Towards an Algebraic Network Information Theory

Bobak Nazer (BU)

Joint work with Sung Hoon Lim (EPFL), Chen Feng (UBC), and Michael Gastpar (EPFL).

DIMACS Workshop on Network Coding: The Next 15 Years

December 17th, 2015

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 codewords elementwise i.i.d.

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 codewords elementwise i.i.d.
- 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 codewords elementwise i.i.d.
- 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 pmfs.

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 codewords elementwise i.i.d.
- 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 pmfs.
- 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 codewords elementwise i.i.d.
- 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 pmfs.
- 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 codewords elementwise i.i.d.
- 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 pmfs.
- 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 random i.i.d. 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.
- Most of the initial efforts have focused on Gaussian networks and have employed nested 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.
- Coding schemes exhibit behavior not found via i.i.d. ensembles.
- However, some classical coding techniques are still unavailable.
- Most of the initial efforts have focused on Gaussian networks and have employed nested lattice codebooks.
- Are these just a collection of intriguing examples or elements of a more general theory?

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.

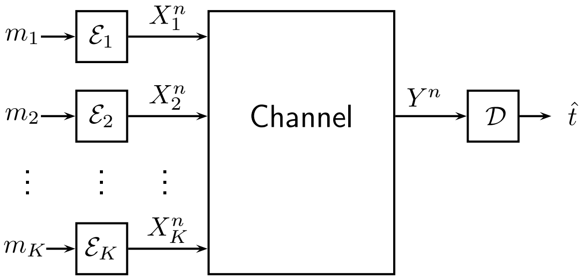
Algebraic Approach:

- 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.
- Most of the initial efforts have focused on Gaussian networks and have employed nested lattice codebooks.
- Are these just a collection of intriguing examples or elements of a more general theory?

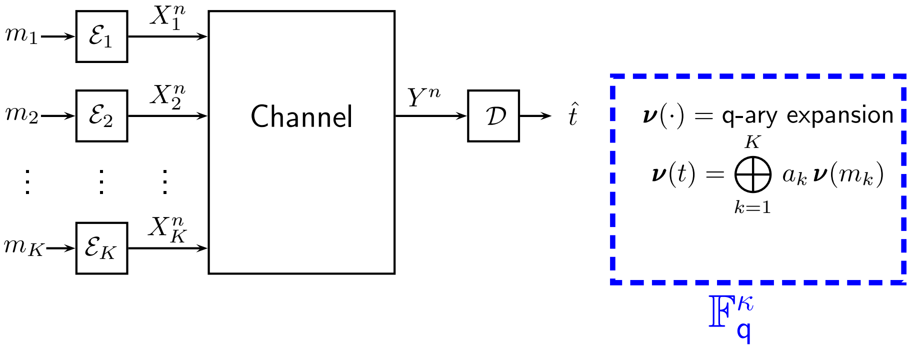
This Talk: We build on previous work and propose a joint typicality approach to algebraic network information theory.

Goal: Send a linear combination of the messages to the receiver.

Goal: Send a linear combination of the messages to the receiver.

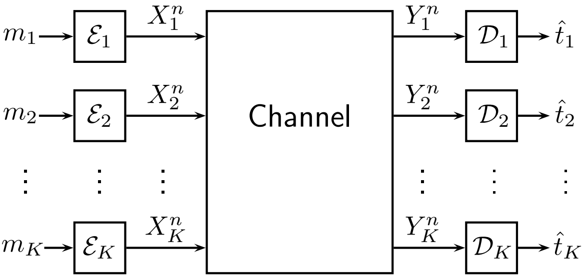


Goal: Send a linear combination of the messages to the receiver.



Compute-and-Forward

Goal: Send linear combinations of the messages to the receivers.



$\boldsymbol{\nu}(\cdot) =$ q-ary expansion

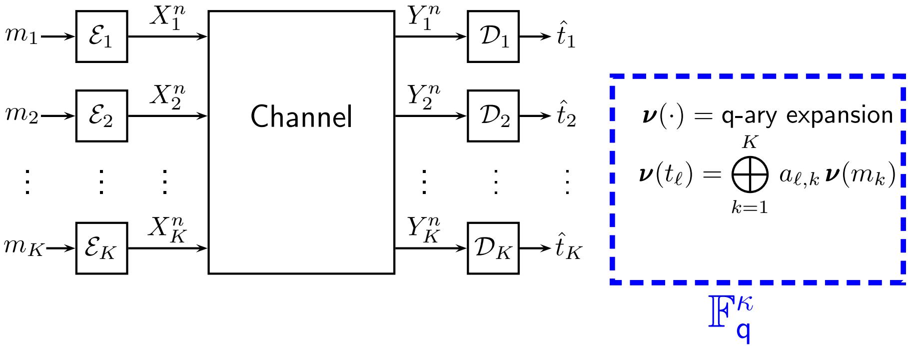
$$\boldsymbol{\nu}(t_\ell) = \bigoplus_{k=1}^{K} a_{\ell,k}\, \boldsymbol{\nu}(m_k)$$

$\mathbb{F}_{\mathsf{q}}^{\kappa}$

Compute-and-Forward

Goal: Send linear combinations of the messages to the receivers.

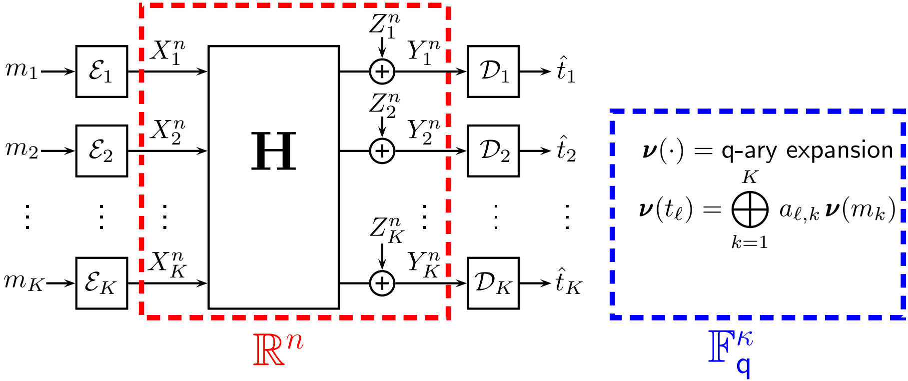
- Compute-and-forward can serve as a framework for communicating messages across a network (e.g., relaying, MIMO uplink/downlink, interference alignment).



Goal: Send linear combinations of the messages to the receivers.

- Compute-and-forward can serve as a framework for communicating messages across a network (e.g., relaying, MIMO uplink/downlink, interference alignment).

- Much of the recent work has focused on Gaussian networks.



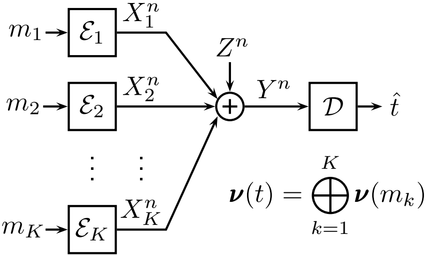
The Usual Approach

The Usual Approach

- Symmetric Gaussian MAC.

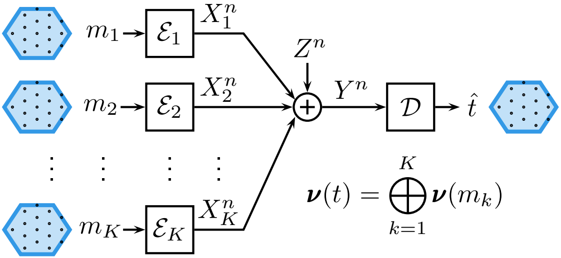
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP.$$



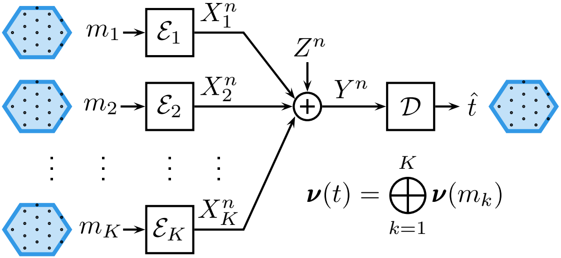
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP.$$
- Use nested lattice codes.



Computation over Gaussian MACs

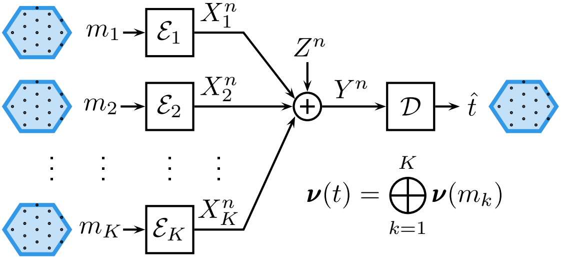
- Symmetric Gaussian MAC.
- Equal power constraints:
  $\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP$.
- Use nested lattice codes.



- **Wilson-Narayanan-Pfister-Sprintson '10, Nazer-Gastpar '11:** Decoding is successful if the rates satisfy

$$R_k < \frac{1}{2}\log^+\left(\frac{1}{2} + P\right) .$$

- Symmetric Gaussian MAC.
- Equal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \le nP.$$
- Use nested lattice codes.



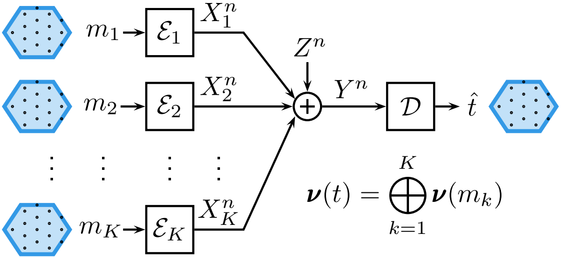
- **Wilson-Narayanan-Pfister-Sprintson '10, Nazer-Gastpar '11:** Decoding is successful if the rates satisfy

$$R_k < \frac{1}{2}\log^+\left(\frac{1}{2} + P\right) .$$

- Cut-set upper bound is $\frac{1}{2}\log(1+P)$.

Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints:
  $\mathbb{E}\|\mathbf{x}_\ell\|^2 \le nP.$
- Use nested lattice codes.



- **Wilson-Narayanan-Pfister-Sprintson '10, Nazer-Gastpar '11:** Decoding is successful if the rates satisfy

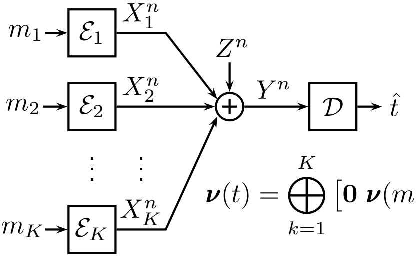
$$R_k < \frac{1}{2}\log^+\left(\frac{1}{2} + P\right) .$$

- Cut-set upper bound is $\frac{1}{2}\log(1+P)$.
- What about the "1+"? Still open! (Ice wine problem.)

Computation over Gaussian MACs

- How about general Gaussian MACs?

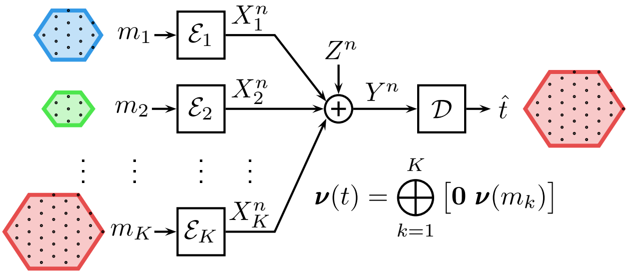
- How about general Gaussian MACs?
- Model using unequal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \le nP_\ell.$$



$$\boldsymbol{\nu}(t) = \bigoplus_{k=1}^{K} \left[\mathbf{0}\ \boldsymbol{\nu}(m_k)\right]$$

Computation over Gaussian MACs

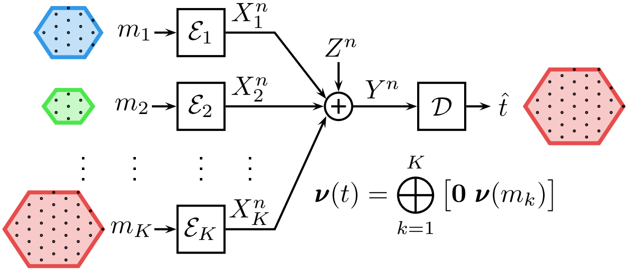
- How about general Gaussian MACs?
- Model using unequal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP_\ell.$$



- **Nam-Chung-Lee '11:** At each transmitter, use the same fine lattice and a different coarse lattice, chosen to meet the power constraint.

- How about general Gaussian MACs?
- Model using unequal power constraints:

$$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP_\ell.$$

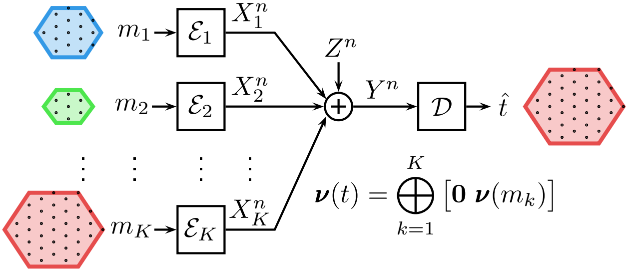


- **Nam-Chung-Lee '11:** At each transmitter, use the same fine lattice and a different coarse lattice, chosen to meet the power constraint.
- Decoding is successful if the rates satisfy

$$R_\ell < \frac{1}{2}\log^+\left(\frac{P_\ell}{\sum_{i=1}^{L} P_i} + P_\ell\right) .$$

Computation over Gaussian MACs

- How about general Gaussian MACs?
- Model using unequal power constraints:
  $$\mathbb{E}\|\mathbf{x}_\ell\|^2 \leq nP_\ell.$$



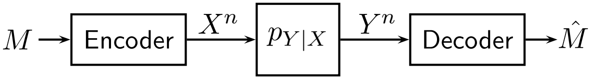
- **Nam-Chung-Lee '11:** At each transmitter, use the same fine lattice and a different coarse lattice, chosen to meet the power constraint.

- Decoding is successful if the rates satisfy

$$R_\ell < \frac{1}{2}\log^+\left(\frac{P_\ell}{\sum_{i=1}^{L} P_i} + P_\ell\right) .$$

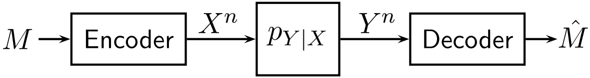
- **Nazer-Cadambe-Ntranos-Caire '15:** Expanded compute-and-forward framework to link unequal power setting to finite fields.

Point-to-Point Channels



- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Point-to-Point Channels

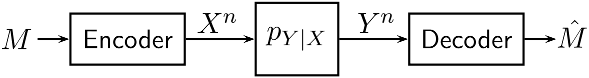


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Theorem (Shannon '48)

$$C = \max_{p_X(x)} I(X;Y)$$

Point-to-Point Channels

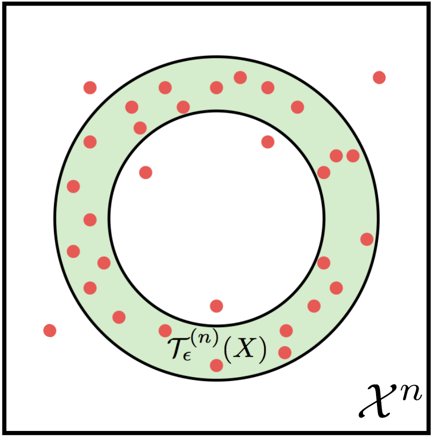


- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Theorem (Shannon '48)

$$C = \max_{p_X(x)} I(X;Y)$$

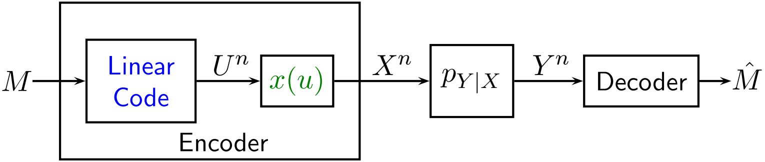
- Proof relies on random i.i.d. codebooks combined with joint typicality decoding.



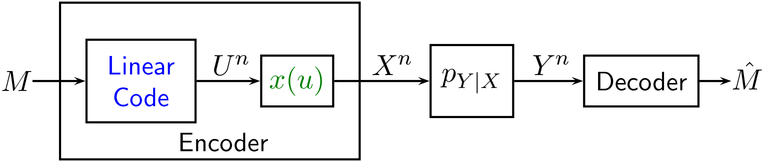
Random i.i.d. Codes

- Codewords are independent of one another.
- Can directly target an input distribution $p_X(x)$.

Point-to-Point Channels: Linear Codes

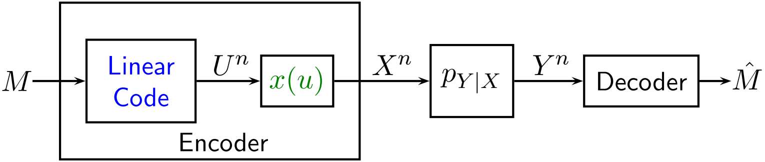


Code Construction:



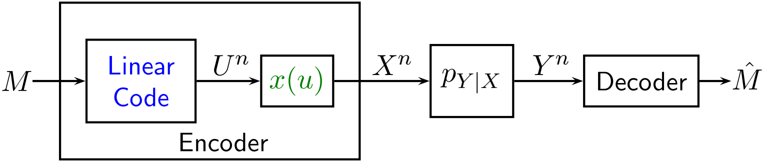
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.



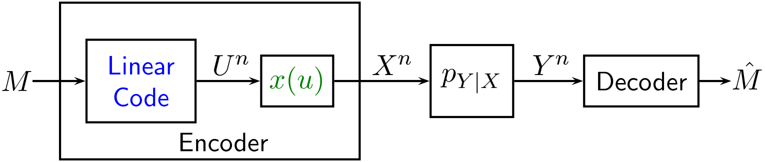
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.



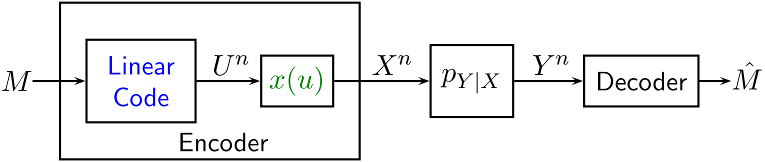
Code Construction:

- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\text{Unif}(\mathbb{F}_{\mathsf{q}})$. Let G be a realization.



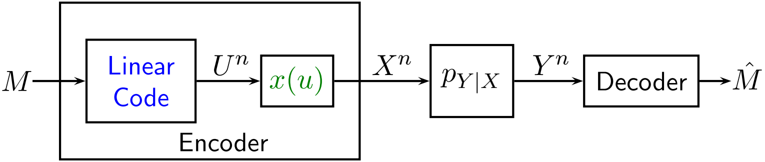
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let d^n be a realization.



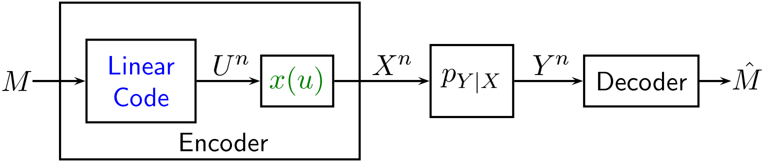
Code Construction:

- Pick a finite field $\mathbb{F}_\mathsf{q}$ and a symbol mapping $x : \mathbb{F}_\mathsf{q} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_\mathsf{q}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_\mathsf{q}^{\kappa}$.



