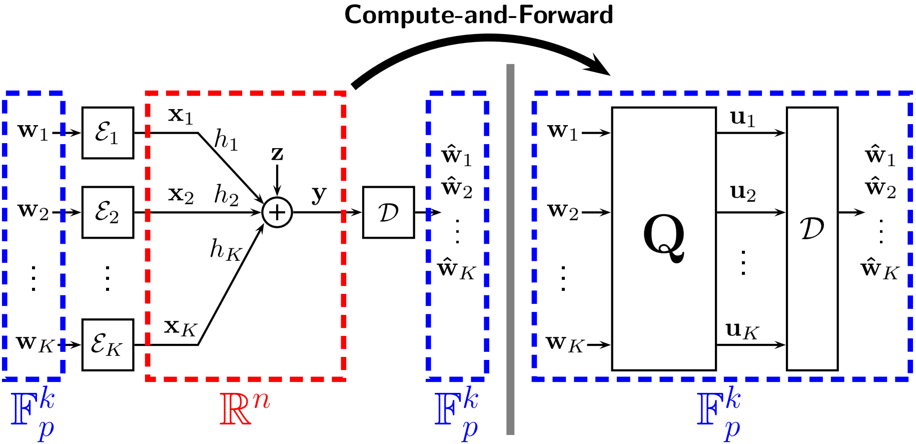


Goal: Convert noisy Gaussian networks into noiseless finite field ones.



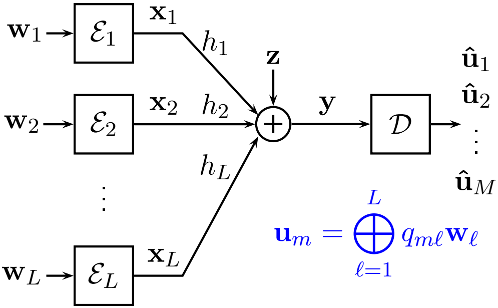
- Which linear combinations can be sent over a given channel?

Goal: Convert noisy Gaussian networks into noiseless finite field ones.



- Which linear combinations can be sent over a given channel?
- Where can this help us?

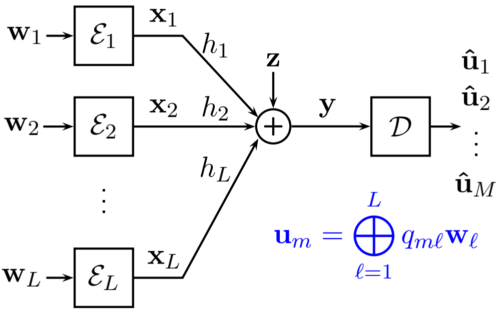
Compute-and-Forward: Problem Statement



$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell$$

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

Compute-and-Forward: Problem Statement

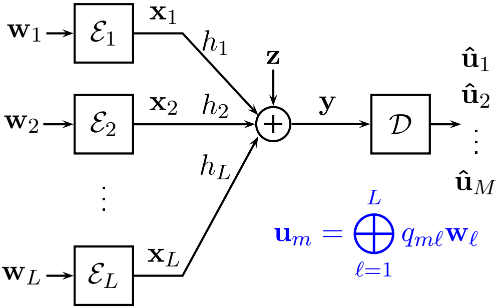


$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell$$

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim\limits_{n\to\infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\Big) = 0$.

Compute-and-Forward: Problem Statement

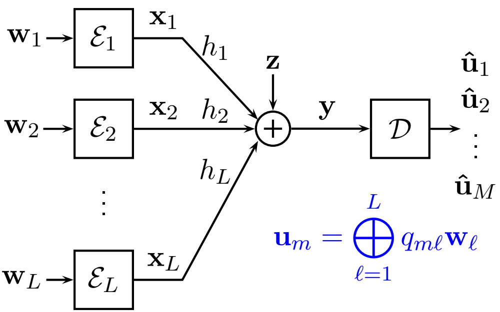


$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell$$

- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim\limits_{n\to\infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\Big) = 0$.
- Receiver can use its channel state information (CSI) to match the linear combination coefficients $q_{m\ell} \in \mathbb{F}_p$ to the channel coefficients $h_\ell \in \mathbb{R}$. Transmitters do not require CSI.

Compute-and-Forward: Problem Statement



- Messages are finite field vectors, $\mathbf{w}_\ell \in \mathbb{F}_p^k$.
- Real-valued inputs and outputs, $\mathbf{x}_\ell, \mathbf{y} \in \mathbb{R}^n$.
- Power constraint, $\frac{1}{n}\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq P$.
- Gaussian noise, $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- Equal rates: $R = \dfrac{k}{n}\log_2 p$

- Decoder wants M linear combinations of the messages with vanishing probability of error $\lim\limits_{n\to\infty} \mathbb{P}\Big(\bigcup_m \{\hat{\mathbf{u}}_m \neq \mathbf{u}_m\}\Big) = 0$.
- Receiver can use its channel state information (CSI) to match the linear combination coefficients $q_{m\ell} \in \mathbb{F}_p$ to the channel coefficients $h_\ell \in \mathbb{R}$. Transmitters do not require CSI.
- What rates are achievable as a function of h_ℓ and $q_{m\ell}$?

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1} \ a_{m2} \ \cdots \ a_{mL}]^{\mathsf{T}} \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = [a_{m\ell}] \bmod p$$

(where we assume an implicit mapping between $\mathbb{F}_p$ and $\mathbb{Z}_p$).

Computation Rate

- Want to characterize achievable rates as a function of h_ℓ and $q_{m\ell}$.
- Easier to think about integer rather than finite field coefficients.
- The linear combination with integer coefficient vector $\mathbf{a}_m = [a_{m1}\ a_{m2}\ \cdots\ a_{mL}]^\mathsf{T} \in \mathbb{Z}^L$ corresponds to

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell}\mathbf{w}_\ell \qquad \text{where } q_{m\ell} = [a_{m\ell}] \bmod p$$

  (where we assume an implicit mapping between $\mathbb{F}_p$ and $\mathbb{Z}_p$).
- **Key Definition:** The computation rate region described by $R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a})$ is *achievable* if, for any $\epsilon > 0$ and n, p large enough, a receiver can decode any linear combinations with integer coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \mathbb{Z}^L$ for which the message rate R satisfies

$$R < \min_m R_{\mathsf{comp}}(\mathbf{h}, \mathbf{a}_m)$$

Compute-and-Forward: Achievable Rates

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \max_{\alpha \in \mathbb{R}} \frac{1}{2} \log^{+} \left(\frac{P}{\alpha^2 + P\|\alpha \mathbf{h} - \mathbf{a}\|^2} \right)$$

is achievable.

