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.
**Compute-and-Forward**

**Goal:** Send a **linear combination** of the messages to the receiver.
Compute-and-Forward

**Goal:** Send a linear combination of the messages to the receiver.
**Compute-and-Forward**

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

\[ m_1 \rightarrow \mathcal{E}_1 \rightarrow X_1^n \rightarrow \text{Channel} \rightarrow D \rightarrow \hat{t} \]

\[ m_2 \rightarrow \mathcal{E}_2 \rightarrow X_2^n \rightarrow \text{Channel} \rightarrow D \rightarrow \hat{t} \]

\[ \vdots \]

\[ m_K \rightarrow \mathcal{E}_K \rightarrow X_K^n \rightarrow \text{Channel} \rightarrow D \rightarrow \hat{t} \]

\[ \nu(\cdot) = \text{q-ary expansion} \]

\[ \nu(t) = \bigoplus_{k=1}^{K} a_k \nu(m_k) \]

\[ F_q^k \]
**Compute-and-Forward**

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

\[ \begin{align*}
    m_1 & \rightarrow E_1 & X_1^n & \rightarrow Y_1^n & \rightarrow \hat{t}_1 \\
    m_2 & \rightarrow E_2 & X_2^n & \rightarrow Y_2^n & \rightarrow \hat{t}_2 \\
    \vdots & \vdots & \vdots & \vdots & \vdots \\
    m_K & \rightarrow E_K & X_K^n & \rightarrow Y_K^n & \rightarrow \hat{t}_K
\end{align*} \]

**Channel**

\[ \nu(\cdot) = \text{q-ary expansion} \]

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

[\mathbb{F}_q^K]
**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).

\[
\begin{align*}
\nu(\cdot) &= \text{q-ary expansion} \\
\nu(t_\ell) &= \bigoplus_{k=1}^{K} a_{\ell,k} \nu(m_k)
\end{align*}
\]
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).

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

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

\[ \nu(\cdot) = q\text{-ary expansion} \]

\[ \nu = F^{'k}_{q} \]
The Usual Approach
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints:
  \[ \mathbb{E}\|x_\ell\|^2 \leq nP. \]
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints: $\mathbb{E}\|x_\ell\|^2 \leq nP$.
- Use nested lattice codes.
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints: \( \mathbb{E}\|x_\ell\|^2 \leq nP. \)
- Use nested lattice codes.

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

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

\[ m_1 \rightarrow \mathcal{E}_1 \rightarrow X_1^n \]
\[ m_2 \rightarrow \mathcal{E}_2 \rightarrow X_2^n \]
\[ \vdots \]
\[ m_K \rightarrow \mathcal{E}_K \rightarrow X_K^n \]
\[ Z^n \rightarrow + \rightarrow Y^n \rightarrow \mathcal{D} \rightarrow \hat{t} \]

\[ \mathbf{v}(t) = \bigoplus_{k=1}^{K} \mathbf{v}(m_k) \]
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints:
  \[ \mathbb{E}\|x_\ell\|^2 \leq nP. \]
- Use nested lattice codes.

- Wilson-Narayanan-Pfister-Sprintson '10, Nazer-Gastpar '11:
  Decoding is successful if the rates satisfy
  \[ R_k < \frac{1}{2} \log^+ \left( \frac{1}{2} + P \right). \]

- Cut-set upper bound is \[ \frac{1}{2} \log(1 + P). \]
Computation over Gaussian MACs

- Symmetric Gaussian MAC.
- Equal power constraints: \( \mathbb{E} \| x_\ell \|^2 \leq nP \).
- Use nested lattice codes.

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

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

- 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?
Computation over Gaussian MACs

- How about general Gaussian MACs?

- Model using unequal power constraints: 
  \[ \mathbb{E} \| \mathbf{x}_\ell \|^2 \leq nP_\ell. \]

\[ \nu(t) = \bigoplus_{k=1}^{K} \left[ 0 \ \nu(m_k) \right] \]
Computation over Gaussian MACs

- How about general Gaussian MACs?

