Towards an Algebraic Network Information Theory

Bobak Nazer (BU)

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

York Number Theory + Wireless Workshop
July 6, 2016

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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.

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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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...
- Guided the development and optimization of modern communication networks.

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:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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...
- Guided the development and optimization of modern communication networks.
- State-of-the-art elegantly captured in the recent textbook of **El Gamal and Kim.**

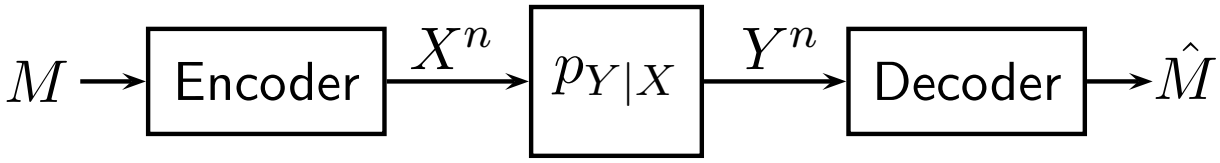
Network Information Theory

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.

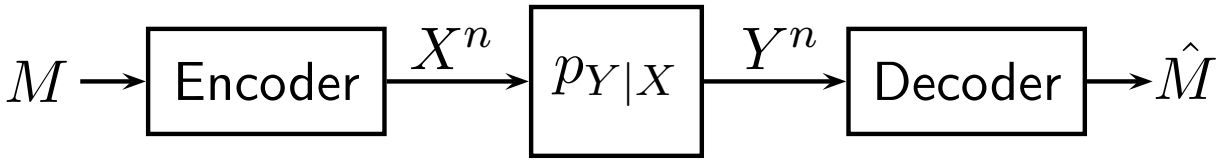
Classical Approach:

- Use average performance of random i.i.d. codebooks to argue good codebooks exist.
- 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...
- Guided the development and optimization of modern communication networks.
- 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 with low implementation complexity.

Point-to-Point Channels

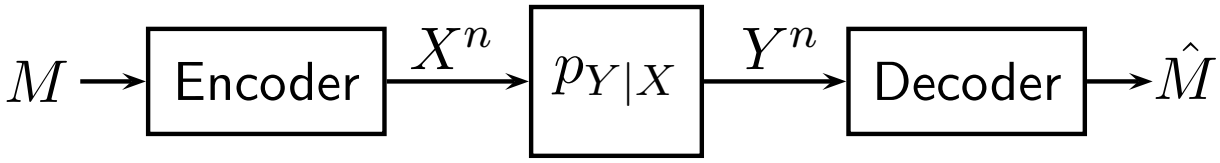


Point-to-Point Channels



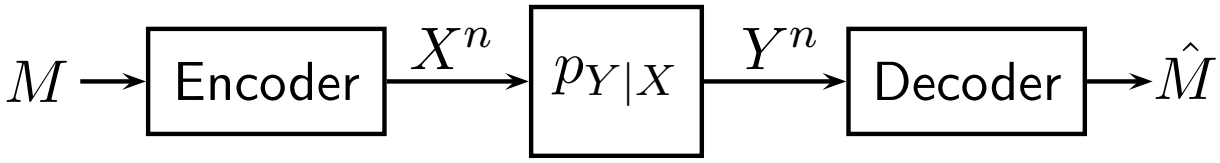
- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$

Point-to-Point Channels



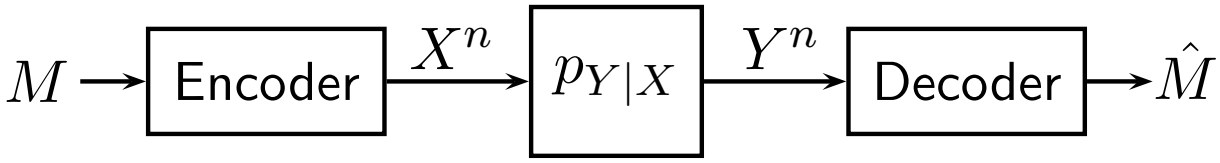
- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$

Point-to-Point Channels



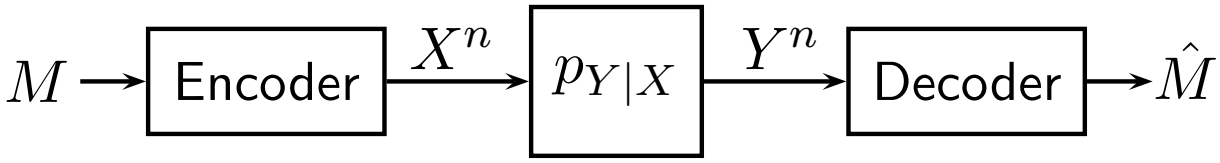
- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$

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 $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

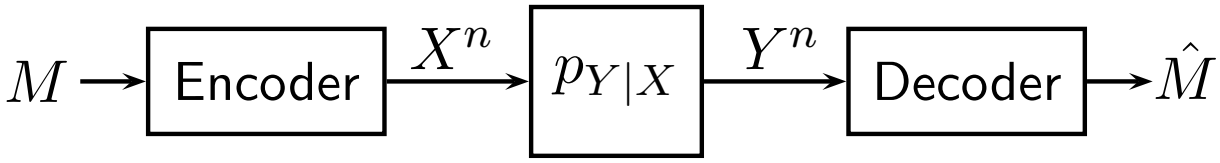
Point-to-Point Channels



- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- Decoder: a mapping $\hat{m}(y^n) \in [2^{nR}]$ for each $y^n \in \mathcal{Y}^n$

Theorem (Shannon '48)

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



- Messages: $m \in [2^{nR}] \triangleq \{0, \ldots, 2^{nR} - 1\}$
- Encoder: a mapping $x^n(m) \in \mathcal{X}^n$ for each $m \in [2^{nR}]$
- Memoryless Channel: $p_{Y^n|X^n}(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i)$
- 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.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.
- This motivates the typical set

$$\mathcal{T}_\epsilon^{(n)}(X) = \left\{x^n : |\pi(x|x^n) - p_X(x)| \le \epsilon p_X(x) \text{ for all } x \in \mathcal{X}\right\}$$

which satisfies $\lim_{n\to\infty} \mathsf{P}\left(X^n \in \mathcal{T}_\epsilon^{(n)}\right) = 1$.

Typicality

- x^n is a length-n sequence with elements from finite alphabet $\mathcal{X}$
- The empirical pmf (i.e., type) of x^n is

$$\pi(x|x^n) = \frac{1}{n}\big|\{i : x_i = x\}\big| \quad x \in \mathcal{X}$$

- If X^n is i.i.d. according to $p_X(x)$, then, by the weak law of large numbers, $\pi(x|x^n)$ converges to $p_X(x)$ in probability.
- This motivates the typical set

$$\mathcal{T}_\epsilon^{(n)}(X) = \left\{x^n : |\pi(x|x^n) - p_X(x)| \leq \epsilon p_X(x) \text{ for all } x \in \mathcal{X}\right\}$$

  which satisfies $\lim_{n\to\infty} \mathsf{P}\left(X^n \in \mathcal{T}_\epsilon^{(n)}\right) = 1$.
- We can generalize this definition to pairs of sequences (X^n, Y^n) that are i.i.d. according to $p_{XY}(x, y)$ and so on...