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^{+} \left(\frac{P}{\mathbf{a}^{\mathsf{T}} \left(P^{-1} \mathbf{I} + \ \mathbf{h} \mathbf{h}^{\mathsf{T}} \right)^{-1} \mathbf{a}} \right)$$

is achievable.

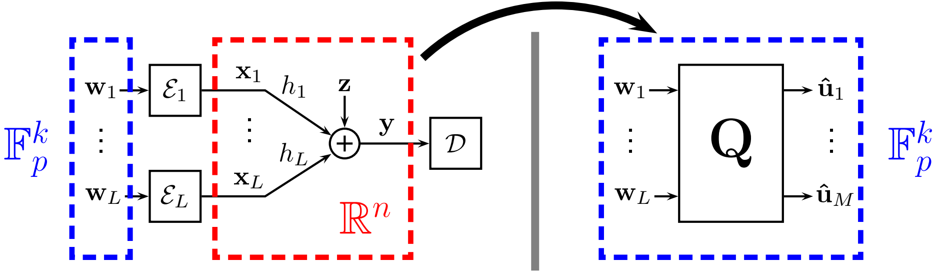
Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\mathbf{a}^\mathsf{T} \left(P^{-1} \mathbf{I} + \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \mathbf{a}} \right)$$

is achievable.

Compute-and-Forward



if $R < \min_m R_{\text{comp}}(\mathbf{h}, \mathbf{a}_m)$ for some $\mathbf{a}_m \in \mathbb{Z}^L$ satisfying $[\mathbf{a}_m] \bmod p = \mathbf{q}_m$.

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\mathbf{a}^\mathsf{T} \big(P^{-1} \mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \big)^{-1} \mathbf{a}} \right)$$

is achievable.

Special Cases:

- Perfect Match: $R_{\text{comp}}(\mathbf{a}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\mathbf{a}\|^2} + P \right)$

Theorem (Nazer-Gastpar '11)

The computation rate region described by

$$R_{comp}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{P}{\mathbf{a}^\mathsf{T} \left(P^{-1} \mathbf{I} + \ \mathbf{h}\mathbf{h}^\mathsf{T} \right)^{-1} \mathbf{a}} \right)$$

is achievable.

Special Cases:

- Perfect Match: $R_{\text{comp}}(\mathbf{a}, \mathbf{a}) = \frac{1}{2} \log^+ \left(\frac{1}{\|\mathbf{a}\|^2} + P \right)$
- Decode a Message:

$$R_{\text{comp}}\Big(\mathbf{h}, [\underbrace{0 \ \cdots \ 0}_{m-1 \text{ zeros}} \ 1 \ 0 \ \cdots \ 0]^\mathsf{T}\Big) = \frac{1}{2} \log \left(1 + \frac{h_m^2 P}{1 + P \sum_{\ell \neq m} h_\ell^2} \right)$$

$$\begin{aligned}\mathbf{y} &= \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} \\ &= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}\end{aligned}$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.

$$\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$$
$$= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}}$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.
- Independent effective noise $\Longrightarrow$ dithering.

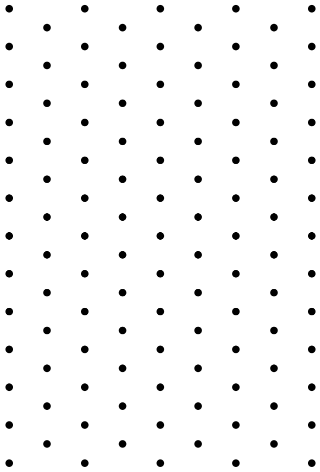
$$\begin{aligned}\mathbf{y} &= \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z} \\ &= \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell + \underbrace{\sum_{\ell=1}^{L} (h_\ell - a_\ell)\mathbf{x}_\ell + \mathbf{z}}_{\text{Effective Noise}} \xrightarrow{\text{Decode}} \bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell\end{aligned}$$

Desired Codebook:

- Closed under integer linear combinations $\Longrightarrow$ lattice codebook.
- Independent effective noise $\Longrightarrow$ dithering.
- Isomorphic to $\mathbb{F}_p^k$ $\Longrightarrow$ nested lattice codebook.

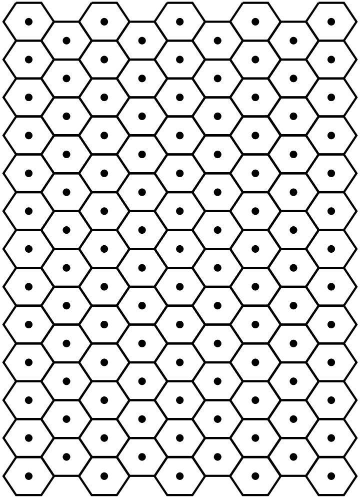
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.



Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.



- Nearest neighbor quantizer:

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

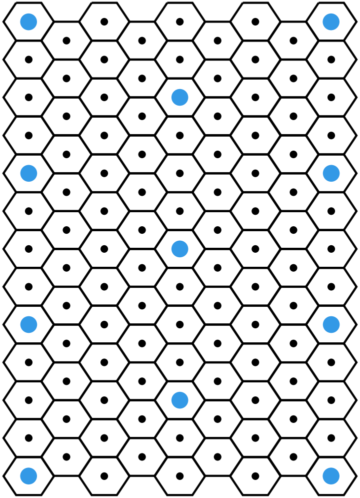
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:

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

- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$



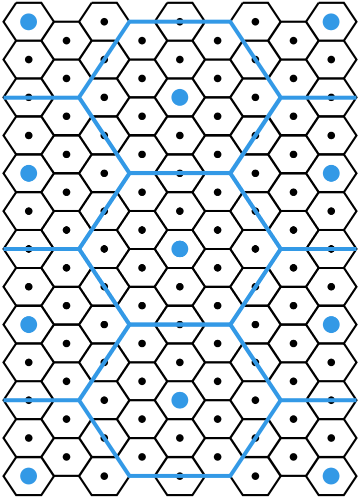
Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.