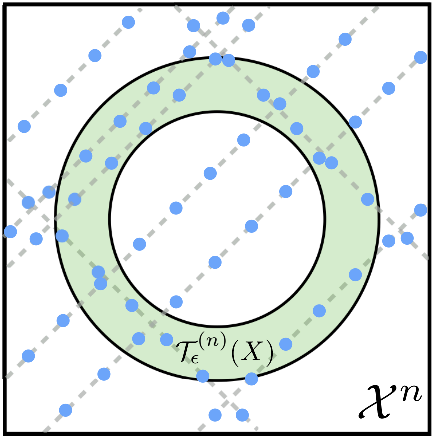
Code Construction:

- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codeword for message m is $u^n(m) = \boldsymbol{\nu}(m)\mathsf{G} \oplus d^n$.



Code Construction:

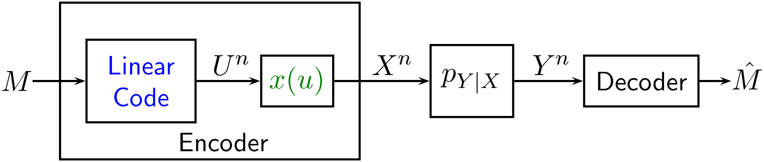
- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let G be a realization.
- Draw a random shift (or "dither") D^n elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$. Let d^n be a realization.
- Take q-ary expansion of message m into the vector $\boldsymbol{\nu}(m) \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codeword for message m is $u^n(m) = \boldsymbol{\nu}(m)\mathsf{G} \oplus d^n$.
- Channel input at time i is $x_i(m) = x(u_i(m))$.



Random Linear Codes

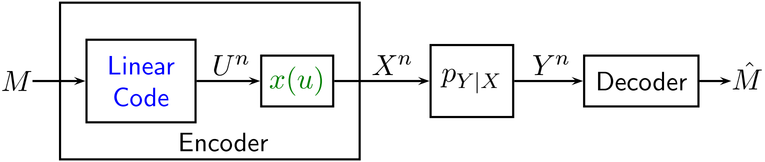
- Codewords are pairwise independent of one another.
- Codewords are uniformly distributed over $\mathbb{F}_q^n$.

Point-to-Point Channels: Linear Codes



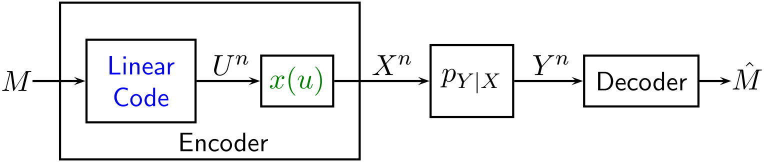
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**

Point-to-Point Channels: Linear Codes



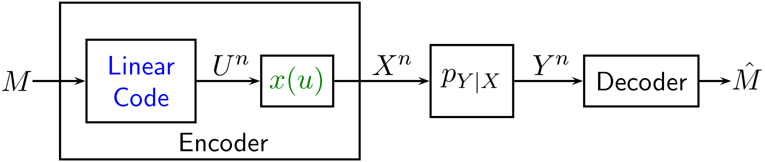
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_\mathsf{q}$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. This can attain the capacity.

Point-to-Point Channels: Linear Codes



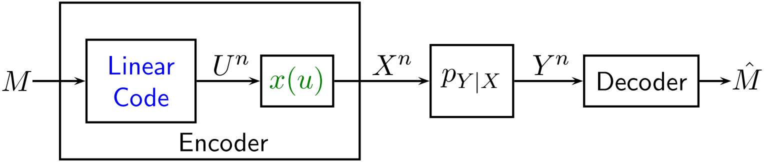
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_\mathsf{q}$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.

Point-to-Point Channels: Linear Codes

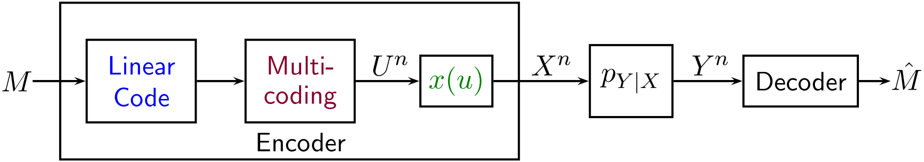


- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_\mathsf{q}$ with q $\gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.
- **Padakandla-Pradhan '13:** It is possible to shape the input distribution using nested linear codes.

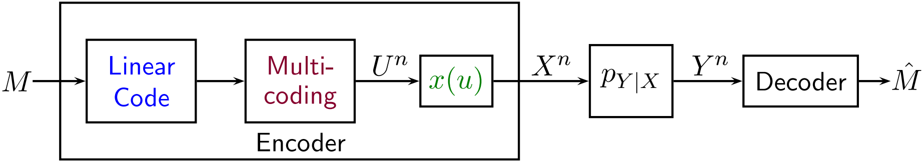
Point-to-Point Channels: Linear Codes



- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\text{Unif}(\mathbb{F}_q)$. This can attain the capacity.
- This will not work for us. Roughly speaking, if each encoder has a different input distribution, the symbol mappings may be quite different, which will disrupt the linear structure of the codebook.
- **Padakandla-Pradhan '13:** It is possible to shape the input distribution using nested linear codes.
- Basic idea: Generate many codewords to represent one message. Search in this "bin" to find a codeword with the desired type, i.e., multicoding.

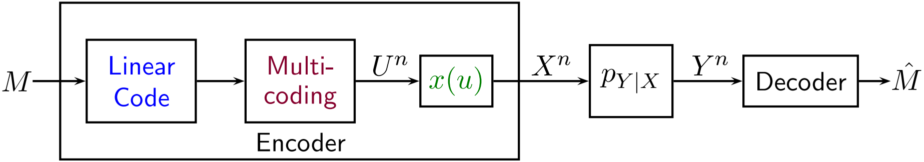


Code Construction:



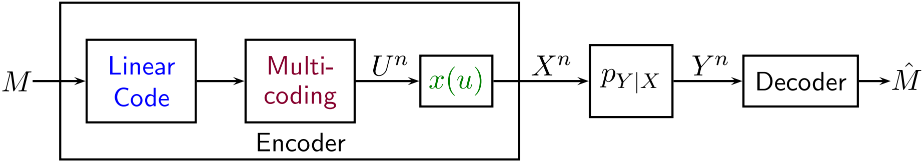
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.



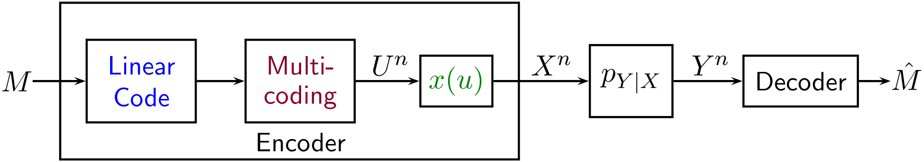
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(\mathsf{q})$.



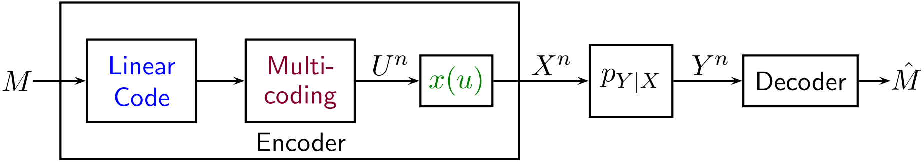
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.



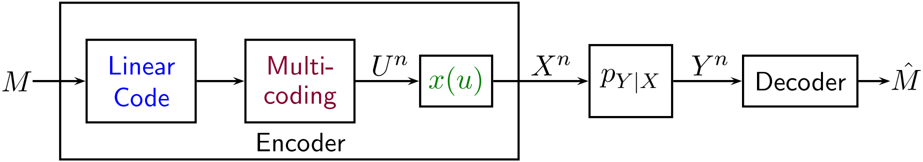
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R}) / \log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.
- Take q-ary expansions $\big[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.



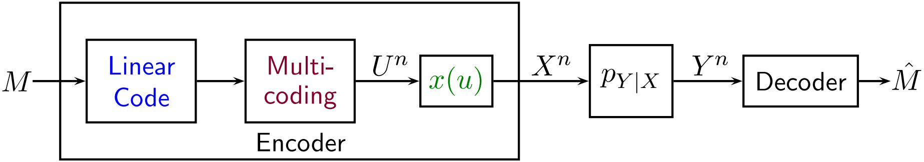
Code Construction:

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R}) / \log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.
- Take q-ary expansions $\big[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear codewords: $u^n(m, l) = \big[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\big]\mathsf{G} \oplus d^n$.


M
Linear Code
Multi-coding
U^n
$x(u)$
Encoder
X^n
$p_{Y|X}$
Y^n
Decoder
$\hat{M}$

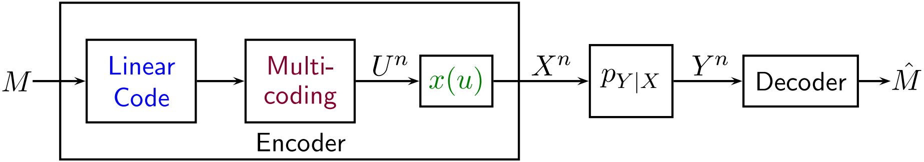
Encoding:

Point-to-Point Channels: Linear Codes + Multicoding



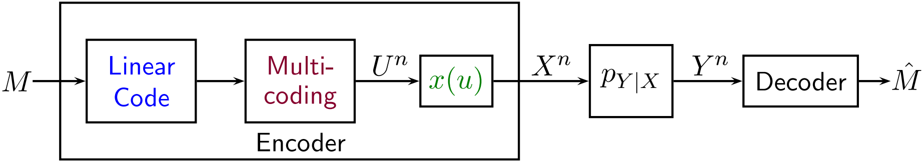
Encoding:

- Fix $p(u)$ and $x(u)$.