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x, y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x, y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\left\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X, Y)\right\} \leq 2^{-n(I(X;Y)-\delta(\epsilon))}$$

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x,y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X,Y)\big\} \leq 2^{-n(I(X;Y)-\delta(\epsilon))}$$

- For any $y^n \in \mathcal{T}_{\epsilon'}^{(n)}(Y)$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, y^n) \in \mathcal{T}_\epsilon^{(n)}(X,Y)\big\} \geq 2^{-n(I(X;Y)+\delta(\epsilon))} \ .$$

Joint Typicality Lemma

- Joint typicality is a powerful framework due to the availability of several key lemmas including

Joint Typicality Lemma

Select $p_{XY}(x,y)$ and $0 < \epsilon' < \epsilon$. Then, there exists $\delta(\epsilon)$ that tends to 0 as $\epsilon \to 0$ such that

- For any $\tilde{y}^n \in \mathcal{Y}^n$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, \tilde{y}^n) \in \mathcal{T}_\epsilon^{(n)}(X,Y)\big\} \leq 2^{-n(I(X;Y)-\delta(\epsilon))}$$

- For any $y^n \in \mathcal{T}_{\epsilon'}^{(n)}(Y)$ and $\tilde{X}^n$ i.i.d. $p_X(\tilde{x})$,

$$\mathsf{P}\big\{(\tilde{X}^n, y^n) \in \mathcal{T}_\epsilon^{(n)}(X,Y)\big\} \geq 2^{-n(I(X;Y)+\delta(\epsilon))} \ .$$

Intuition: Probability that i.i.d. $\tilde{X}^n$ looks jointly typical $\approx 2^{-nI(X;Y)}$

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.

Point-to-Point Capacity: Achievability Proof

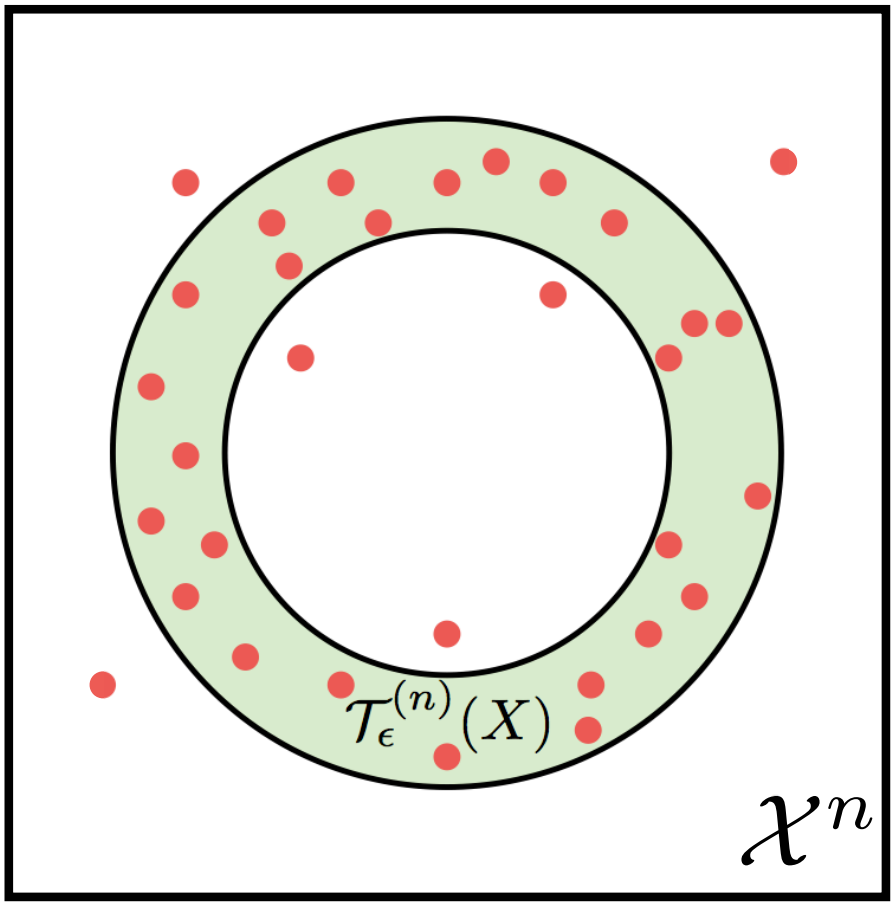
- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.
  - True codeword is not jointly typical, $(X^n(m), Y^n) \notin \mathcal{T}_\epsilon^{(n)}$. Probability goes to zero via WLLN.

Point-to-Point Capacity: Achievability Proof

- **Code Construction:** Generate 2^{nR} random codewords $X^n(1), \ldots, X^n(2^{nR})$ with each element drawn i.i.d. $p_X(x)$.
- **Encoding:** For message $m \in [2^{nR}]$, send codeword $X^n(m)$.
- **Decoding:** Search for $\hat{m}$ such that $(X^n(\hat{m}), Y^n)$ is jointly typical. If only one such $\hat{m}$, output it as the message estimate. Otherwise, declare an error.
- **Error Analysis:** Two possibilities.
  - True codeword is not jointly typical, $(X^n(m), Y^n) \notin \mathcal{T}_\epsilon^{(n)}$. Probability goes to zero via WLLN.
  - Some other codeword is jointly typical,

$$\begin{aligned} \mathsf{P}\left\{ \bigcup_{\tilde{m} \neq m} \left\{ (X^n(\tilde{m}), Y^n) \in \mathcal{T}_\epsilon^{(n)} \right\} \right\} &\leq \sum_{\tilde{m} \neq m} \mathsf{P}\left\{ (X^n(\tilde{m}), Y^n) \in \mathcal{T}_\epsilon^{(n)} \right\} \\ &\leq \sum_{\tilde{m} \neq m} 2^{-nI(X;Y)-\delta(\epsilon)} \\ &< 2^{nR}\, 2^{-nI(X;Y)-\delta(\epsilon)} . \end{aligned}$$

  Probability goes to zero if $R < I(X;Y) - \delta(\epsilon)$.



Random i.i.d. Codes

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

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?

Network Information 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:

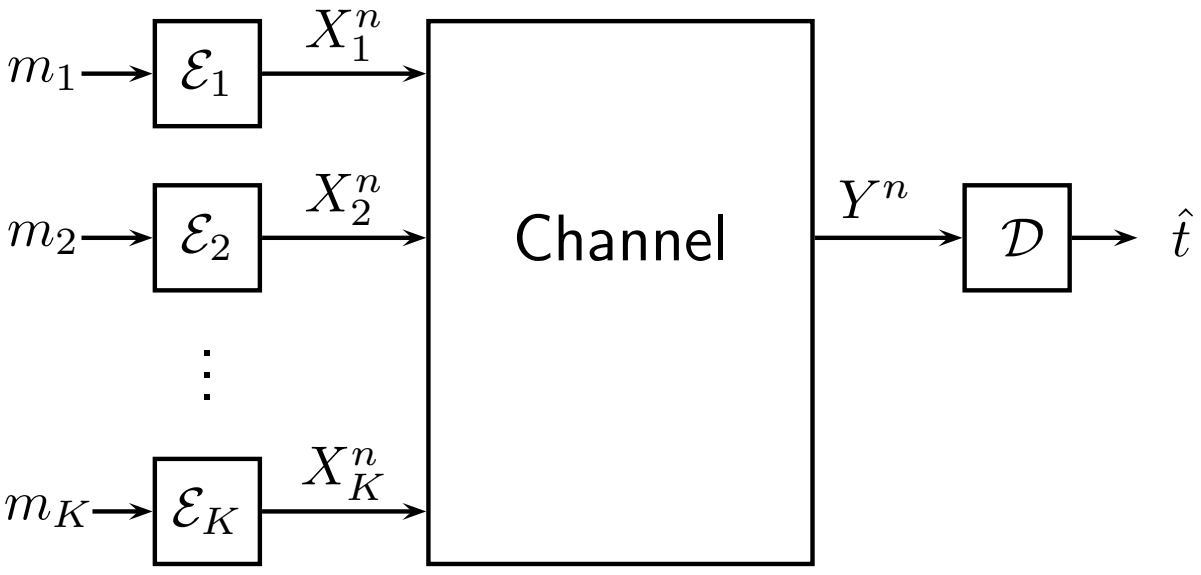
- Utilize linear or lattice codebooks.
- Compelling examples starting from the work of Körner and Marton on distributed compression and, more recently, many papers on physical-layer network coding, distributed dirty paper coding, and interference alignment.
- Coding schemes exhibit behavior not found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