- Model using unequal power constraints:
  \[ \mathbb{E}\|x_\ell\|^2 \leq nP_\ell. \]

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

- How about general Gaussian MACs?
- Model using unequal power constraints:
  \[ \mathbb{E}\|x_\ell\|^2 \leq nP_\ell. \]
- 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:
  \[ \mathbb{E}\|x_{\ell}\|^2 \leq nP_\ell. \]

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

- \textbf{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

\[ M \rightarrow \text{Encoder} \xrightarrow{X^n} p_{Y|X} \xrightarrow{Y^n} \text{Decoder} \rightarrow \hat{M} \]

- 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. Codebooks

- Codewords are independent of one another.
- Can directly target an input distribution $p_X(x)$. 
Code Construction:
**Code Construction:**
- Pick a finite field $\mathbb{F}_q$ and a symbol mapping $x : \mathbb{F}_q \rightarrow \mathcal{X}$. 
Code Construction:
- Pick a finite field $\mathbb{F}_q$ and a symbol mapping $x : \mathbb{F}_q \to \mathcal{X}$.
- Set $\kappa = nR/\log(q)$. 
Code Construction:

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

**Code Construction:**

- Pick a finite field $\mathbb{F}_q$ and a symbol mapping $x : \mathbb{F}_q \rightarrow \mathcal{X}$.
- Set $\kappa = nR / \log(q)$.
- Draw a random generator matrix $G \in \mathbb{F}_q^{\kappa \times n}$ elementwise i.i.d. $\text{Unif}(\mathbb{F}_q)$. Let $G$ be a realization.
- Draw a random shift (or “dither”) $D^n$ elementwise i.i.d. $\text{Unif}(\mathbb{F}_q)$. Let $d^n$ be a realization.
**Point-to-Point Channels: Linear Codes**

\[
\begin{array}{cccccc}
M & \rightarrow & \text{Linear Code} & \rightarrow & U^n & \rightarrow & x(u) & \rightarrow & X^n & \rightarrow & Y^n & \rightarrow & \text{Decoder} & \rightarrow & \hat{M} \\
\end{array}
\]

**Encoder**

**Code Construction:**

- Pick a finite field \( \mathbb{F}_q \) and a symbol mapping \( x : \mathbb{F}_q \rightarrow \mathcal{X} \).
- Set \( \kappa = nR/ \log(q) \).
- Draw a random generator matrix \( G \in \mathbb{F}_q^{\kappa \times n} \) elementwise i.i.d. \( \text{Unif}(\mathbb{F}_q) \). Let \( G \) be a realization.
- Draw a random shift (or “dither”) \( D^n \) elementwise i.i.d. \( \text{Unif}(\mathbb{F}_q) \). Let \( d^n \) be a realization.
- Take \( q \)-ary expansion of message \( m \) into the vector \( \nu(m) \in \mathbb{F}_q^\kappa \).
**Code Construction:**

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

---

Point-to-Point Channels: Linear Codes
**Point-to-Point Channels: Linear Codes**

![Diagram of point-to-point channels with linear codes]

**Code Construction:**

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

---

$M \xrightarrow{\text{Linear Code}} U^n \xrightarrow{x(u)} X^n \xrightarrow{p_{Y|X}} Y^n \xrightarrow{\text{Decoder}} \hat{M}$
Random i.i.d. Codebooks

- Codewords are pairwise independent of one another.
- Codewords are uniformly distributed over $\mathbb{F}_q^n$. 
Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, Ahlswede ’71.
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 $q \gg X$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\text{Unif}(\mathbb{F}_q)$. This can attain the capacity.
• 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.
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 $q \gg X$ 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.
• 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 $q \gg X$ 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.
Point-to-Point Channels: Linear Codes + 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 $\kappa = n(R + \hat{R})/\log(q)$. 
Point-to-Point Channels: Linear Codes + Multicoding

Code Construction:
- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R}) / \log(q)$.
- 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 \( \kappa = n(R + \hat{R})/\log(q) \).
- Pick generator matrix \( G \) and dither \( d^n \) as before.
- Take q-ary expansions \([\vec{v}(m) \, \vec{v}(l)] \in F_q^\kappa\).
**Code Construction:**

- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(q)$.
- Pick generator matrix $G$ and dither $d^n$ as before.
- Take q-ary expansions $[\bm{v}(m) \; \bm{v}(l)] \in \mathbb{F}_q^\kappa$.
- **Linear codewords:** $u^n(m, l) = [\bm{v}(m) \; \bm{v}(l)] G \oplus d^n$. 
Point-to-Point Channels: Linear Codes + Multicoding