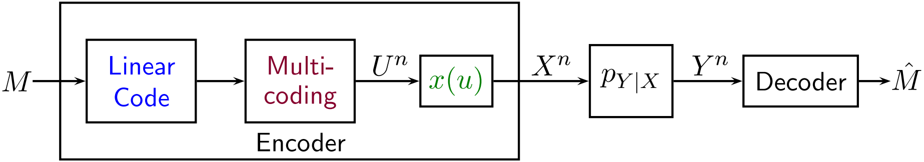
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m, l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$



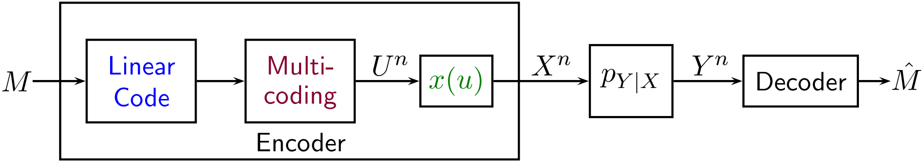
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).



Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

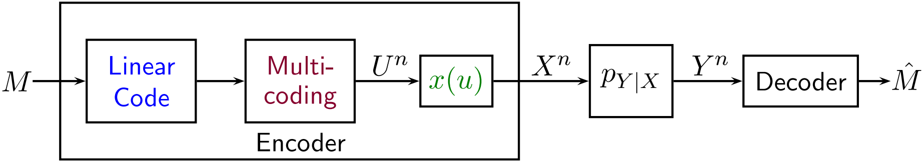


Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

Point-to-Point Channels: Linear Codes + Multicoding



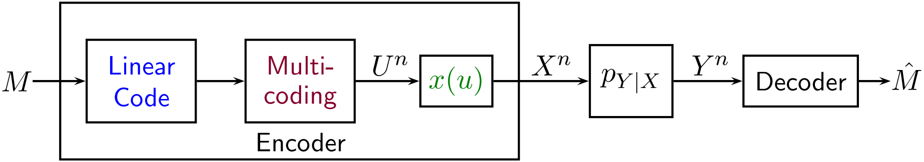
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

- Joint Typicality Decoding: Find the unique index $\hat{m}$ such that $\big(u^n(\hat{m},\hat{l}),\, y^n\big) \in \mathcal{T}_{\epsilon}^{(n)}(U,Y)$ for some index $\hat{l}$.

Point-to-Point Channels: Linear Codes + Multicoding



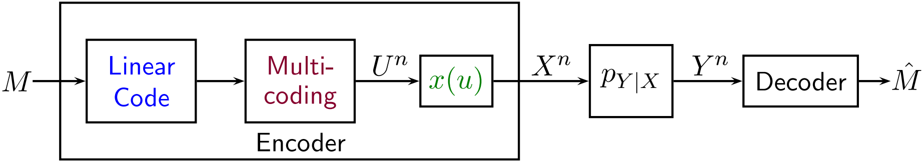
Encoding:

- Fix $p(u)$ and $x(u)$.
- Multicoding: For each m, find an index l such that $u^n(m,l) \in \mathcal{T}_{\epsilon'}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_{\mathsf{q}})$ (where p_{q} is uniform over $\mathbb{F}_{\mathsf{q}}$).
- Transmit $x_i = x\big(u_i(m,l)\big)$.

Decoding:

- Joint Typicality Decoding: Find the unique index $\hat{m}$ such that $\big(u^n(\hat{m},\hat{l}),\, y^n\big) \in \mathcal{T}_{\epsilon}^{(n)}(U,Y)$ for some index $\hat{l}$.
- Succeeds w.h.p. if $R + \hat{R} < I(U;Y) + D(p_U \| p_{\mathsf{q}})$

Point-to-Point Channels: Linear Codes + Multicoding



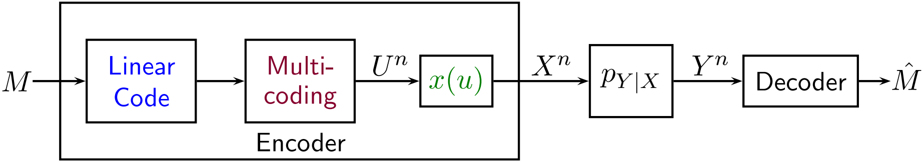
Theorem (Padakandla-Pradhan '13)

Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

Point-to-Point Channels: Linear Codes + Multicoding



Theorem (Padakandla-Pradhan '13)

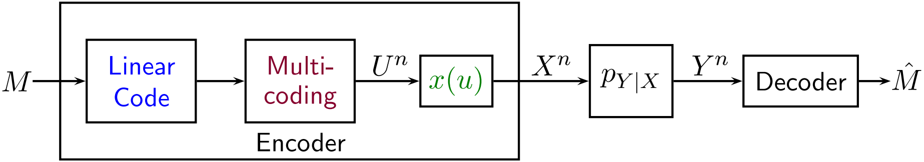
Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

- This is the basic coding framework that we will use for each transmitter.

Point-to-Point Channels: Linear Codes + Multicoding



Theorem (Padakandla-Pradhan '13)

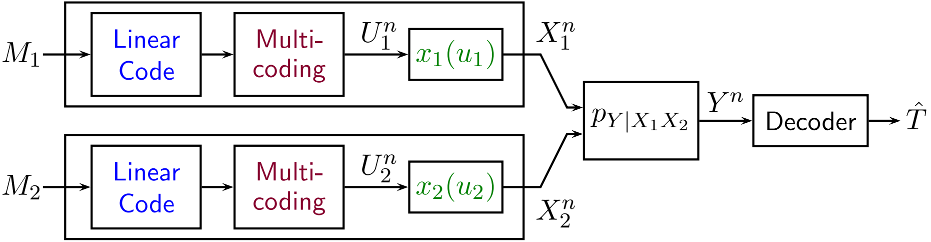
Any rate R satisfying

$$R < \max_{p(u),\, x(u)} I(U;Y)$$

is achievable. This is equal to the capacity if $\mathsf{q} \geq |\mathcal{X}|$.

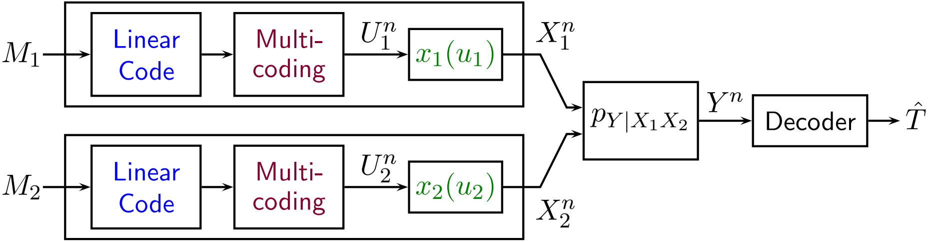
- This is the basic coding framework that we will use for each transmitter.
- Next, let's examine a two-transmitter, one-receiver "compute-and-forward" network.

Nested Linear Coding Architecture



Code Construction:

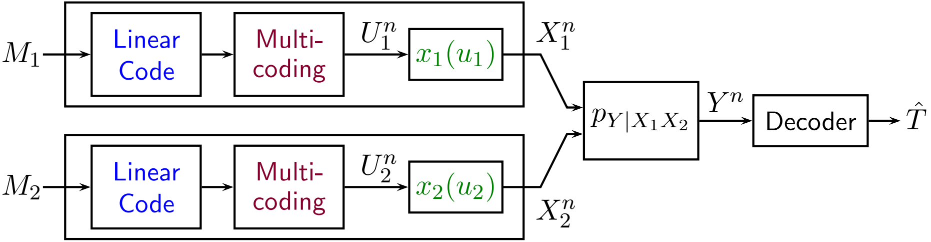
- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.



Code Construction:

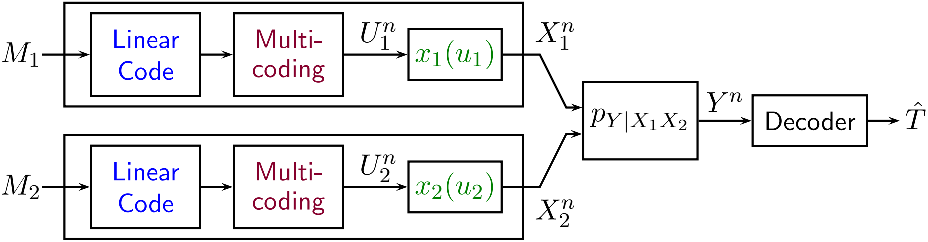
- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.

Nested Linear Coding Architecture



Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.