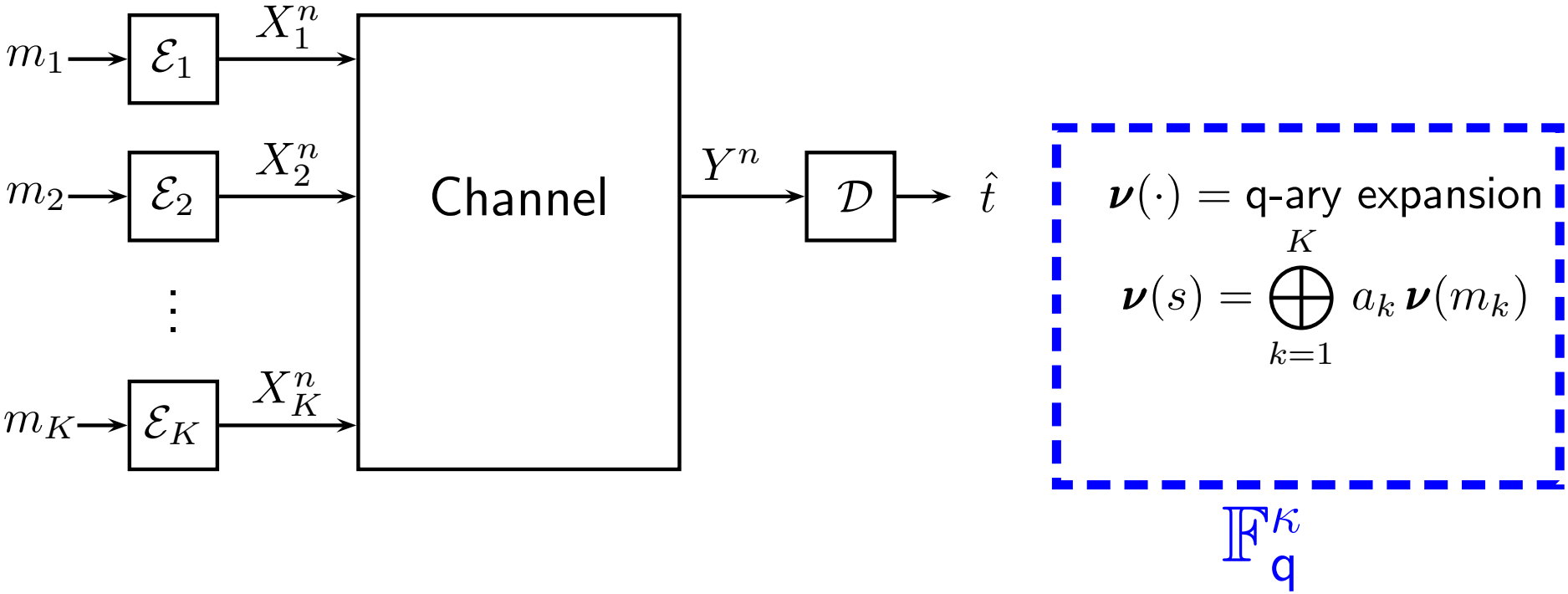
Compute-and-Forward

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

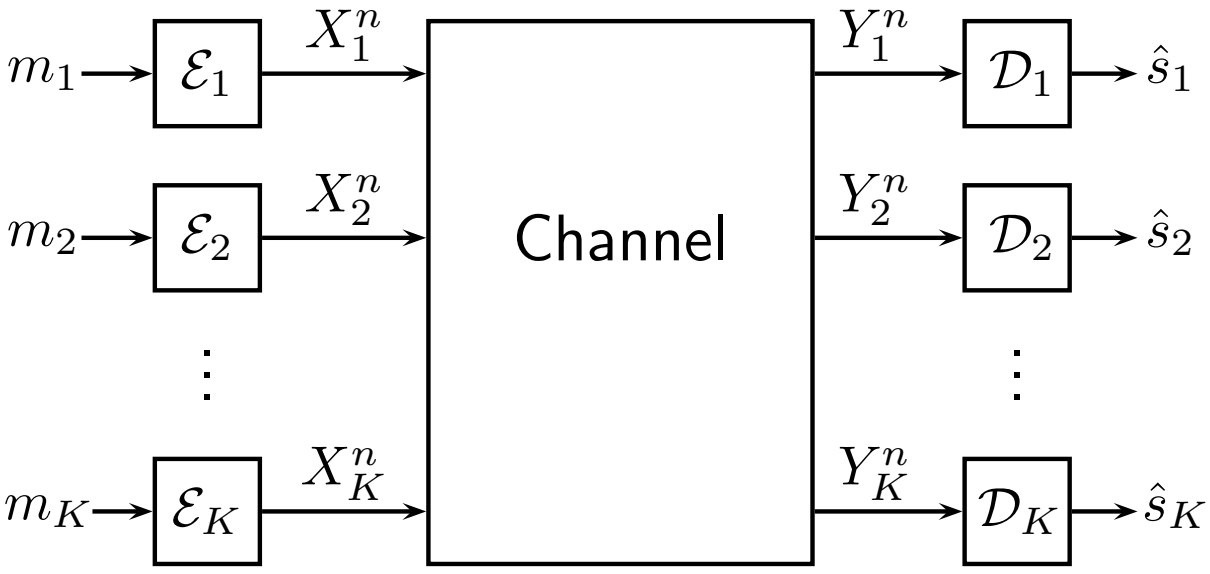
Goal: Send a linear combination of the messages to the receiver.