Encoding:
Encoding:
- Fix $p(u)$ and $x(u)$. 
**Encoding:**

- **Fix** $p(u)$ and $x(u)$.
- **Multicoding:** For each $m$, find an index $l$ such that $u^n(m, l) \in T_{\epsilon'}^{(n)}(U)$
Point-to-Point Channels: Linear Codes + Multicoding

Encoding:
- Fix $p(u)$ and $x(u)$.
- Multicoding: For each $m$, find an index $l$ such that $u^n(m, l) \in \mathcal{T}^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \parallel p_q)$ (where $p_q$ is uniform over $\mathbb{F}_q$).
Encoding:

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

- **Multicoding**: For each $m$, find an index $l$ such that $u^n(m, l) \in T_e^n(U)$

- Succeeds w.h.p. if $\hat{R} > D(p_U \parallel p_q)$ (where $p_q$ is uniform over $\mathbb{F}_q$).

- Transmit $x_i = x(u_i(m, l))$. 
Encoding:
- Fix $p(u)$ and $x(u)$.
- **Multicoding**: For each $m$, find an index $l$ such that $u^n(m, l) \in T^{(n)}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_q)$ (where $p_q$ is uniform over $\mathbb{F}_q$).
- Transmit $x_i = x(u_i(m, l))$.

Decoding:
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 $\hat{R} > D(p_U \parallel p_q)$ (where $p_q$ is uniform over $\mathbb{F}_q$).
- Transmit $x_i = x(u_i(m, l))$.

Decoding:
- **Joint Typicality Decoding**: Find the unique index $\hat{m}$ such that $(u^n(\hat{m}, \hat{l}), y^n) \in \mathcal{T}_\epsilon^{(n)}(U, Y)$ for some index $\hat{l}$. 
Encoding:
- Fix $p(u)$ and $x(u)$.
- Multicoding: For each $m$, find an index $l$ such that $u^n(m, l) \in T^{(n)}_{\epsilon}(U)$
- Succeeds w.h.p. if $\hat{R} > D(p_U \| p_q)$ (where $p_q$ is uniform over $\mathbb{F}_q$).
- Transmit $x_i = x(u_i(m, l))$.

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

- This is the basic coding framework that we will use for each transmitter.
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 $q \geq |X|$. 

- 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 $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$. 
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 $\kappa = n(\max\{R_1 + \hat{R}_1, R_2 + \hat{R}_2\}) / \log(q)$.
**Code Construction:**

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{n\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1, R_2 + \hat{R}_2\})/\log(q)$.
- Pick generator matrix $G$ and dithers $d_{1m}, d_{2m}$ as before.
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 $\kappa = n(\max\{R_1 + \hat{R}_1, R_2 + \hat{R}_2\})/\log(q)$.
- Pick generator matrix $G$ and dithers $d_{1}^{m}, d_{2}^{m}$ as before.
- Take q-ary expansions $[\nu(m_1) \nu(l_1)] \in \mathbb{F}_q^{\kappa}$
- $[\nu(m_2) \nu(l_2) \mathbf{0}] \in \mathbb{F}_q^{\kappa}$ Zero-padding
Nested Linear Coding Architecture

Code Construction:

- Messages $m_k \in [2^{nR_k}]$ and auxiliary indices $l_k \in [2^{\hat{R}_k}]$, $k = 1, 2$.
- Set $\kappa = n(\max\{R_1 + \hat{R}_1, \ R_2 + \hat{R}_2\}) / \log(q)$.
- Pick generator matrix $G$ and dithers $d_{1n}^m, d_{2n}^m$ as before.
- Take q-ary expansions $[\eta(m_1, l_1)] \in \mathbb{F}_q^\kappa$
  $[\eta(m_2, l_2)] \in \mathbb{F}_q^\kappa$
Nested Linear Coding Architecture

\[ M_1 \xrightarrow{\text{Linear Code}} \xrightarrow{\text{Multi-coding}} X_1^n \]
\[ M_2 \xrightarrow{\text{Linear Code}} \xrightarrow{\text{Multi-coding}} X_2^n \]
\[ Y^n \rightarrow \text{Decoder} \rightarrow \hat{T} \]