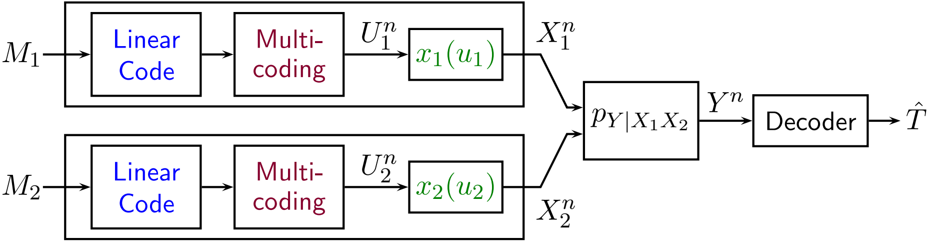


Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\begin{bmatrix}\boldsymbol{\nu}(m_1) & \boldsymbol{\nu}(l_1)\end{bmatrix} \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

  $\begin{bmatrix}\boldsymbol{\nu}(m_2) & \boldsymbol{\nu}(l_2) & \mathbf{0}\end{bmatrix} \in \mathbb{F}_{\mathsf{q}}^{\kappa}$ Zero-padding

Nested Linear Coding Architecture

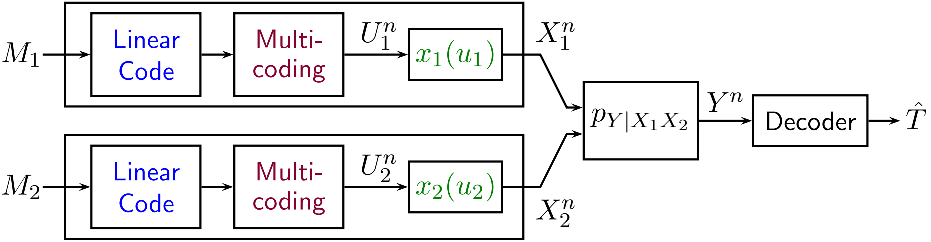


Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\})/\log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\left[\boldsymbol{\eta}(m_1, l_1)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

$$\left[\boldsymbol{\eta}(m_2, l_2)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$$

Nested Linear Coding Architecture



Code Construction:

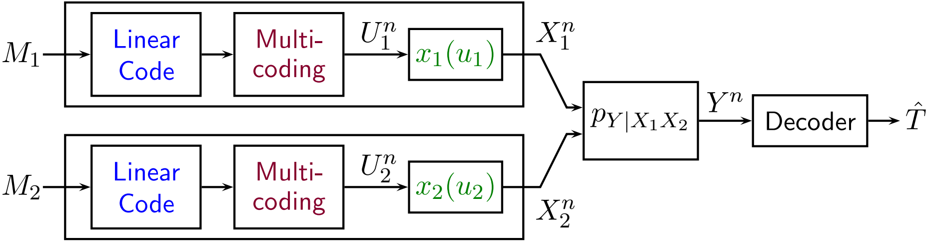
- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1,\ R_2 + \hat{R}_2\}) / \log(\mathsf{q})$.
- Pick generator matrix G and dithers $d_1^n,\ d_2^n$ as before.
- Take q-ary expansions $\left[\boldsymbol{\eta}(m_1, l_1)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

  $$\left[\boldsymbol{\eta}(m_2, l_2)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$$

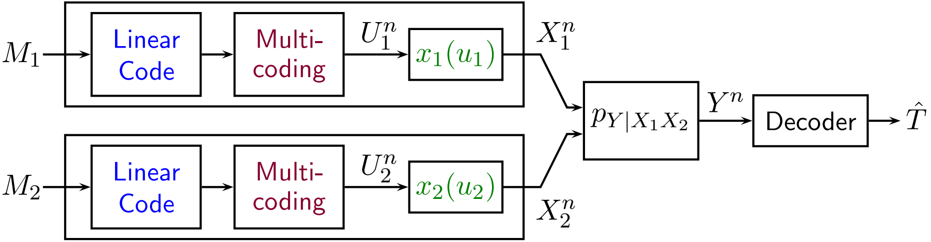
- Linear codewords: $u_1^n(m_1, l_1) = \boldsymbol{\eta}(m_1, l_1)\mathsf{G} \oplus d_1^n$

  $$u_2^n(m_2, l_2) = \boldsymbol{\eta}(m_2, l_2)\mathsf{G} \oplus d_2^n$$

Nested Linear Coding Architecture



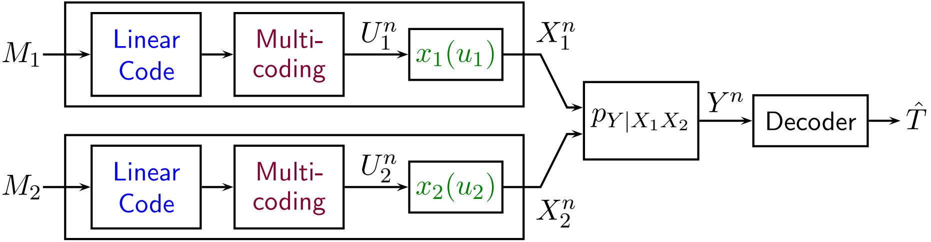
Encoding:



Encoding:

- Fix $p(u_1)$, $p(u_2)$, $x_1(u_1)$, and $x_2(u_2)$.

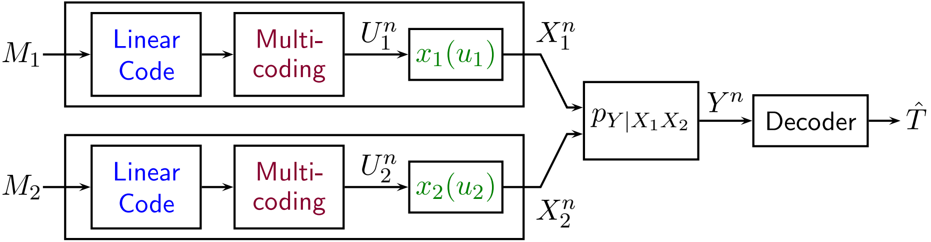
Nested Linear Coding Architecture



Encoding:

- Fix $p(u_1)$, $p(u_2)$, $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$.

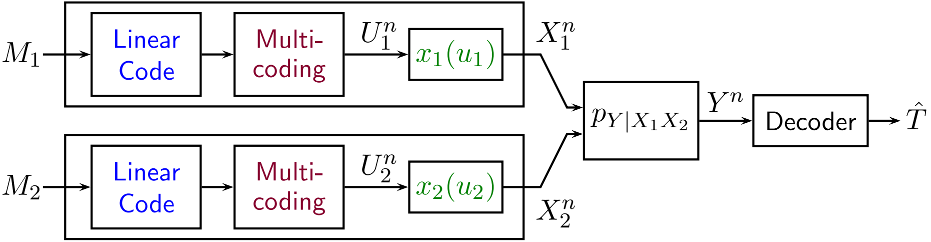
Nested Linear Coding Architecture



Encoding:

- Fix $p(u_1)$, $p(u_2)$, $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$.
- Succeeds w.h.p. if $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}})$.

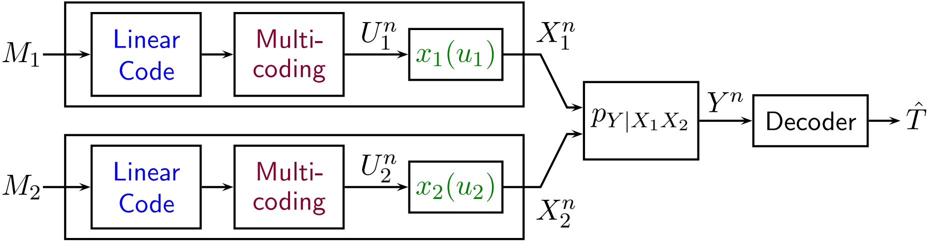
Nested Linear Coding Architecture



Encoding:

- Fix $p(u_1)$, $p(u_2)$, $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$.
- Succeeds w.h.p. if $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}})$.
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big)$.

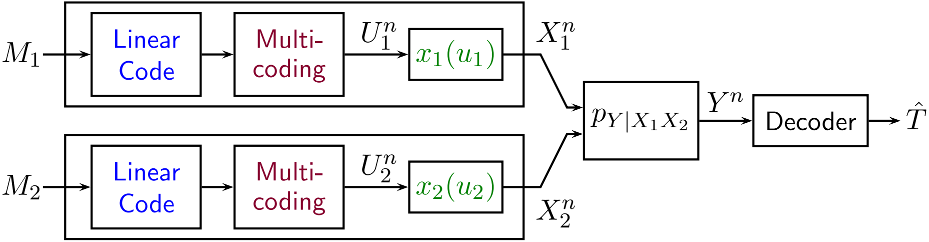
Nested Linear Coding Architecture



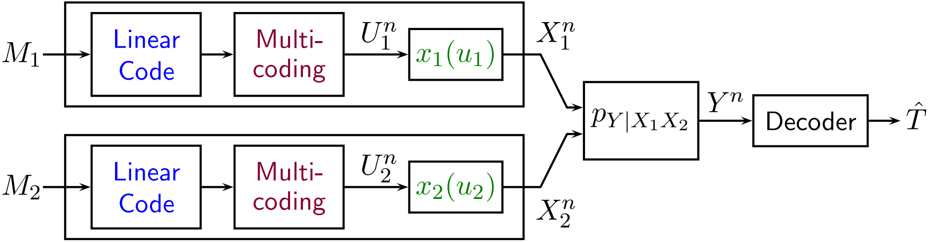
Encoding:

- Fix $p(u_1)$, $p(u_2)$, $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$.
- Succeeds w.h.p. if $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}})$.
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big)$.