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



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



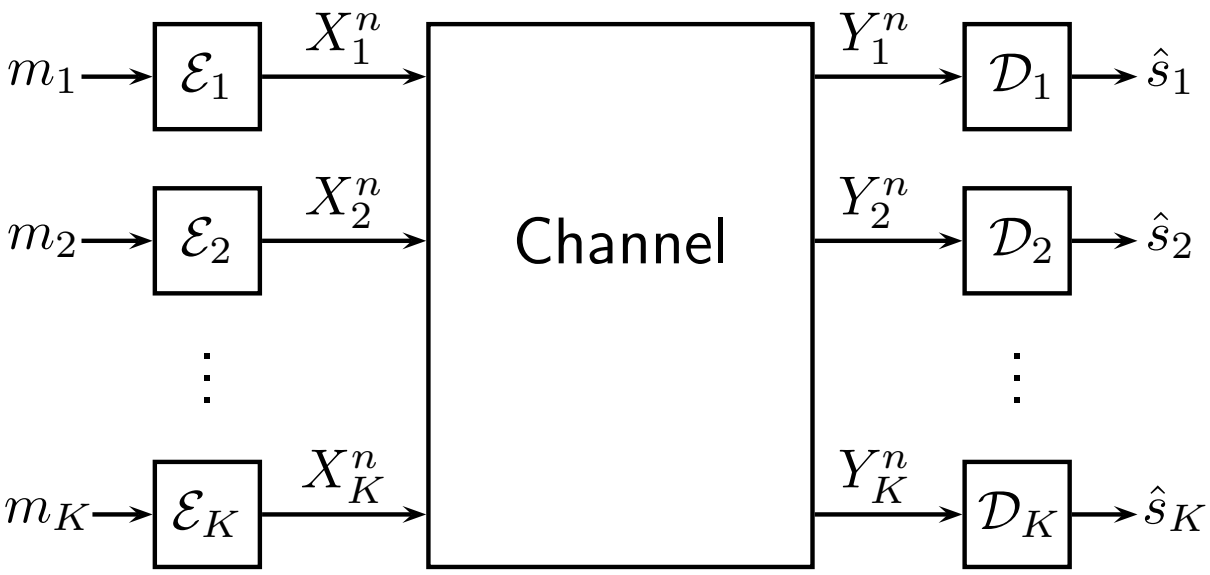
$\boldsymbol{\nu}(\cdot) =$ q-ary expansion

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

$\mathbb{F}_q^\kappa$

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).



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

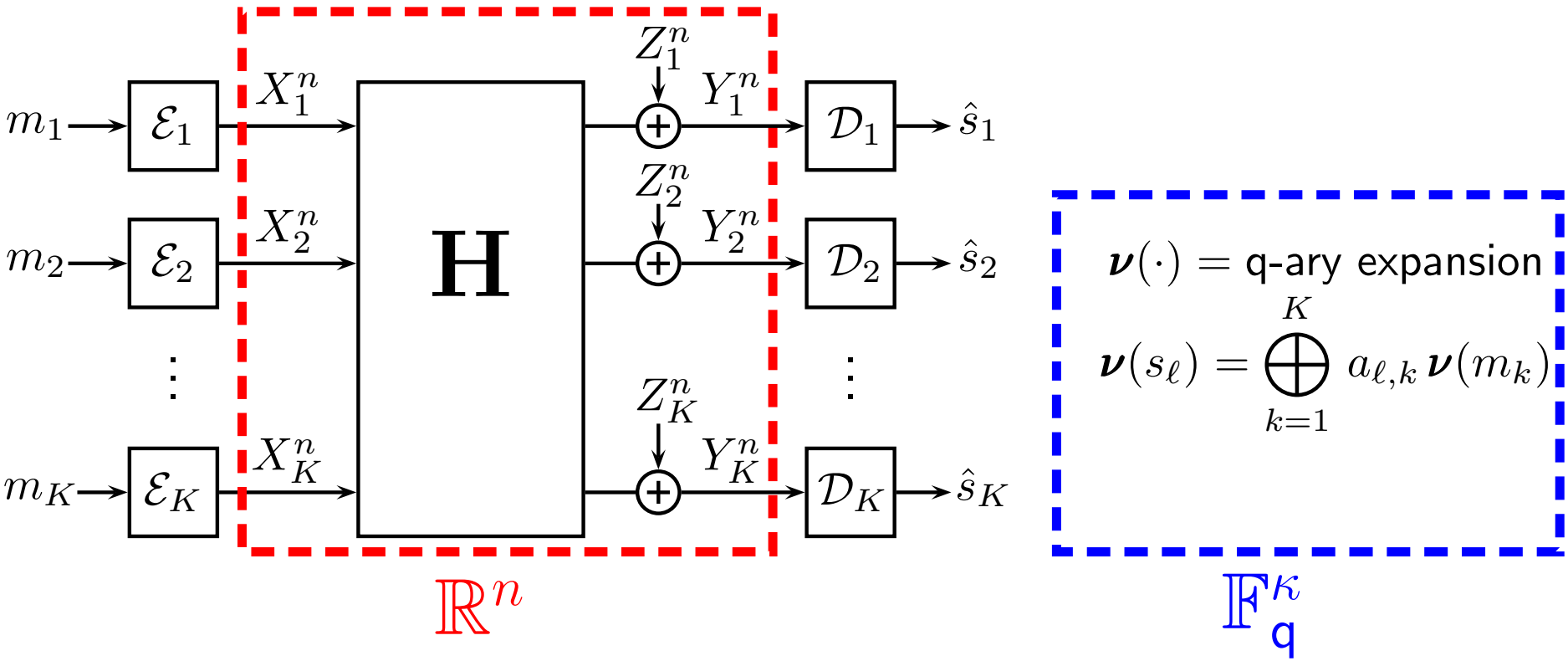
$$\boldsymbol{\nu}(s_\ell) = \bigoplus_{k=1}^{K} a_{\ell,k}\, \boldsymbol{\nu}(m_k)$$

$\mathbb{F}_q^\kappa$

Compute-and-Forward

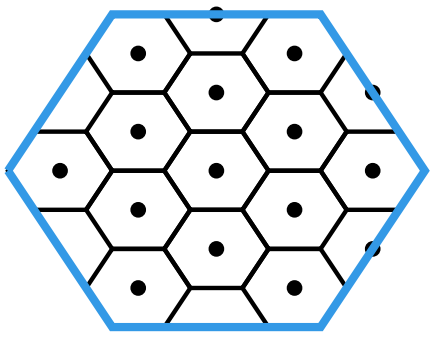
Goal: Send linear combinations of the messages to the receivers.

- Compute-and-forward can serve as a framework for communicating messages across a network (e.g., relaying, MIMO uplink/downlink, interference alignment).
- Much of the recent work has focused on Gaussian networks.



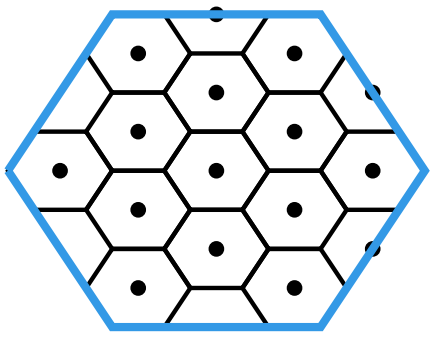
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.



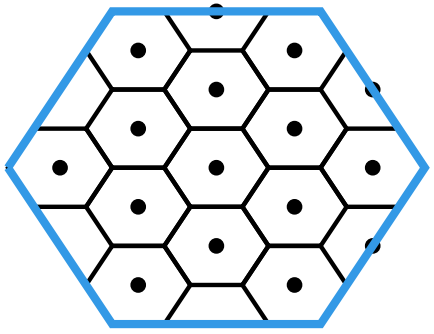
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.
- Fine lattice Λ_F protects against noise.



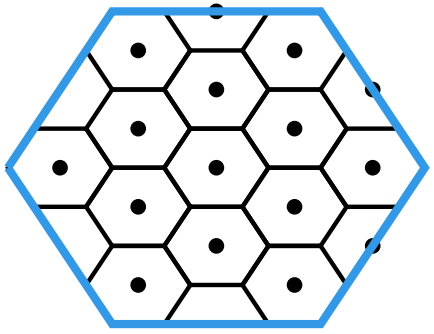
Nested Lattice Codes

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



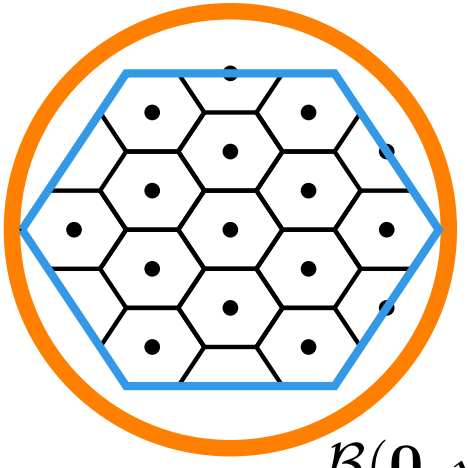
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.
- Fine lattice Λ_F protects against noise.
- Coarse lattice Λ_C enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.