- Nearest neighbor quantizer:

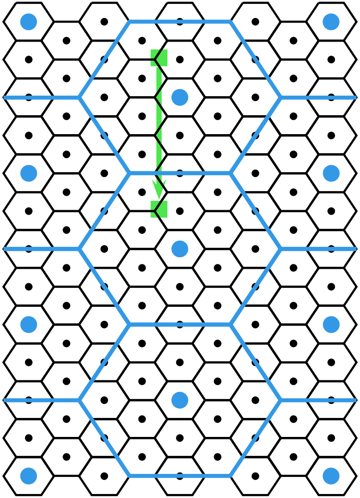
$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|_2$$

- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\mathsf{FINE}}$



Nested Lattices

- A lattice is a discrete subgroup of $\mathbb{R}^n$.



- Nearest neighbor quantizer:

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

- Two lattices Λ and Λ_{FINE} are nested if $\Lambda \subset \Lambda_{\text{FINE}}$

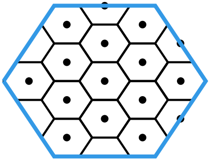
- Quantization error serves as modulo operation:

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

Distributive Law:
$$\big[\mathbf{x}_1 + a[\mathbf{x}_2] \bmod \Lambda\big] \bmod \Lambda = [\mathbf{x}_1 + a\mathbf{x}_2] \bmod \Lambda \quad \text{for all } a \in \mathbb{Z}.$$

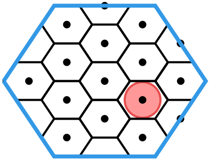
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.

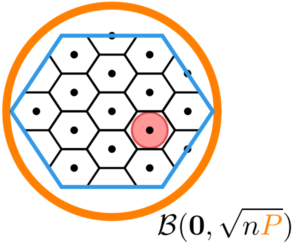


- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.

- Fine lattice Λ_{FINE} protects against noise.



- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.
- Fine lattice Λ_{FINE} protects against noise.
- Coarse lattice Λ enforces the power constraint.



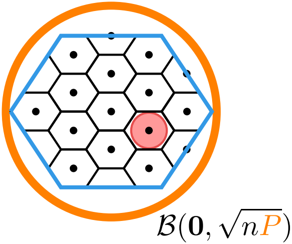
- Nested Lattice Code: Formed by taking all elements of Λ_{FINE} that lie in the fundamental Voronoi region of Λ.

- Fine lattice Λ_{FINE} protects against noise.

- Coarse lattice Λ enforces the power constraint.

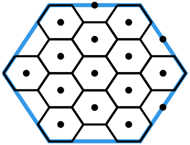
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '12.**

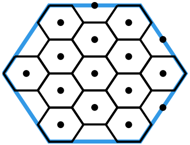
- **Erez-Zamir '04:** Nested lattice codes can achieve the point-to-point Gaussian capacity.



Compute-and-Forward: Illustration

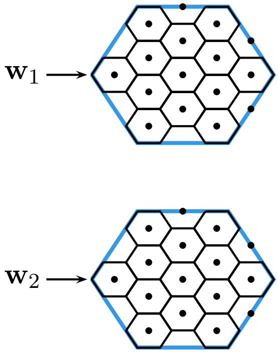
All users employ the same nested lattice code:





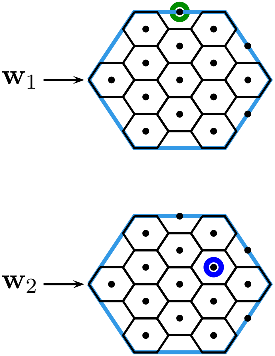
Compute-and-Forward: Illustration

Choose message vectors over finite field $\mathbf{w}_\ell \in \mathbb{F}_p^k$:

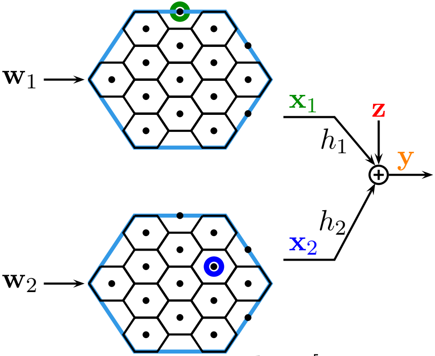


Compute-and-Forward: Illustration

Map $\mathbf{w}_\ell$ to lattice point $\mathbf{t}_\ell = \phi(\mathbf{w}_\ell)$:



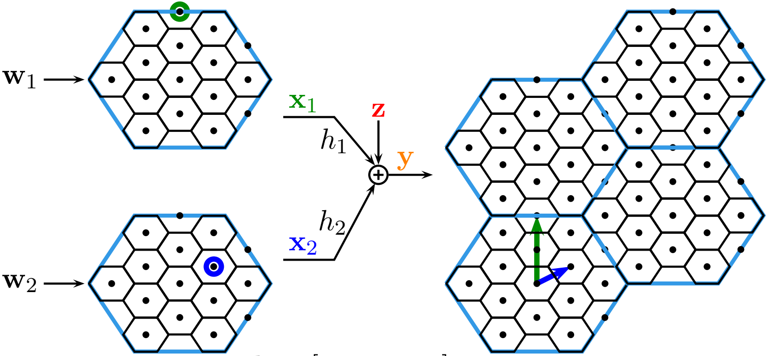
Transmit lattice points over the channel:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Transmit lattice points over the channel:

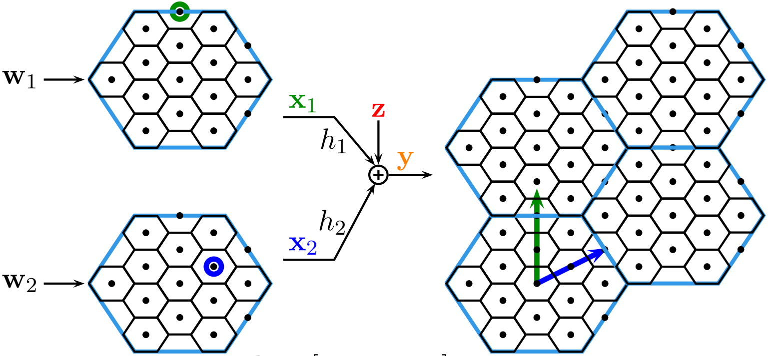


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

Lattice codewords are scaled by channel coefficients:

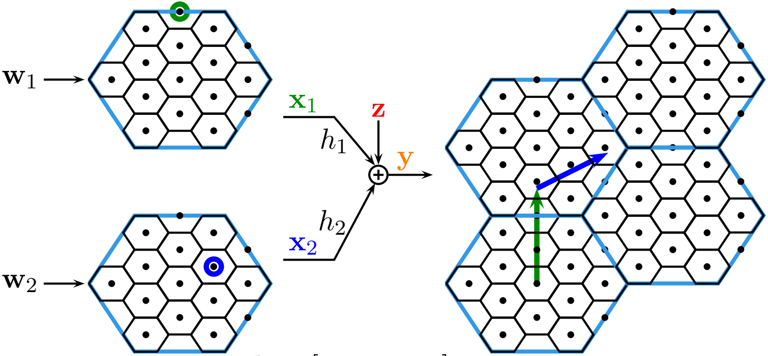


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Compute-and-Forward: Illustration

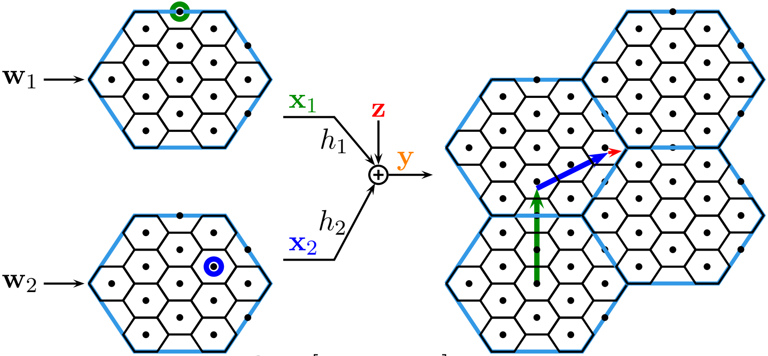
Scaled codewords added together plus noise:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

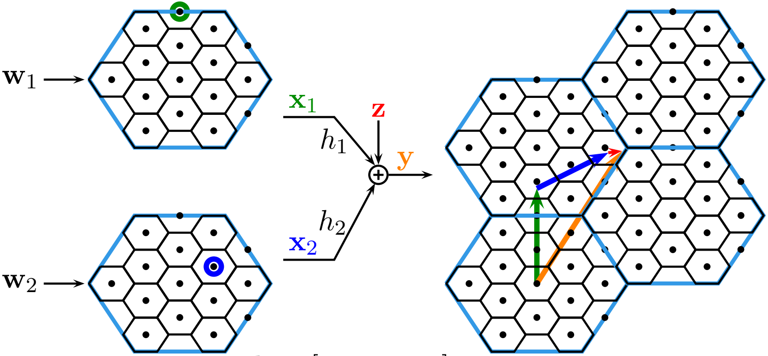
Scaled codewords added together plus noise:



$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Extra noise penalty for non-integer channel coefficients:

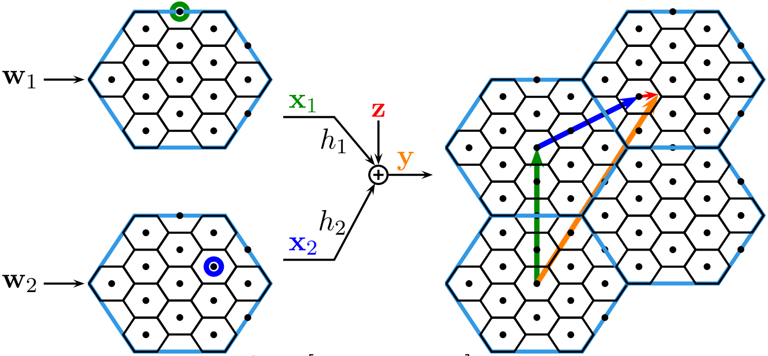


$$\mathbf{h} = [\ 1.4 \quad 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $1 + P\|\mathbf{h} - \mathbf{a}_m\|^2$

Scale output by α to reduce non-integer noise penalty:

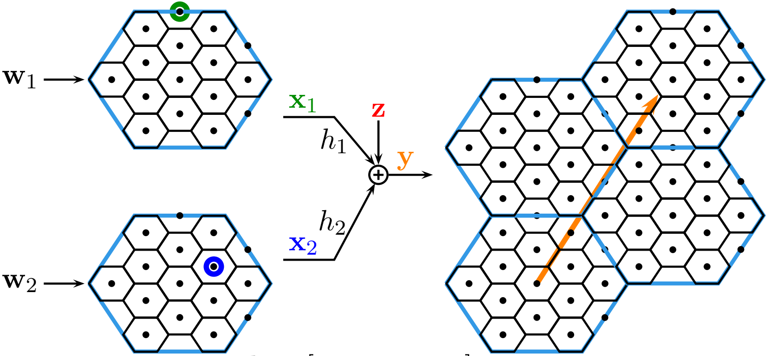


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Scale output by α to reduce non-integer noise penalty:

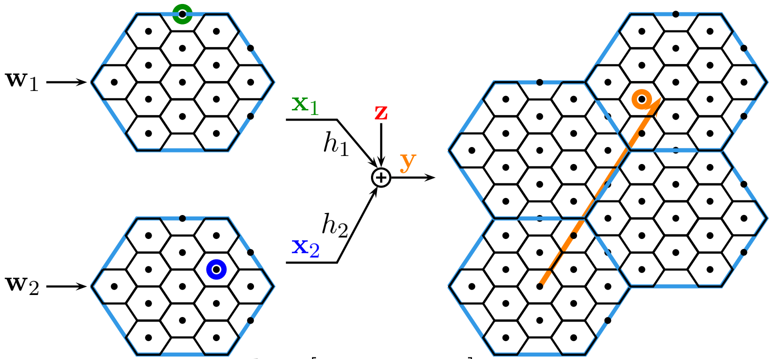


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Decode to the closest lattice point:

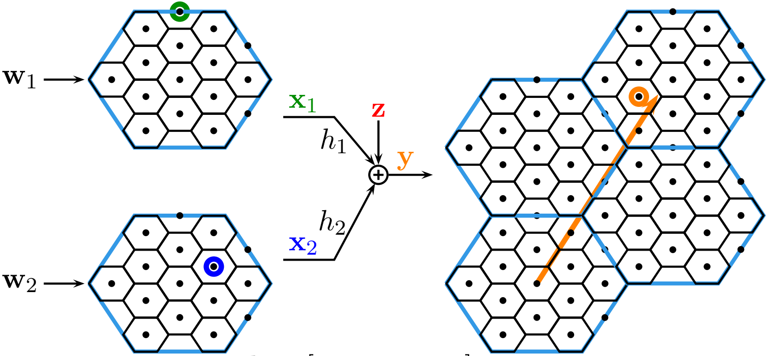


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Recover integer linear combination $\bmod \Lambda_{\mathsf{C}}$:

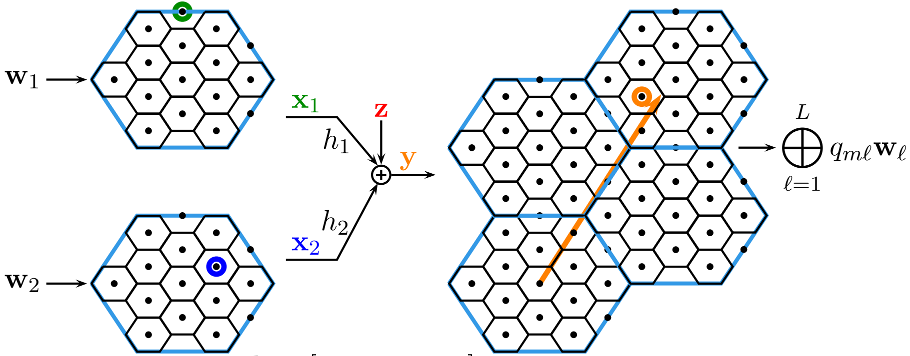


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

Map back to linear combination of the messages:

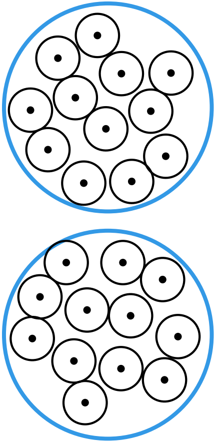


$$\alpha\mathbf{h} = [\ \alpha 1.4 \quad \alpha 2.1\ ]$$

$$\mathbf{a}_m = [\ 2 \quad 3\ ]$$

Effective noise: $\alpha^2 + P\|\alpha\mathbf{h} - \mathbf{a}_m\|^2$

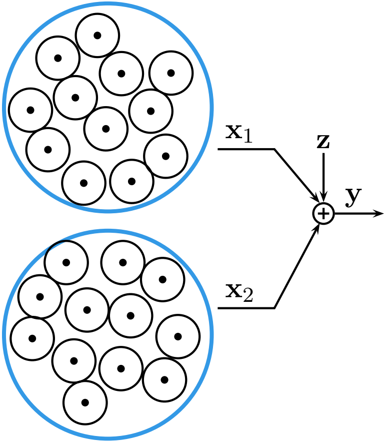
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

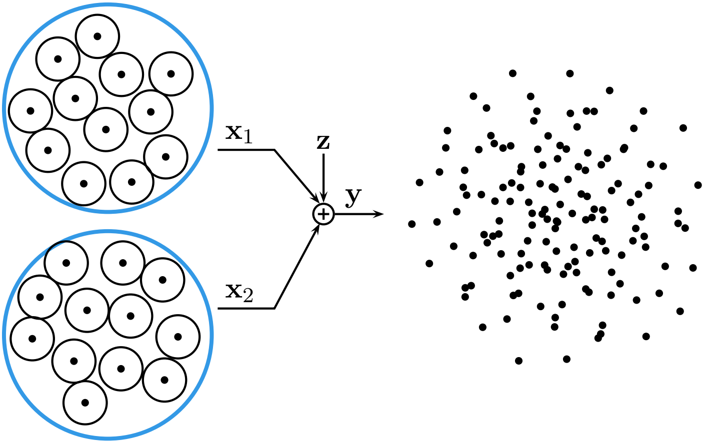
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

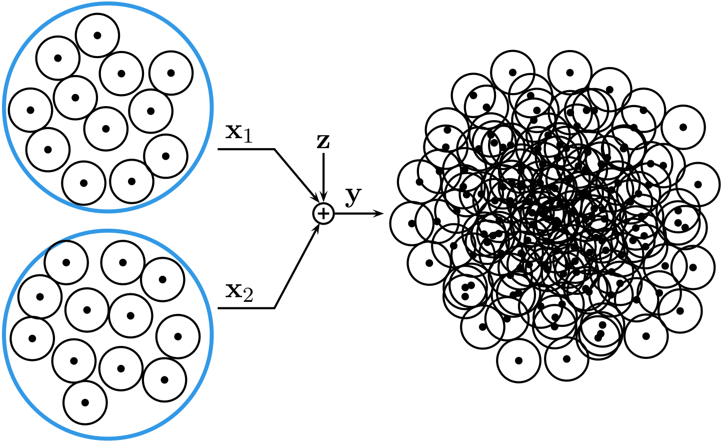
Random i.i.d. codes are not good for computation



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

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



2^{nR} codewords each.

2^{n2R} possible sums of codewords.

(Algebraic) Network Channel Coding

- Compute-and-forward is a useful setting to develop algebraic multi-user coding techniques.

(Algebraic) Network Channel Coding

- Compute-and-forward is a useful setting to develop algebraic multi-user coding techniques.

- **Ordentlich-Erez-Nazer '13:** In a K-user Gaussian multiple-access channel, the sum of the K best computation rates is exactly equal to the multiple-access sum capacity. Algebraic successive cancellation gives this operational meaning.

- Compute-and-forward is a useful setting to develop algebraic multi-user coding techniques.
- **Ordentlich-Erez-Nazer '13:** In a K-user Gaussian multiple-access channel, the sum of the K best computation rates is exactly equal to the multiple-access sum capacity. Algebraic successive cancellation gives this operational meaning.
- Upcoming work on a compute-and-forward framework for discrete memoryless networks.