Nested Linear Coding Architecture



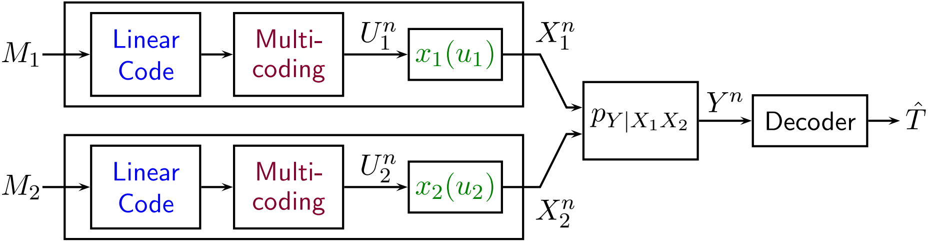
Computation Problem:



Computation Problem:

- Consider the coefficients $\mathbf{a} \in \mathbb{F}_q^2$, $\mathbf{a} = [a_1,\ a_2]$

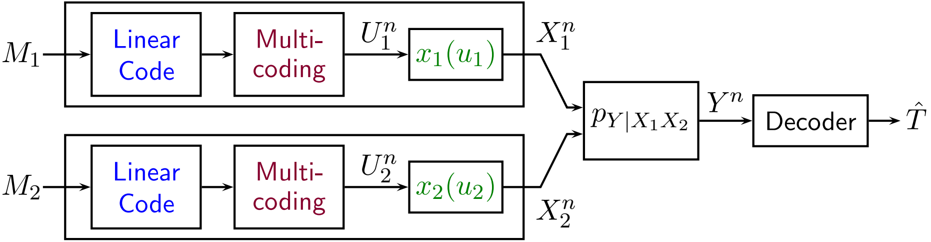
Nested Linear Coding Architecture



Computation Problem:

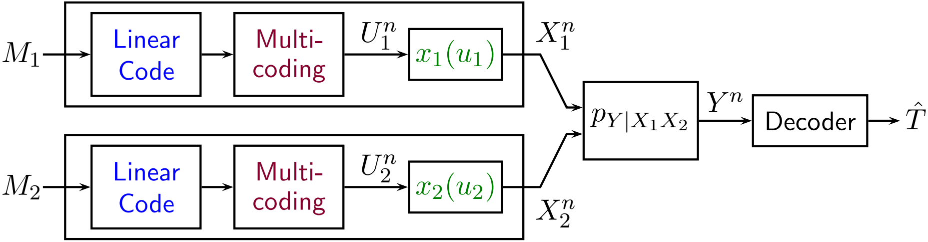
- Consider the coefficients $\mathbf{a} \in \mathbb{F}_\mathsf{q}^2$, $\mathbf{a} = [a_1,\ a_2]$
- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, the linear combination of codewords with coefficient vector $\mathbf{a}$ is

$$\begin{aligned} & a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ & = \big[a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\big] \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \\ & = \boldsymbol{\nu}(t) \mathsf{G} \oplus d_w^n \\ & = w^n(t), \quad t \in [2^{n \max\{R_1 + \hat{R}_1, R_2 + \hat{R}_2\}}] \end{aligned}$$



Computation Problem:

- Let M_k be the chosen message and L_k the chosen index from the multicoding step.

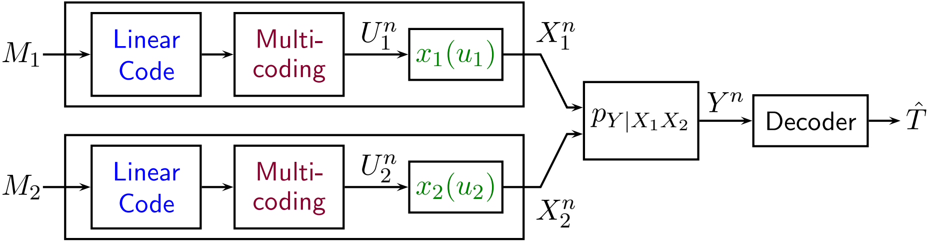


Computation Problem:

- Let M_k be the chosen message and L_k the chosen index from the multicoding step.
- Decoder wants a linear combination of the codewords:

$$W^n(T) = a_1 U_1^n(M_1, L_1) \oplus a_2 U_2^n(M_2, L_2)$$

Nested Linear Coding Architecture



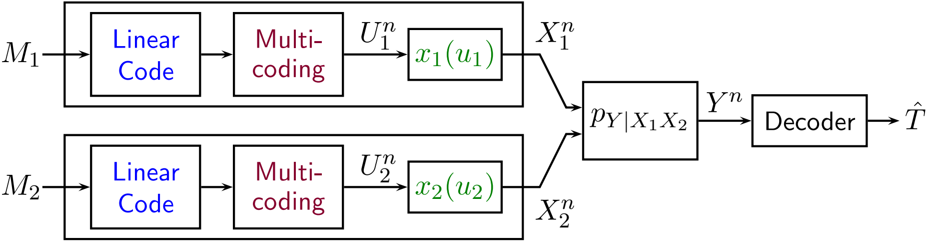
Computation Problem:

- Let M_k be the chosen message and L_k the chosen index from the multicoding step.
- Decoder wants a linear combination of the codewords:

$$W^n(T) = a_1 U_1^n(M_1, L_1) \oplus a_2 U_2^n(M_2, L_2)$$

- Decoder: $\hat{t}(y^n) \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$, $y^n \in \mathcal{Y}^n$
- Probability of Error: $\mathsf{P}_\epsilon^{(n)} = \mathsf{P}\{T \neq \hat{T}\}$

Nested Linear Coding Architecture

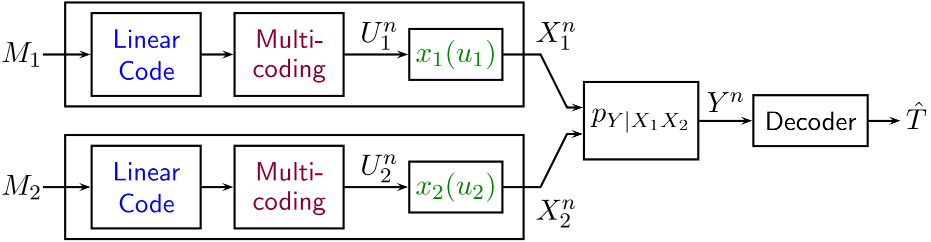


Computation Problem:

- Let M_k be the chosen message and L_k the chosen index from the multicoding step.
- Decoder wants a linear combination of the codewords:

$$W^n(T) = a_1 U_1^n(M_1, L_1) \oplus a_2 U_2^n(M_2, L_2)$$

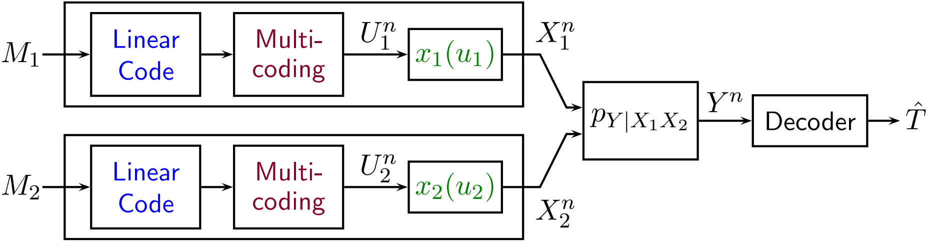
- Decoder: $\hat{t}(y^n) \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$, $y^n \in \mathcal{Y}^n$
- Probability of Error: $\mathsf{P}_\epsilon^{(n)} = \mathsf{P}\{T \neq \hat{T}\}$
- A rate pair is achievable if there exists a sequence of codes such that $\mathsf{P}_\epsilon^{(n)} \to 0$ as $n \to \infty$.



Decoding:

- Joint Typicality Decoding: Find an index $t \in [2^{n\max(R_1+\hat{R}_1, R_2+\hat{R}_2)}]$ such that $(w^n(t), y^n) \in \mathcal{T}_\epsilon^{(n)}$.

Nested Linear Coding Architecture



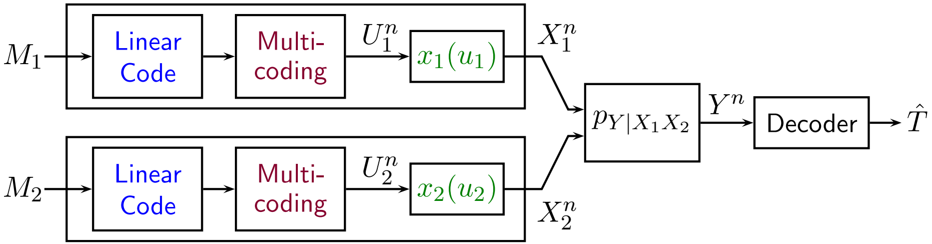
Theorem (Lim-Chen-Nazer-Gastpar Allerton '15)

A rate pair (R_1, R_2) *is achievable if*

$$R_1 < I(W;Y) - I(W;U_2),$$
$$R_2 < I(W;Y) - I(W;U_1),$$

for some $p(u_1)p(u_2)$ *and functions* $x_1(u_1)$, $x_2(u_2)$, *where* $\mathcal{U}_k = \mathbb{F}_\mathsf{q}$, $k = 1, 2$, *and* $W = a_1U_1 \oplus a_2U_2$.

Nested Linear Coding Architecture



Theorem (Lim-Chen-Nazer-Gastpar Allerton '15)

A rate pair (R_1, R_2) *is achievable if*

$$R_1 < I(W;Y) - I(W;U_2),$$
$$R_2 < I(W;Y) - I(W;U_1),$$

for some $p(u_1)p(u_2)$ *and functions* $x_1(u_1)$, $x_2(u_2)$, *where* $\mathcal{U}_k = \mathbb{F}_\mathsf{q}$, $k = 1, 2$, *and* $W = a_1U_1 \oplus a_2U_2$.

- **Padakandla-Pradhan '13:** Special case where $R_1 = R_2$.

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.
- Union bound: $\mathsf{P}_\epsilon^{(n)} \le \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$.

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.
- Union bound: $\mathsf{P}_\epsilon^{(n)} \leq \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$.
- Notice that the L_k depend on the codebook so Y^n and $W^n(t)$ are not independent.