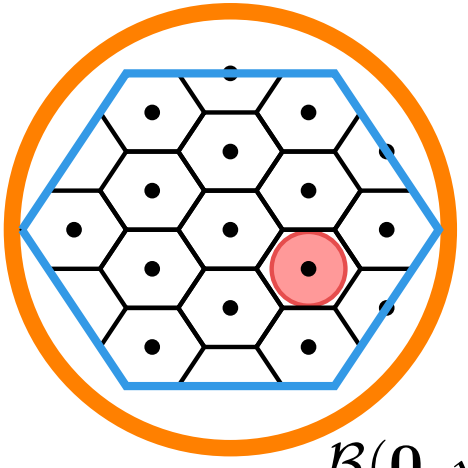
Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.
- Fine lattice Λ_F protects against noise.
- Coarse lattice Λ_C enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.



Nested Lattice Codes

- Nested Lattice Code: Formed by taking all elements of Λ_F that lie in the fundamental Voronoi region of Λ_C.
- Fine lattice Λ_F protects against noise.
- Coarse lattice Λ_C enforces the power constraint.
- Existence of good nested lattice codes: **Loeliger '97, Forney-Trott-Chung '00, Erez-Litsyn-Zamir '05, Ordentlich-Erez '16.**
- **Erez-Zamir '04:** Nested lattice codes can achieve the Gaussian capacity.
- **Zamir-Shamai-Erez '02:** Excellent framework for multi-terminal binning.



Disclaimer

- In the interest of conveying the key intuitions, I will now switch to heuristic arguments for the next few slides.

- Although the steps I will show do not constitute a formal proof, all of the conclusions we will reach can be shown rigorously (with some extra steps in between).

- Interested in the details? See
  B. Nazer and M. Gastpar, Compute-and-Forward: Harnessing Interference through Structured Codes, IEEE Transactions on Information Theory, vol. 57, no. 10, pp. 6463-6486, October 2011.

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\mathsf{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\mathsf{eff}}\|^2 \leq \sigma^2_{\mathsf{eff}}$, then $\mathsf{P}(\mathbf{z}_{\mathsf{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\mathsf{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\mathsf{eff}}\|^2 \leq \sigma^2_{\mathsf{eff}}$, then $\mathsf{P}(\mathbf{z}_{\mathsf{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

- The number of codewords in the nested lattice codebook $\Lambda_{\mathsf{F}} \cap \mathcal{V}_{\mathsf{C}}$ is

$$\frac{\mathrm{Vol}(\mathcal{V}_{\mathsf{C}})}{\mathrm{Vol}(\mathcal{V}_{\mathsf{F}})} \approx \frac{\mathrm{Vol}\left(\mathcal{B}\left(\mathbf{0}, \sqrt{n\mathsf{SNR}}\right)\right)}{\mathrm{Vol}\left(\mathcal{B}\left(\mathbf{0}, \sqrt{n\sigma^2_{\mathsf{eff}}}\right)\right)} = \left(\frac{P}{\sigma^2_{\mathsf{eff}}}\right)^{n/2}$$

Nested Lattice Code Heuristics

- The Voronoi region $\mathcal{V}_{\mathsf{C}}$ of the coarse lattice Λ_{C} enforces the power constraint: If $\mathbf{x} \in \mathcal{V}_{\mathsf{C}}$, then $\frac{1}{n}\|\mathbf{x}\|^2 \leq P$.

- The Voronoi region $\mathcal{V}_{\mathsf{F}}$ of the fine lattice Λ_{F} tolerates noise up to variance σ^2_{eff}: For "well-behaved" noise $\mathbf{z}_{\mathsf{eff}}$, if $\frac{1}{n}\mathbb{E}\|\mathbf{z}_{\mathsf{eff}}\|^2 \leq \sigma^2_{\mathsf{eff}}$, then $\mathsf{P}(\mathbf{z}_{\mathsf{eff}} \notin \mathcal{V}_{\mathsf{F}}) < \delta$ for some small δ.

- The number of codewords in the nested lattice codebook $\Lambda_{\mathsf{F}} \cap \mathcal{V}_{\mathsf{C}}$ is

$$\frac{\mathrm{Vol}(\mathcal{V}_{\mathsf{C}})}{\mathrm{Vol}(\mathcal{V}_{\mathsf{F}})} \approx \frac{\mathrm{Vol}\left(\mathcal{B}\left(\mathbf{0}, \sqrt{n\mathsf{SNR}}\right)\right)}{\mathrm{Vol}\left(\mathcal{B}\left(\mathbf{0}, \sqrt{n\sigma^2_{\mathsf{eff}}}\right)\right)} = \left(\frac{P}{\sigma^2_{\mathsf{eff}}}\right)^{n/2}$$

- Can show that the achievable rate satisfies $R > \frac{1}{2}\log\left(\frac{P}{\sigma^2_{\mathsf{eff}}}\right) - \delta$.

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z}$$

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{k=1}^{K} (h_k - a_k)\mathbf{x}_k + \mathbf{z}}_{\text{Effective Noise}}$$

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{k=1}^{K} (h_k - a_k)\mathbf{x}_k + \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{k=1}^{K} (h_k - a_k)\mathbf{x}_k + \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = 1 + P\sum_{k=1}^{K}(h_k - a_k)^2 = 1 + P\|\mathbf{h} - \boldsymbol{a}\|^2$$

where $\mathbf{h} \in \mathbb{R}^K$ and $\boldsymbol{a} \in \mathbb{Z}^K$.

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{k=1}^{K} (h_k - a_k)\mathbf{x}_k + \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = 1 + P\sum_{k=1}^{K}(h_k - a_k)^2 = 1 + P\|\mathbf{h} - \boldsymbol{a}\|^2$$

  where $\mathbf{h} \in \mathbb{R}^K$ and $\boldsymbol{a} \in \mathbb{Z}^K$.

- Overall, we can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{P}{1 + P\|\mathbf{h} - \boldsymbol{a}\|^2}\right)$.

Compute-and-Forward: First Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. We can write this as

$$\mathbf{y} = \sum_{k=1}^{K} h_k \mathbf{x}_K + \mathbf{z} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{k=1}^{K} (h_k - a_k)\mathbf{x}_k + \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = 1 + P\sum_{k=1}^{K}(h_k - a_k)^2 = 1 + P\|\mathbf{h} - \boldsymbol{a}\|^2$$

  where $\mathbf{h} \in \mathbb{R}^K$ and $\boldsymbol{a} \in \mathbb{Z}^K$.

- Overall, we can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{P}{1 + P\|\mathbf{h} - \boldsymbol{a}\|^2}\right)$.
- Best $\boldsymbol{a} = \text{round}(\mathbf{h})$. (Not an interesting Diophantine problem.)