**Code Construction:**

- Messages \( m_k \in [2^{nR_k}] \) and auxiliary indices \( l_k \in [2^{n\hat{R}_k}], \ k = 1, 2. \)
- Set \( \kappa = n(\max\{R_1 + \hat{R}_1, \ R_2 + \hat{R}_2\})/\log(q). \)
- Pick generator matrix \( G \) and dithers \( d_{1}^n, d_{2}^n \) as before.
- Take q-ary expansions 
  \[
  [\eta(m_1, l_1)] \in \mathbb{F}_q^\kappa \\
  [\eta(m_2, l_2)] \in \mathbb{F}_q^\kappa
  \]
- Linear codewords:
  \[
  u_1^n(m_1, l_1) = \eta(m_1, l_1)G \oplus d_1^n \\
  u_2^n(m_2, l_2) = \eta(m_2, l_2)G \oplus d_2^n
  \]
Nested Linear Coding Architecture

$M_1 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow U_1^n \rightarrow x_1(u_1) \rightarrow X_1^n \rightarrow p_{Y|X_1X_2} \rightarrow Y^n \rightarrow \text{Decoder} \rightarrow \hat{T}$

$M_2 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow U_2^n \rightarrow x_2(u_2) \rightarrow X_2^n \rightarrow \text{Decoder} \rightarrow \hat{T}$

Encoding:
Nested Linear Coding Architecture

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

```
M_1 \rightarrow [Linear Code] \rightarrow [Multi-coding] \rightarrow X_1^n \rightarrow p_{Y|X_1X_2} \rightarrow Y^n \rightarrow \text{Decoder} \rightarrow \hat{T}
```

```
M_2 \rightarrow [Linear Code] \rightarrow [Multi-coding] \rightarrow X_2^n
```
Nested Linear Coding Architecture

Encoding:

- **Fix** $p(u_1), p(u_2), x_1(u_1), \text{ and } x_2(u_2)$.

- **Multicoding**: For each $m_k$, find an index $l_k$ such that $w^n_k(m_k, l_k) \in \mathcal{T}^{(n)}_{c_i}(U_k)$. 
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 $u_k^n(m_k, l_k) \in \mathcal{T}^{(n)}_{ε}(U_k)$.
- Succeeds w.h.p. if $\hat{R}_k > D(p_{U_k} \| p_q)$. 

\[ M_1 \xrightarrow{\text{Linear Code}} \xrightarrow{\text{Multicoding}} x_1(u_1) \xrightarrow{X_1^n} Y^n \xrightarrow{\text{Decoder}} \hat{T} \]

\[ M_2 \xrightarrow{\text{Linear Code}} \xrightarrow{\text{Multicoding}} x_2(u_2) \xrightarrow{X_2^n} \]
**Nested Linear Coding Architecture**

\[ M_1 \xrightarrow{\text{Linear Code}} \text{Multi-coding} \xrightarrow{U_1^n} x_1(u_1) \]

\[ M_2 \xrightarrow{\text{Linear Code}} \text{Multi-coding} \xrightarrow{U_2^n} x_2(u_2) \]

\[ p_{Y|X_1X_2} \xrightarrow{Y^n} \text{Decoder} \xrightarrow{\hat{T}} \]

**Encoding:**

- **Fix** \( p(u_1), p(u_2), x_1(u_1), \text{and} x_2(u_2). \)
- **Multicoding:** For each \( m_k \), find an index \( l_k \) such that \( u^n_k(m_k, l_k) \in T_{\epsilon}^{(n)}(U_k). \)
- **Succeeds w.h.p.** if \( \hat{\epsilon}_k > D(p_{U_k} \| p_q). \)
- **Transmit** \( x_{ki} = x_k(u_{ki}(m_k, l_k)). \)
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
  \[ u_k^n(m_k, l_k) \in \mathcal{T}_e^{(n)}(U_k). \]
- Succeeds w.h.p. if \( \hat{R}_k > D(p_{U_k} || p_q) \).
- Transmit \( x_{ki} = x_k(u_{ki}(m_k, l_k)) \).
Nested Linear Coding Architecture

\[ M_1 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow U_1^n \rightarrow x_1(u_1) \rightarrow X_1^n \rightarrow p_{Y|X_1X_2} \rightarrow Y^n \rightarrow \text{Decoder} \rightarrow \hat{T} \]

\[ M_2 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow U_2^n \rightarrow x_2(u_2) \rightarrow X_2^n \rightarrow p_{Y|X_1X_2} \rightarrow Y^n \rightarrow \text{Decoder} \rightarrow \hat{T} \]