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.
- Union bound: $\mathsf{P}_\epsilon^{(n)} \leq \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$.
- Notice that the L_k depend on the codebook so Y^n and $W^n(t)$ are not independent.
- To get around this issue, we analyze

$$\mathsf{P}(\mathcal{E}) = \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)}, U_1^n(0,0) \in \mathcal{T}_\epsilon^{(n)}, U_2^n(0,0) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$$

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.
- Union bound: $\mathsf{P}_\epsilon^{(n)} \leq \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$.
- Notice that the L_k depend on the codebook so Y^n and $W^n(t)$ are not independent.
- To get around this issue, we analyze

$$\mathsf{P}(\mathcal{E}) = \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)}, U_1^n(0,0) \in \mathcal{T}_\epsilon^{(n)}, U_2^n(0,0) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$$

- Conditioned on $\mathcal{M}$, $Y^n \to (U_1^n(0,0), U_2^n(0,0)) \to W^n(t)$

Proof Sketch

- WLOG assume $\mathcal{M} = \{M_1 = 0, M_2 = 0, L_1 = 0, L_2 = 0\}$.
- Union bound: $\mathsf{P}_\epsilon^{(n)} \leq \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$.
- Notice that the L_k depend on the codebook so Y^n and $W^n(t)$ are not independent.
- To get around this issue, we analyze

$$\mathsf{P}(\mathcal{E}) = \sum_{t \neq 0} \mathsf{P}\big\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)}, U_1^n(0,0) \in \mathcal{T}_\epsilon^{(n)}, U_2^n(0,0) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\big\}$$

- Conditioned on $\mathcal{M}$, $Y^n \to (U_1^n(0,0), U_2^n(0,0)) \to W^n(t)$
- $\mathsf{P}(\mathcal{E})$ tends to zero as $n \to \infty$ if

$$\begin{aligned} &R_k + \hat{R}_k + \hat{R}_1 + \hat{R}_2 \\ &< I(W;Y) + D(p_W || p_{\mathsf{q}}) + D(p_{U_1} || p_{\mathsf{q}}) + D(p_{U_2} || p_{\mathsf{q}}) \end{aligned}$$

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output
  $Y = h_1 X_1 + h_2 X_2 + Z$

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output $Y = h_1 X_1 + h_2 X_2 + Z$
- Want to recover $a_1 X_1^n + a_2 X_2^n$ for some integers a_1, a_2.

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output
  $Y = h_1 X_1 + h_2 X_2 + Z$
- Want to recover $a_1 X_1^n + a_2 X_2^n$ for some integers a_1, a_2.
- Gaussian noise: $Z \sim \mathcal{N}(0, 1)$

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output $Y = h_1 X_1 + h_2 X_2 + Z$
- Want to recover $a_1 X_1^n + a_2 X_2^n$ for some integers a_1, a_2.
- Gaussian noise: $Z \sim \mathcal{N}(0, 1)$
- Usual power constraint: $\mathsf{E}[X_k^2] \leq P$

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output $Y = h_1 X_1 + h_2 X_2 + Z$
- Want to recover $a_1 X_1^n + a_2 X_2^n$ for some integers a_1, a_2.
- Gaussian noise: $Z \sim \mathcal{N}(0, 1)$
- Usual power constraint: $\mathsf{E}[X_k^2] \leq P$
- Via Gaussian quantization arguments, we can recover the following theorem.

Compute-and-Forward over a Gaussian MAC

- Consider a Gaussian MAC with real-valued channel output $Y = h_1 X_1 + h_2 X_2 + Z$
- Want to recover $a_1 X_1^n + a_2 X_2^n$ for some integers a_1, a_2.
- Gaussian noise: $Z \sim \mathcal{N}(0,1)$
- Usual power constraint: $\mathsf{E}[X_k^2] \leq P$
- Via Gaussian quantization arguments, we can recover the following theorem.

Theorem (Nazer-Gastpar '11)

For any channel vector $\mathbf{h}$ *and integer coefficient vector* $\mathbf{a}$*, any rate tuple satisfying* $R_k < R_{\textit{comp}}(\mathbf{h}, \mathbf{a})$ *for* k *s.t.* $a_k \neq 0$ *is achievable where*

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

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.
- For example, **Ordentlich-Erez-Nazer '14** approximates the sum capacity of the symmetric Gaussian interference channel via decoding two linear combinations.

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.
- For example, **Ordentlich-Erez-Nazer '14** approximates the sum capacity of the symmetric Gaussian interference channel via decoding two linear combinations.
- **Ordentlich-Erez-Nazer '13** improves upon compute-and-forward for two or more linear combinations via successive cancellation.

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.
- For example, **Ordentlich-Erez-Nazer '14** approximates the sum capacity of the symmetric Gaussian interference channel via decoding two linear combinations.
- **Ordentlich-Erez-Nazer '13** improves upon compute-and-forward for two or more linear combinations via successive cancellation.
- What about jointly decoding the linear combinations?

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.
- For example, **Ordentlich-Erez-Nazer '14** approximates the sum capacity of the symmetric Gaussian interference channel via decoding two linear combinations.
- **Ordentlich-Erez-Nazer '13** improves upon compute-and-forward for two or more linear combinations via successive cancellation.
- What about jointly decoding the linear combinations?
- **Ordentlich-Erez '13** derived bounds for lattice-based codes.

Beyond One Linear Combination

- In some scenarios, it is of interest to decode two or more linear combinations at each receiver.
- For example, **Ordentlich-Erez-Nazer '14** approximates the sum capacity of the symmetric Gaussian interference channel via decoding two linear combinations.
- **Ordentlich-Erez-Nazer '13** improves upon compute-and-forward for two or more linear combinations via successive cancellation.
- What about jointly decoding the linear combinations?
- **Ordentlich-Erez '13** derived bounds for lattice-based codes.
- **This talk:** We can analyze this via joint typicality decoding to get an achievable rate region.

Jointly Decoding Two Linear Combinations of K Codewords

- At node $k \in [1 : K]$, the message M_k is encoded using the nested linear coding architecture.

Jointly Decoding Two Linear Combinations of K Codewords

- At node $k \in [1:K]$, the message M_k is encoded using the nested linear coding architecture.
- Let L_k be the chosen index from the multicoding step.

Jointly Decoding Two Linear Combinations of K Codewords

- At node $k \in [1 : K]$, the message M_k is encoded using the nested linear coding architecture.
- Let L_k be the chosen index from the multicoding step.
- The objective of the receiver is to compute two linear combinations of the codewords,

$$W_1^n(T_1) = \bigoplus_{k=1}^{K} a_{1k} u_k^n(M_k, L_k)$$
$$W_2^n(T_2) = \bigoplus_{k=1}^{K} a_{2k} u_k^n(M_k, L_k) \ ,$$

with vanishing probability of error.

Jointly Decoding Two Linear Combinations of K Codewords

- At node $k \in [1:K]$, the message M_k is encoded using the nested linear coding architecture.
- Let L_k be the chosen index from the multicoding step.
- The objective of the receiver is to compute two linear combinations of the codewords,

$$W_1^n(T_1) = \bigoplus_{k=1}^{K} a_{1k} u_k^n(M_k, L_k)$$
$$W_2^n(T_2) = \bigoplus_{k=1}^{K} a_{2k} u_k^n(M_k, L_k) \ ,$$

  with vanishing probability of error.
- **Key Technical Issue:** Random linear codewords are pairwise independent, but not 4-wise independent!

Jointly Decoding Two Linear Combinations of K Codewords

Theorem (Lim-Chen-Nazer-Gastpar Allerton '15)

A rate tuple $(R_1, \ldots, R_K)$ is achievable for computing two linear combinations if

$$
\begin{aligned}
R_k &< \min\{H(U_k) - H(V|Y), H(U_k) - H(W_1, W_2|Y, V)\}, \quad k \in \mathcal{K}_1 \\
R_j &< I(W_2; Y, W_1) - H(W_2) + H(U_j), \quad j \in \mathcal{K}_2, \\
R_k + R_j &< I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\end{aligned}
$$

$$or$$

$$
\begin{aligned}
R_k &< I(W_1; Y, W_2) - H(W_1) + H(U_k), \quad k \in \mathcal{K}_1, \\
R_j &< \min\{H(U_j) - H(V|Y), H(U_j) - H(W_1, W_2|Y, V)\}, \quad j \in \mathcal{K}_2, \\
R_k + R_j &< I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\end{aligned}
$$

for some $\prod_{k=1}^{K} p(u_k)$ and $x_k(u_k)$ and non-zero vector $\boldsymbol{b} \in \mathbb{F}_{\mathsf{q}}^2$,
where $\mathcal{K}_j = \{k \in [1:K] : a_{jk} \neq 0\}$, $j = 1, 2$
and $V = b_1 W_1 \oplus b_2 W_2$.

Jointly Decoding Two Linear Combinations of K Codewords

Theorem (Lim-Chen-Nazer-Gastpar Allerton '15)

A rate tuple $(R_1, \dots, R_K)$ is achievable for computing two linear combinations if

$$
\begin{aligned}
R_k &< \min\{H(U_k) - H(V|Y), H(U_k) - H(W_1, W_2|Y, V)\}, \quad k \in \mathcal{K}_1 \\
R_j &< I(W_2; Y, W_1) - H(W_2) + H(U_j), \quad j \in \mathcal{K}_2, \\
R_k + R_j &< I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\end{aligned}
$$

or

$$
\begin{aligned}
R_k &< I(W_1; Y, W_2) - H(W_1) + H(U_k), \quad k \in \mathcal{K}_1, \\
R_j &< \min\{H(U_j) - H(V|Y), H(U_j) - H(W_1, W_2|Y, V)\}, \quad j \in \mathcal{K}_2, \\
R_k + R_j &< I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\end{aligned}
$$

for some $\prod_{k=1}^{K} p(u_k)$ and $x_k(u_k)$ and non-zero vector $\boldsymbol{b} \in \mathbb{F}_{\mathsf{q}}^2$, where $\mathcal{K}_j = \{k \in [1 : K] : a_{jk} \neq 0\}$, $j = 1, 2$ and $V = b_1 W_1 \oplus b_2 W_2$.