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} \; =$$

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}}$$

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR}\sum_{k=1}^{K}(\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2 .$$

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR}\sum_{k=1}^{K}(\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2 .$$

- We can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{\text{SNR}}{\alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2}\right)$.

Compute-and-Forward: Second Attempt

- Each encoder maps its message m_k to a lattice codeword $\mathbf{x}_k$.
- The decoder observes $\mathbf{y}$. It scales by $\alpha \in \mathbb{R}$ to get $\mathbf{h} \in \mathbb{R}^K$ "closer" to $\boldsymbol{a} \in \mathbb{Z}^K$. We can write this as

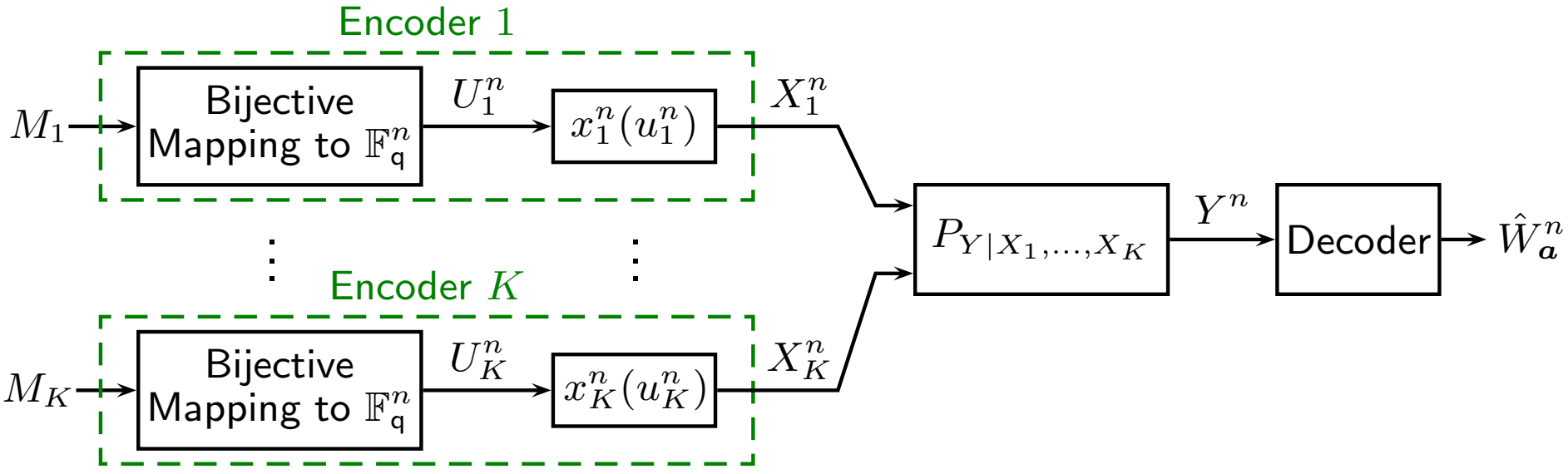
$$\alpha \mathbf{y} = \underbrace{\sum_{k=1}^{K} a_k \mathbf{x}_k}_{\text{Lattice Codeword}} + \underbrace{\sum_{\ell=1}^{L} (\alpha h_k - a_k)\mathbf{x}_k + \alpha \mathbf{z}}_{\text{Effective Noise}} = \mathbf{v} + \mathbf{z}_{\text{eff}}$$

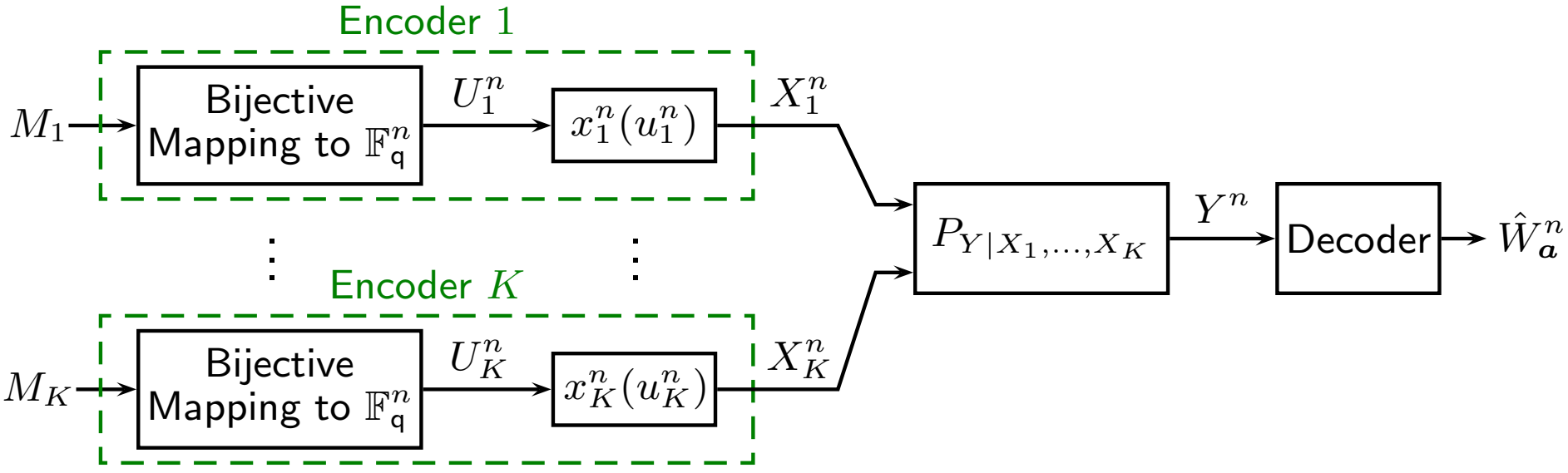
- The effective noise variance is

$$\sigma^2_{\text{eff}} = \frac{1}{n}\mathbb{E}\|\mathbf{z}_{\text{eff}}\|^2 = \alpha^2 + \text{SNR}\sum_{k=1}^{K}(\alpha h_k - a_k)^2 = \alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2 .$$

- We can decode $\mathbf{v}$ if $R < \frac{1}{2}\log\left(\frac{\text{SNR}}{\alpha^2 + \text{SNR}\|\alpha \mathbf{h} - \boldsymbol{a}\|^2}\right)$.
- Finding the best $\boldsymbol{a}$ corresponds to finding the shortest vector in the lattice $(\text{SNR}^{-1}\mathbf{I} + \mathbf{h}\mathbf{h}^{\mathsf{T}})^{-1/2}\mathbb{Z}^K$.