Computation Problem:
Nested Linear Coding Architecture

Computation Problem:
- Consider the coefficients $\mathbf{a} \in \mathbb{F}_q^2$, $\mathbf{a} = [a_1, a_2]$
**Nested Linear Coding Architecture**

![Diagram of Nested Linear Coding Architecture]

**Computation Problem:**

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

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

- $M_1 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow x_1(u_1) \rightarrow X_1^n \rightarrow p_{Y|X_1X_2} \rightarrow Y^n \rightarrow \text{Decoder} \rightarrow \hat{T}$

- $M_2 \rightarrow \text{Linear Code} \rightarrow \text{Multi-coding} \rightarrow x_2(u_2) \rightarrow X_2^n$
Nested Linear Coding Architecture

Computation Problem:

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

$$W^n(T) = a_1 U^n_1(M_1, L_1) \oplus a_2 U^n_2(M_2, L_2)$$
**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:
  \[ W^n(T) = a_1 U^n_1(M_1, L_1) \oplus a_2 U^n_2(M_2, L_2) \]
- 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: $P_{\varepsilon}^{(n)} = P\{T \neq \hat{T}\}$
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: $P^{(n)} = P\{T \neq \hat{T}\}$

- A rate pair is achievable if there exists a sequence of codes such that $P^{(n)} \to 0$ as $n \to \infty$. 
Decoding:

- **Joint Typicality Decoding:** Find an index $t \in \left[2^n \max(R_1 + \hat{R}_1, R_2 + \hat{R}_2) \right]$ such that $(w^n(t), y^n) \in \mathcal{T}_\epsilon^{(n)}$. 
**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 \(\mathcal{U}_k = \mathbb{F}_q, k = 1, 2\), and \(W = a_1 U_1 \oplus a_2 U_2\).
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 $U_k = \mathbb{F}_q$, $k = 1, 2$, and $W = a_1 U_1 \oplus a_2 U_2$.

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

Nested Linear Coding Architecture
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: $P_{\epsilon}^{(n)} \leq \sum_{t \neq 0} P\{(W^n(t), Y^n) \in \mathcal{T}_\epsilon^{(n)} | \mathcal{M}\}$.
Proof Sketch

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

- 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

$$P(\mathcal{E}) = \sum_{t \neq 0} P\{ (W^n(t), Y^n) \in T^{(n)}_\epsilon, U^n_1(0, 0) \in T^{(n)}_\epsilon, U^n_2(0, 0) \in T^{(n)}_\epsilon | \mathcal{M} \}$$
Proof Sketch

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

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

- Conditioned on \( \mathcal{M} \), \( Y^n \rightarrow (U_1^n(0, 0), U_2^n(0, 0)) \rightarrow W^n(t) \)
Proof Sketch

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

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

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

$$R_k + \hat{R}_k + \hat{R}_1 + \hat{R}_2 < I(W; Y) + D(p_W || p_q) + D(p_{U_1} || p_q) + D(p_{U_2} || p_q)$$
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: \( Z \sim \mathcal{N}(0, 1) \)
• 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: \( \mathbb{E}[X_k^2] \leq P \)
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: \( \mathbb{E}[X_k^2] \leq P \)
- 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: \( \mathbb{E}[X_k^2] \leq P \)

- 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 \( R_k < R_{\text{comp}}(\mathbf{h}, \mathbf{a}) \) for \( k \) s.t. \( a_k \neq 0 \) is achievable where*

\[
R_{\text{comp}}(\mathbf{h}, \mathbf{a}) = \frac{1}{2} \log^+ \left( \frac{P}{\mathbf{a}^T (P^{-1} \mathbf{I} + \mathbf{h} \mathbf{h}^T)^{-1} \mathbf{a}} \right)
\]
Beyond One Linear Combination