- The auxiliary linear combination V plays a key role in classifying dependent competing pairs in the error analysis.

Theorem (Lim-Chen-Nazer-Gastpar Allerton '15)

A rate pair (R_1, R_2) *is achievable for the discrete memoryless multiple-access channel if*

$$R_1 < \max_{\mathbf{a} \neq \mathbf{0}} \min\{H(U_1) - H(W|Y),\ H(U_1) - H(U_1, U_2|Y, W)\},$$

$$R_2 < I(X_2; Y|X_1),$$

$$R_1 + R_2 < I(X_1, X_2; Y),$$

or

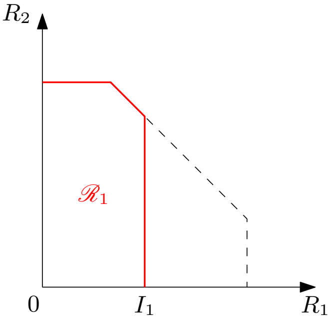
$$R_1 < I(X_1; Y|X_2),$$

$$R_2 < \max_{\mathbf{a} \neq \mathbf{0}} \min\{H(U_2) - H(W|Y),\ H(U_2) - H(U_1, U_2|Y, W)\},$$

$$R_1 + R_2 < I(X_1, X_2; Y)$$

for some $p(u_1)p(u_2)$ *and* $x_1(u_1)$, $x_2(u_2)$, *where* $W = a_1 U_1 \oplus a_2 U_2$.

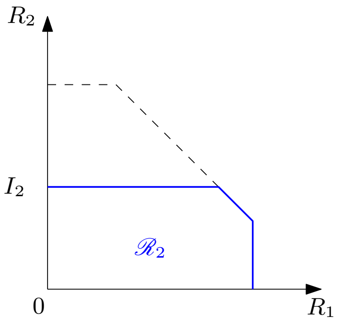
Multiple-Access Rate Region



$$
\begin{aligned}
R_1 &< I_1, \\
R_2 &< I(X_2; Y | X_1), \\
R_1 + R_2 &< I(X_1, X_2; Y),
\end{aligned}
$$

where $I_1 = \max_{\mathbf{a} \neq \mathbf{0}} \min\{H(U_1) - H(W|Y),\ H(U_1) - H(U_1, U_2|Y, W)\}$

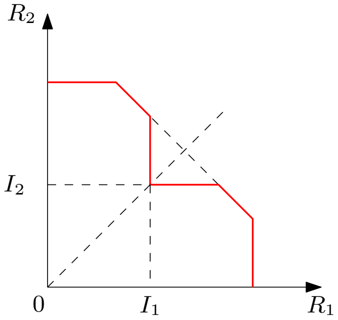
Multiple-Access Rate Region



$$R_1 < I(X_1; Y | X_2),$$
$$R_2 < I_2,$$
$$R_1 + R_2 < I(X_1, X_2; Y),$$

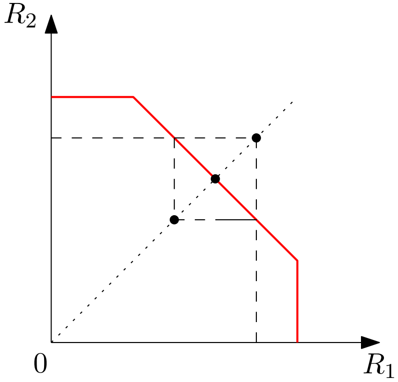
where $I_2 = \max_{\mathbf{a} \neq \mathbf{0}} \min\{H(U_2) - H(W|Y),\ H(U_2) - H(U_1, U_2|Y, W)\}$

Multiple-Access Rate Region



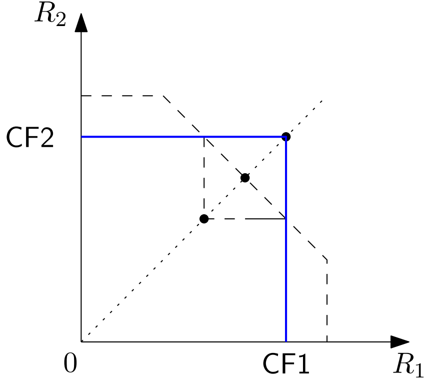
- Multiple-access rate region via nested linear codes:

$$\mathscr{R}_1 \cup \mathscr{R}_2$$



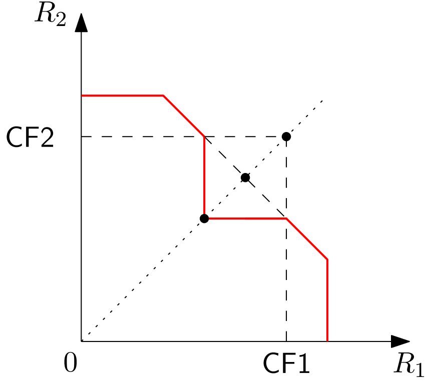
MAC Capacity Region

- Even if the receiver is only interested in recovering one linear combination it can sometimes help to decode two!



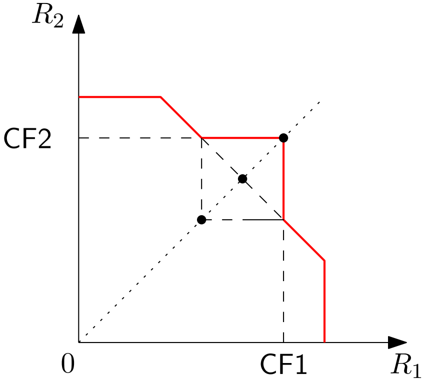
Decode One Linear Combination

- Even if the receiver is only interested in recovering one linear combination it can sometimes help to decode two!



Multiple-Access via Nested Linear Codes

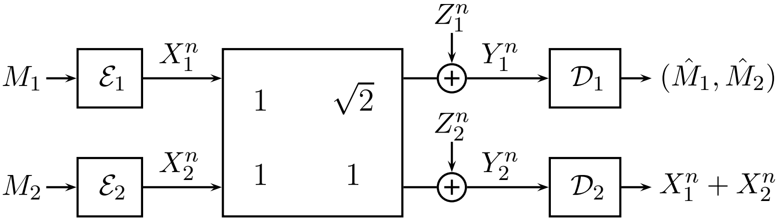
- Even if the receiver is only interested in recovering one linear combination it can sometimes help to decode two!



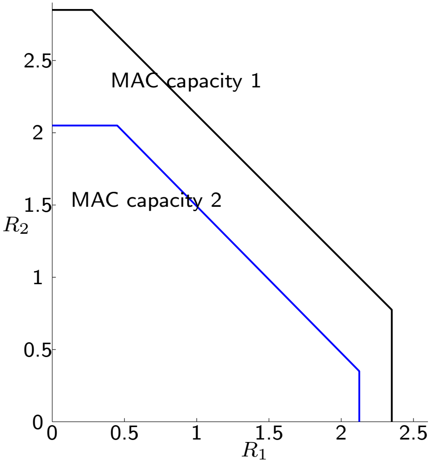
Union

- Even if the receiver is only interested in recovering one linear combination it can sometimes help to decode two!

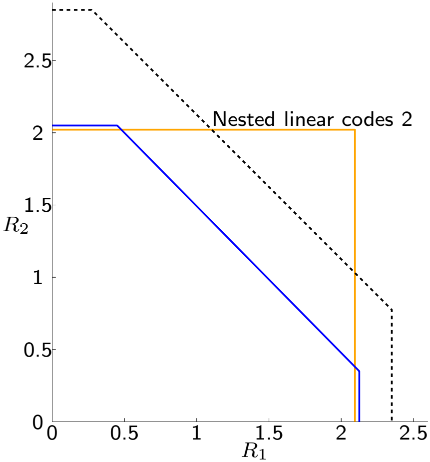
Case Study: Two-Sender, Two-Receiver Network



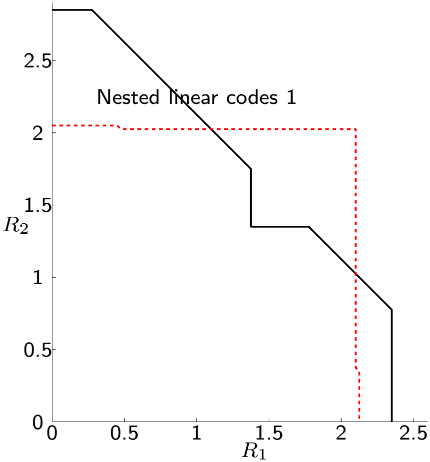
Case Study: Two-Sender, Two-Receiver Network



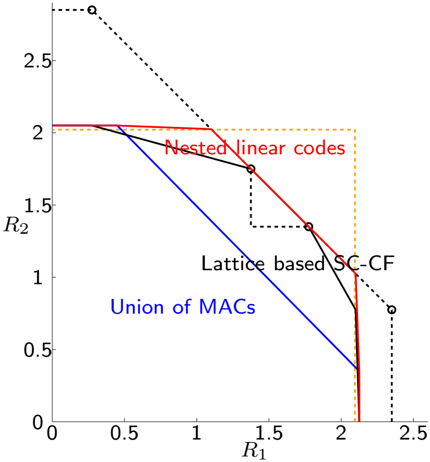
Case Study: Two-Sender, Two-Receiver Network



Case Study: Two-Sender, Two-Receiver Network



Case Study: Two-Sender, Two-Receiver Network



Concluding Remarks

- First steps towards bringing algebraic network information theory back into the realm of joint typicality.
- Joint decoding rate region for compute-and-forward that outperforms parallel and successive decoding.