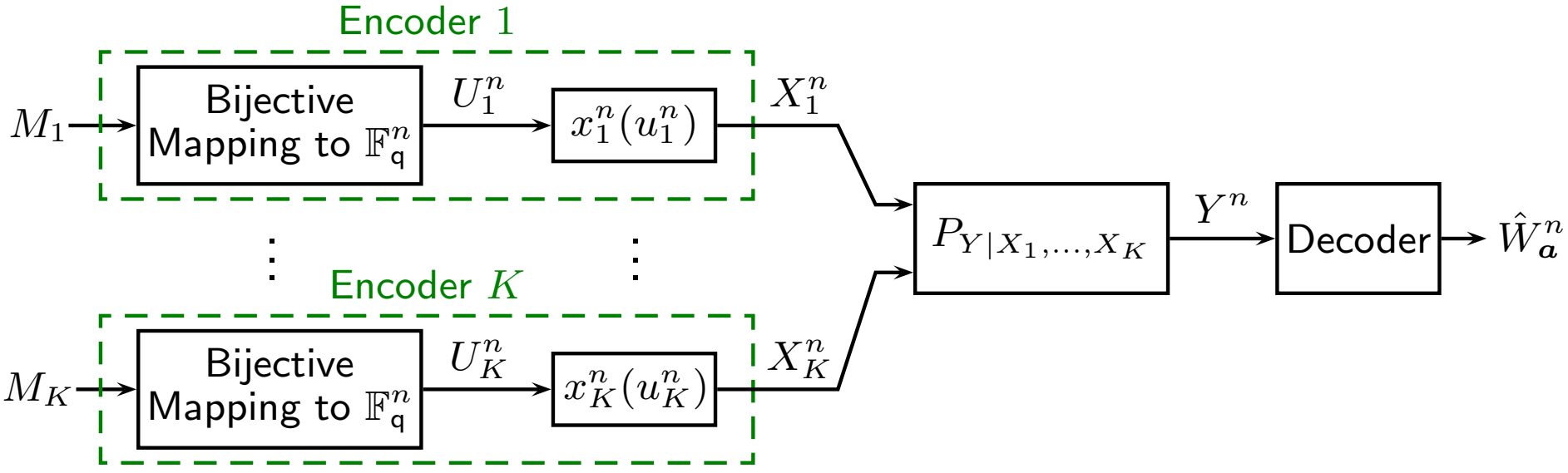
Compute-and-Forward: Beyond Gaussian Channels





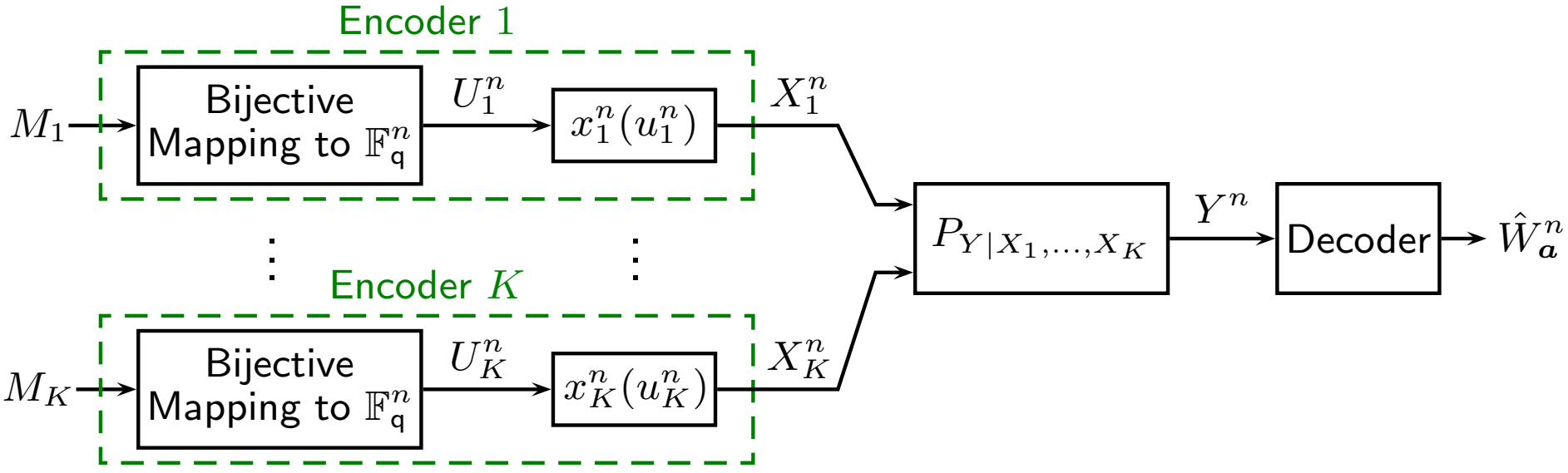
- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.

Compute-and-Forward: Beyond Gaussian Channels



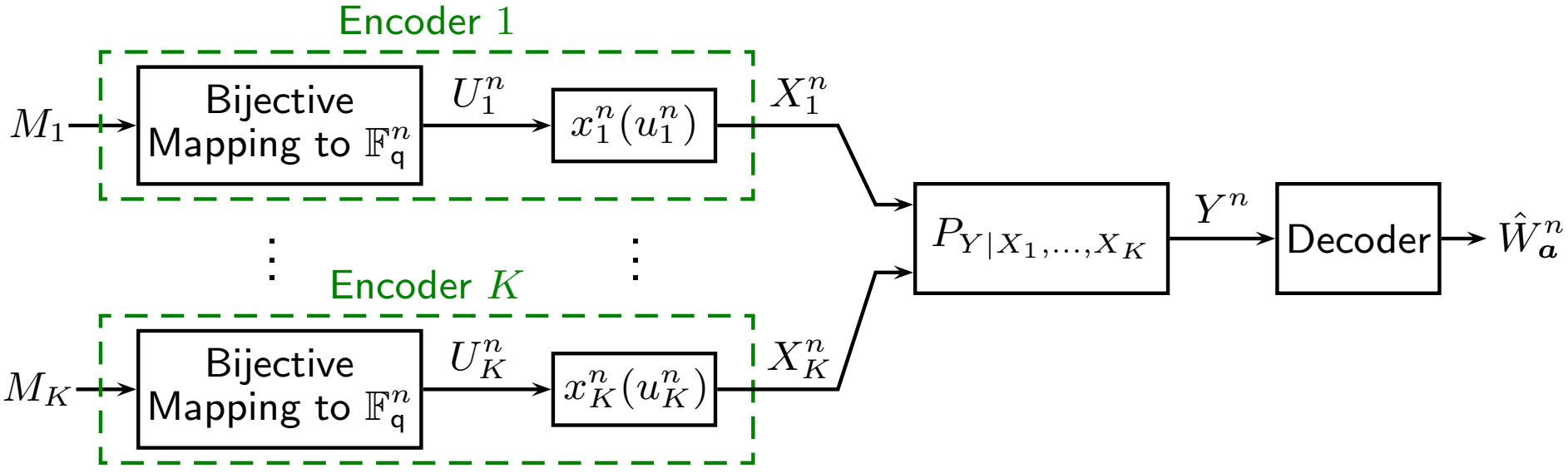
- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_{\mathsf{q}}^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*