- In some scenarios, it is of interest to decode **two or more linear combinations** at each receiver.
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.
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.
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?
• 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.
• 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.
At node $k \in [1 : K]$, the message $M_k$ is encoded using the nested linear coding architecture.
• 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,

$$W_1^n(T_1) = \bigoplus_{k=1}^{K} \, a_{1k} \, u^n_k(M_k, L_k)$$

$$W_2^n(T_2) = \bigoplus_{k=1}^{K} \, a_{2k} \, u^n_k(M_k, L_k) \; ,$$

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

\[
R_k < \min\{H(U_k) - H(V|Y), H(U_k) - H(W_1, W_2|Y, V)\}, \quad k \in \mathcal{K}_1
\]

\[
R_j < I(W_2; Y, W_1) - H(W_2) + H(U_j), \quad j \in \mathcal{K}_2,
\]

\[
R_k + R_j < I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\]

or

\[
R_k < I(W_1; Y, W_2) - H(W_1) + H(U_k), \quad k \in \mathcal{K}_1,
\]

\[
R_j < \min\{H(U_j) - H(V|Y), H(U_j) - H(W_1, W_2|Y, V)\}, \quad j \in \mathcal{K}_2,
\]

\[
R_k + R_j < I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\]

for some $\prod_{k=1}^K p(u_k)$ and $x_k(u_k)$ and non-zero vector $b \in \mathbb{F}_q^2$, where $\mathcal{K}_j = \{k \in [1: K] : a_{jk} \neq 0\}, j = 1, 2$ and $V = b_1W_1 \oplus b_2W_2$. 


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

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

\[
R_k < \min \{ H(U_k) - H(V|Y), H(U_k) - H(W_1, W_2|Y, V) \}, \quad k \in \mathcal{K}_1
\]

\[
R_j < I(W_2; Y, W_1) - H(W_2) + H(U_j), \quad j \in \mathcal{K}_2,
\]

\[
R_k + R_j < I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\]

\text{or}

\[
R_k < I(W_1; Y, W_2) - H(W_1) + H(U_k), \quad k \in \mathcal{K}_1,
\]

\[
R_j < \min \{ H(U_j) - H(V|Y), H(U_j) - H(W_1, W_2|Y, V) \}, \quad j \in \mathcal{K}_2,
\]

\[
R_k + R_j < I(W_1, W_2; Y) - H(W_1, W_2) + H(U_k) + H(U_j), \quad k \in \mathcal{K}_1, j \in \mathcal{K}_2
\]

for some \(\prod_{k=1}^{K} p(u_k)\) and \(x_k(u_k)\) and non-zero vector \(b \in \mathbb{F}^2_q\), where \(\mathcal{K}_j = \{ k \in [1 : K] : a_{j,k} \neq 0 \}, j = 1, 2\)

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

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

\[
R_2 < I(X_2; Y|X_1),
\]

\[
R_1 + R_2 < I(X_1, X_2; Y),
\]

or

\[
R_1 < I(X_1; Y|X_2),
\]

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

\[
R_1 + R_2 < I(X_1, X_2; Y)
\]

for some \(p(u_1)p(u_2)\) and \(x_1(u_1), x_2(u_2)\), where \(W = a_1 U_1 \oplus a_2 U_2\).
$R_1 < I_1,$

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

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

where $I_1 = \max_{a \neq 0} \min \{H(U_1) - H(W|Y), H(U_1) - H(U_1, U_2|Y, W)\}$
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 \( I_2 = \max_{a \neq 0} \min \{ H(U_2) - H(W | Y), H(U_2) - H(U_1, U_2 | Y, W) \} \)
• Multiple-access rate region via nested linear codes:

\[ \mathcal{R}_1 \cup \mathcal{R}_2 \]
Even if the receiver is only interested in recovering one linear combination it can sometimes help to decode two!
**“Two Help One”**

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

\[ M_1 \rightarrow \mathcal{E}_1 \xrightarrow{X_1^n} 1 \sqrt{2} \xrightarrow{1\ 1} Y_1^n \xrightarrow{\mathcal{D}_1} (\hat{M}_1, \hat{M}_2) \]

\[ M_2 \rightarrow \mathcal{E}_2 \xrightarrow{X_2^n} Z_1^n \xrightarrow{Y_2^n} Z_2^n \xrightarrow{\mathcal{D}_2} X_1^n + X_2^n \]
Case Study: Two-Sender, Two-Receiver Network

MAC capacity 1

MAC capacity 2

$R_1$ vs. $R_2$
Case Study: Two-Sender, Two-Receiver Network

Graph showing the performance of nested linear codes 2 in a two-sender, two-receiver network.
Case Study: Two-Sender, Two-Receiver Network

Nested linear codes 1
Case Study: Two-Sender, Two-Receiver Network

- Nested linear codes
- Lattice based SC-CF
- Union of MACs
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.