- Compute-and-forward is a useful setting to develop algebraic multi-user coding techniques.
- **Ordentlich-Erez-Nazer '13:** In a K-user Gaussian multiple-access channel, the sum of the K best computation rates is exactly equal to the multiple-access sum capacity. Algebraic successive cancellation gives this operational meaning.
- Upcoming work on a compute-and-forward framework for discrete memoryless networks.
- Let's take a look at an application of compute-and-forward to interference alignment.

Interference-Free Capacity

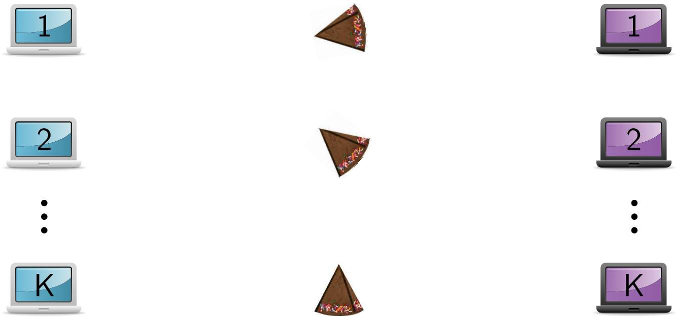
Interference-Free Capacity

Time Division

Time Division

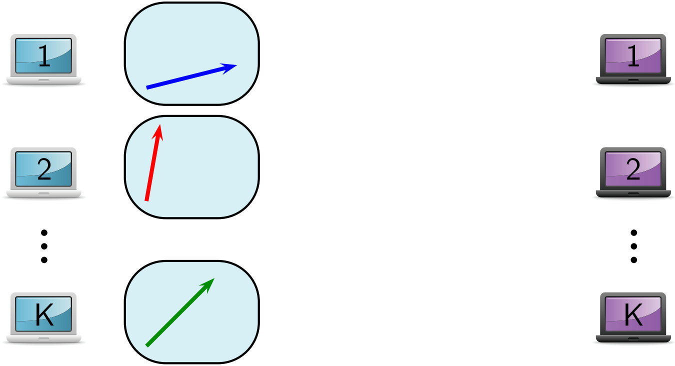
Time Division

Time Division



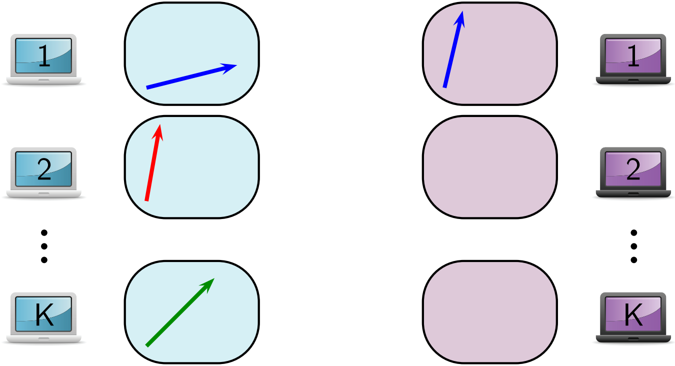
- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

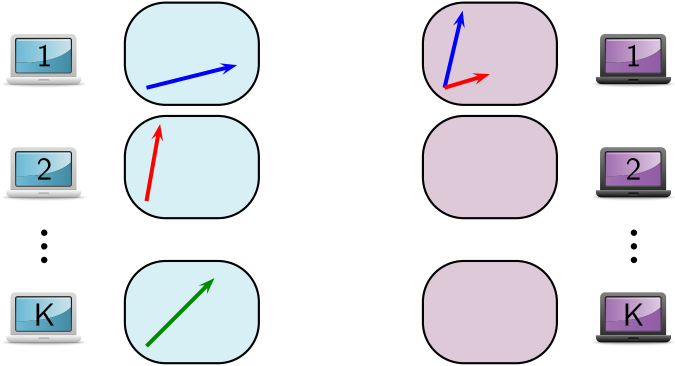


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

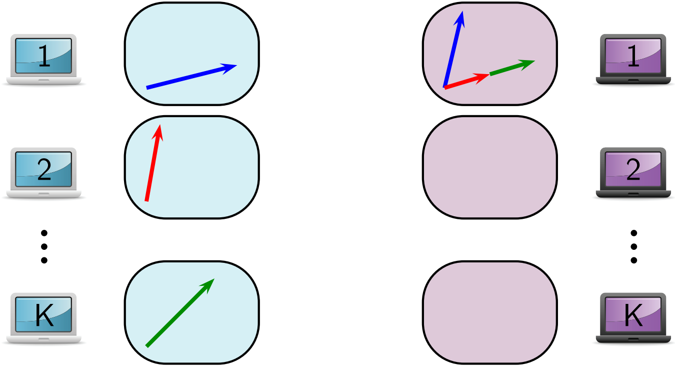


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.



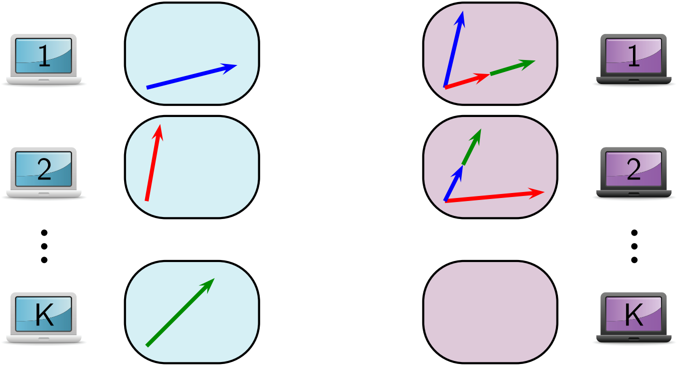
- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

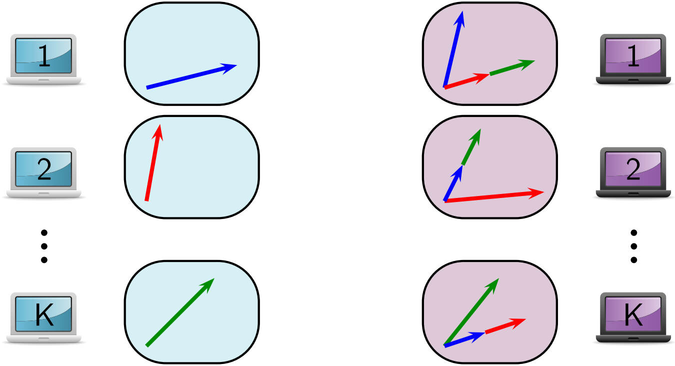


- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment



- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.



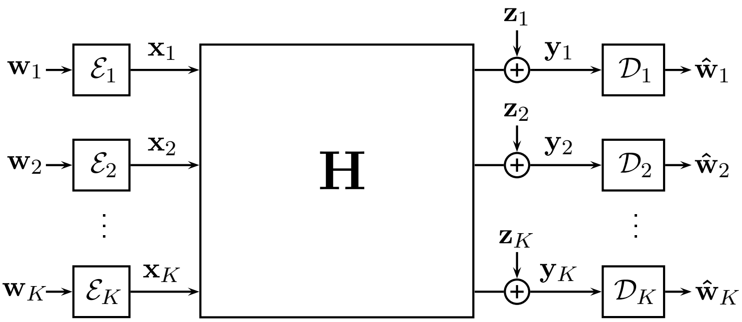
- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Interference Alignment

- **Cadambe-Jafar '08:** Alignment can achieve $K/2$ degrees-of-freedom for the K-user interference channel.

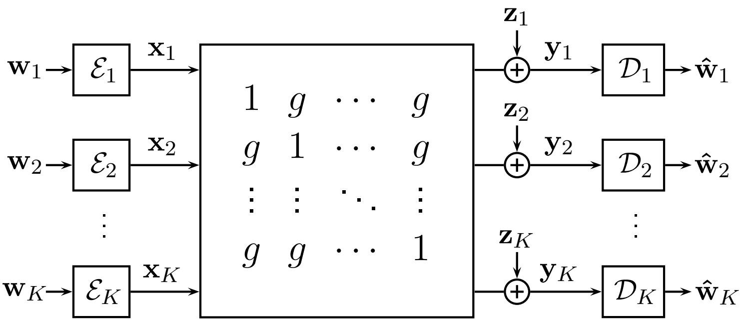
- **Birk-Kol '98:** Alignment for index coding. **Maddah-Ali - Motahari - Khandani '08:** Alignment for the MIMO X channel. See **Jafar '11** monograph (or recent e-book) for a richer history.

Symmetric K-User Gaussian Interference Channel



- Signal space alignment (e.g., beamforming) is infeasible.
- Signal scale alignment attains $K/2$ degrees-of-freedom for almost all channel gains, **Motahari et al. '09, Wu-Shamai-Verdu '11.**
- At finite SNR, the approximate capacity known in some special cases: two-user **Etkin-Tse-Wang '08**, many-to-one and one-to-many **Bresler-Parekh-Tse '10**, cyclic **Zhou-Yu '13**.

Symmetric K-User Gaussian Interference Channel



- Signal space alignment (e.g., beamforming) is infeasible.
- Signal scale alignment attains $K/2$ degrees-of-freedom for almost all channel gains, **Motahari et al. '09, Wu-Shamai-Verdu '11.**
- At finite SNR, the approximate capacity known in some special cases: two-user **Etkin-Tse-Wang '08**, many-to-one and one-to-many **Bresler-Parekh-Tse '10**, cyclic **Zhou-Yu '13**.
- Let's look at the symmetric case.

Effective Multiple-Access Channel

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Effective Multiple-Access Channel

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Successive Cancellation Decoding:

- Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$, then decode $\mathbf{x}_k$.
- Only optimal when interference is very strong, **Sridharan et al. '08**.

- Each receiver sees an effective two-user multiple-access channel,

$$\mathbf{y}_k = \mathbf{x}_k + g \sum_{\ell \neq k} \mathbf{x}_\ell + \mathbf{z}_k \ .$$

Successive Cancellation Decoding:

- Decode and subtract interference $\sum_{\ell \neq k} \mathbf{x}_\ell$, then decode $\mathbf{x}_k$.

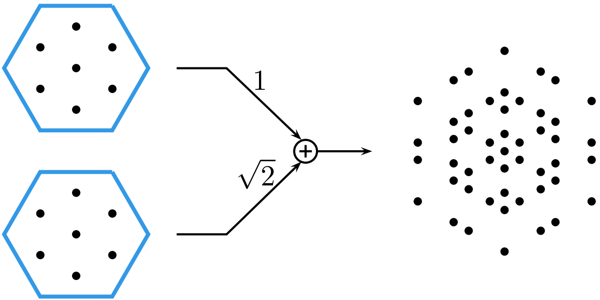
- Only optimal when interference is very strong, **Sridharan et al. '08**.

Joint Decoding:

- Direct analysis is hindered by dependencies between codeword pairs.

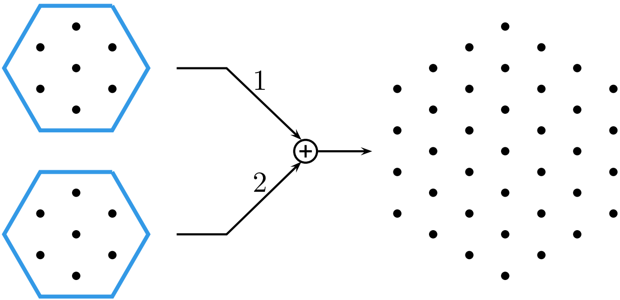
- Existing work only applies at very high SNR, **Ordentlich-Erez '13.**

Example: Two-User Lattice Alignment



- Two lattice codewords can be recovered from their linear combination if the ratio of the coefficients is irrational.

Example: Two-User Lattice Alignment



- Two lattice codewords can be recovered from their linear combination if the ratio of the coefficients is irrational.
- If the ratio is rational, it is not always possible to uniquely identify the pair of codewords.

Alignment via Two Equations

- High SNR behavior: $K/2$ degrees-of-freedom can be attained up to a set of channel gains of measure zero. Loss of degrees-of-freedom for rational coefficients. **Etkin-Ordentlich '09, Motahari et al. '09, Wu-Shamai-Verdu '11.**

Alignment via Two Equations

- High SNR behavior: $K/2$ degrees-of-freedom can be attained up to a set of channel gains of measure zero. Loss of degrees-of-freedom for rational coefficients. **Etkin-Ordentlich '09, Motahari et al. '09, Wu-Shamai-Verdu '11.**

- **Ordentlich-Erez-Nazer '14:** Decode two linear combinations:

$$a_1 \mathbf{x}_k + a_2 \sum_{\ell \neq k} \mathbf{x}_\ell \qquad\qquad b_1 \mathbf{x}_k + b_2 \sum_{\ell \neq k} \mathbf{x}_\ell$$

  using the compute-and-forward framework. If the coefficients are linearly independent, we can solve for the desired message.

Alignment via Two Equations

- High SNR behavior: $K/2$ degrees-of-freedom can be attained up to a set of channel gains of measure zero. Loss of degrees-of-freedom for rational coefficients. **Etkin-Ordentlich '09, Motahari et al. '09, Wu-Shamai-Verdu '11.**

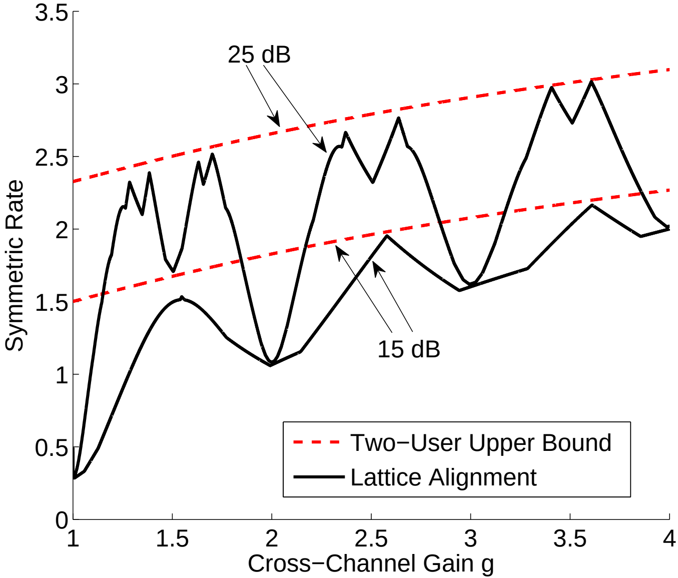
- **Ordentlich-Erez-Nazer '14:** Decode two linear combinations:

$$a_1 \mathbf{x}_k + a_2 \sum_{\ell \neq k} \mathbf{x}_\ell \qquad\qquad b_1 \mathbf{x}_k + b_2 \sum_{\ell \neq k} \mathbf{x}_\ell$$

  using the compute-and-forward framework. If the coefficients are linearly independent, we can solve for the desired message.

- Set of "bad rationals" depends on the SNR. Only rationals with denominator $\mathsf{SNR}^{1/4}$ or smaller cause issues.

Symmetric K-User Gaussian Interference Channel



Approximate Capacity Results: Strong Regime

- Using the fact that the sum of the computation rates is nearly equal to the multiple-access sum capacity, we can approximate the sum capacity of the symmetric K-user Gaussian interference channel in all regimes.

$$R_{\mathsf{sym}} > \frac{1}{2}\log\left(1 + (1+2g^2)\mathsf{SNR}\right) - \max_{\mathbf{a}\in\mathbb{Z}^2} R_{\mathsf{comp}}\left([1\ g]^{\mathsf{T}}, \mathbf{a}\right) - 1$$

- Via basic results from Diophantine approximation, we can approximate the sum capacity up to an outage set.

Approximate Capacity Results: Strong Regime

- Using the fact that the sum of the computation rates is nearly equal to the multiple-access sum capacity, we can approximate the sum capacity of the symmetric K-user Gaussian interference channel in all regimes.

$$R_{\mathsf{sym}} > \frac{1}{2}\log\left(1 + (1+2g^2)\mathsf{SNR}\right) - \max_{\mathbf{a}\in\mathbb{Z}^2} R_{\mathsf{comp}}\left([1\ g]^{\mathsf{T}}, \mathbf{a}\right) - 1$$

- Via basic results from Diophantine approximation, we can approximate the sum capacity up to an outage set.

- **Sample Result:** In the strong interference regime,

$$\frac{1}{4}\log^{+}(g^2\mathsf{SNR}) - \frac{c}{2} - 3 \leq C_{\mathsf{sym}} \leq \frac{1}{4}\log^{+}(g^2\mathsf{SNR}) + 1$$

  for all channel gains except for an outage set whose measure is a fraction of 2^{-c} of the interval $1 < |g| < \sqrt{\mathsf{SNR}}$, for any $c > 0$.

Generalizations

- What about beyond the symmetric case?

Generalizations

- What about beyond the symmetric case?
- **Ntranos-Cadambe-Nazer-Caire '13:** Framework for lattice interference alignment for any setting where we have "stream-by-stream" alignment.

Some topics we did not have a chance to cover:

- Relaying: **Wilson-Narayanan-Pfister-Sprintson '10**, **Nam-Chung-Lee '10, '11**, **Goseling-Gastpar-Weber '11**, **Song-Devroye '13**, **Nokleby-Aazhang '12**
- Cellular and MIMO Networks: **Sanderovich-Peleg-Shamai '11**, **Nazer-Sanderovich-Gastpar-Shamai '09**, **Zhan-Nazer-Erez-Gastpar '12**, **Hong-Caire '13**, **Ordentlich-Erez '13**
- Distributed Dirty-Paper Coding: **Philosof-Zamir '09**, **Philosof-Zamir-Erez-Khisti '11**, **Wang '12**
- Joint Source-Channel Coding: **Kochman-Zamir '09**, **Nazer-Gastpar '07, '08**, **Soundararajan-Vishwanath '12**
- Physical-Layer Secrecy: **He-Yener '11, '14**, **Kashyap-Shashank-Thangaraj '12**

Concluding Remarks

- Codes with algebraic structure can sometimes outperform i.i.d. ensembles.
- Ongoing efforts towards developing an algebraic framework for network source and channel coding.
- Preliminary efforts have focused on the Gaussian case but discrete memoryless analogues of these results now seem within reach.
- An open question: How should we choose the underlying algebraic structure?
- Tutorial slides from 2014 European School of Information Theory available on my website.
- Upcoming textbook by Ram Zamir on "Lattice Coding for Signals and Networks."