Compute-and-Forward: Beyond Gaussian Channels



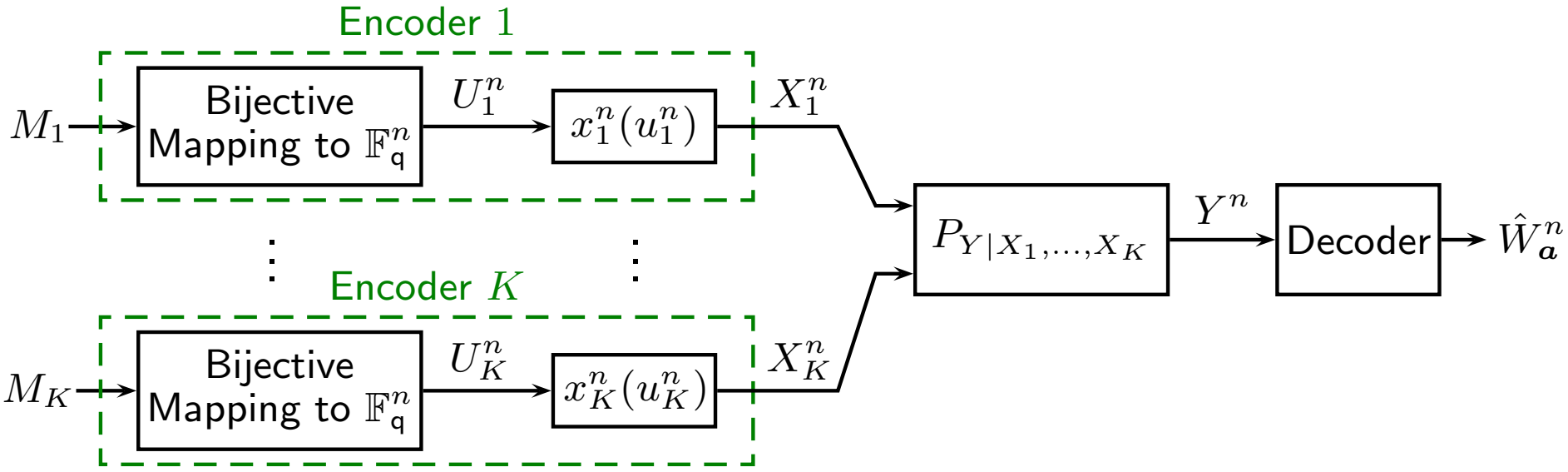
- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_{\mathsf{q}}^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_{\mathsf{q}}^K$

Compute-and-Forward: Beyond Gaussian Channels



- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_{\mathsf{q}}^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_{\mathsf{q}}^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_{\mathsf{q}}^n$ to each $y^n \in \mathcal{Y}^n$.

Compute-and-Forward: Beyond Gaussian Channels



- Messages: $m_k \in [2^{nR_k}] \triangleq \{0, \ldots, 2^{nR_k} - 1\}$, $k = 1, \ldots, K$.
- Encoders: mappings $(u_k^n, x_k^n)(m_k) \in \mathbb{F}_q^n \times \mathcal{X}_k^n$, $k = 1, \ldots, K$ such that $u_k^n(m_k)$ is *bijective.*
- Linear Combination: $w_{\boldsymbol{a}}^n \triangleq \bigoplus_k a_k u_k^n(m_k)$, $\boldsymbol{a} = [a_1 \ \cdots \ a_K] \in \mathbb{F}_q^K$
- Decoder: assigns an estimate $\hat{w}_{\boldsymbol{a}}^n \in \mathbb{F}_q^n$ to each $y^n \in \mathcal{Y}^n$.
- Probability of Error: For uniformly distributed messages $M_1, \ldots, M_K$, want vanishing probability of error $\mathsf{P}\{\hat{W}_{\boldsymbol{a}}^n \neq W_{\boldsymbol{a}}^n\}$.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

High-Level Intuition:

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.

- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences. (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.

- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences. (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical. (Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.

- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences. (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical. (Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$. (Suboptimal decoding rule)
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

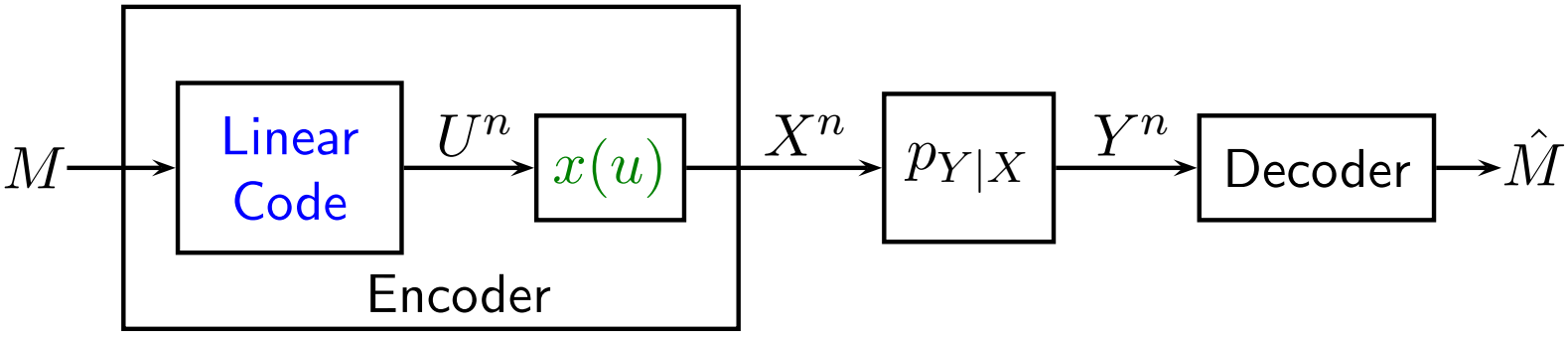
$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

High-Level Intuition: (Low-Level Reality)

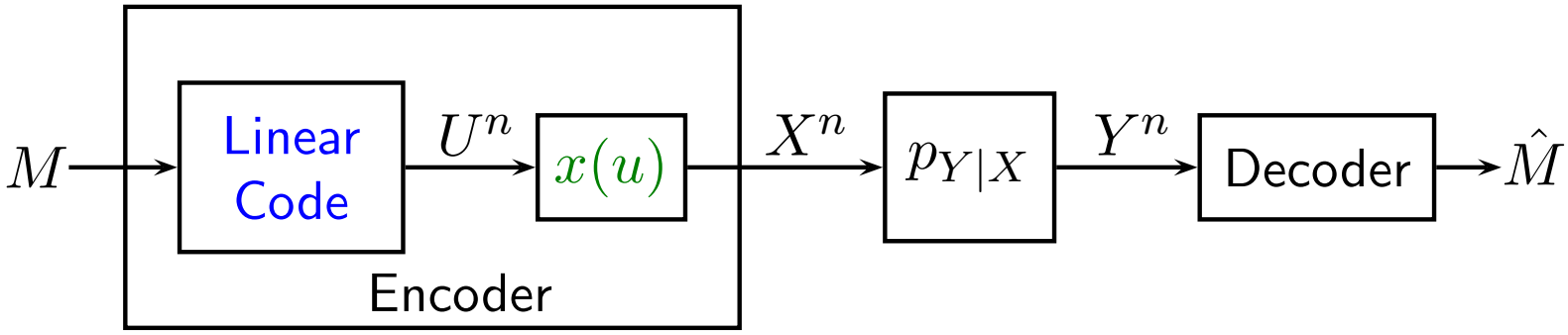
- Input Distribution: Want U_k^n to look typical with respect to pmf $p_{U_k}(u_k)$. There are $\approx 2^{nH(U_k)}$ typical sequences.
  (Linear codewords look uniform.)
- True Codeword: Want $(W_{\boldsymbol{a}}^n, Y^n)$ to look jointly typical.
  (Proof is actually a bit involved.)
- Decoder searches for sequences $\tilde{w}_{\boldsymbol{a}}^n$ that are jointly typical with Y^n. There are $\approx 2^{nH(W_{\boldsymbol{a}}|Y)}$ possible sequences. If only one such sequence is jointly typical, declare it as the estimate $\hat{W}_{\boldsymbol{a}}^n$ of the linear combination $W_{\boldsymbol{a}}^n = a_1 U_1^n \oplus \cdots \oplus a_K U_K^n$.
  (Suboptimal decoding rule)
- We can show that, for this decoding strategy, we can achieve any rate tuple $(R_1, \ldots, R_K)$ satisfying

$$R_k < H(U_k) - H(W_{\boldsymbol{a}}|Y).$$

  (Not a mutual information and can be negative.)

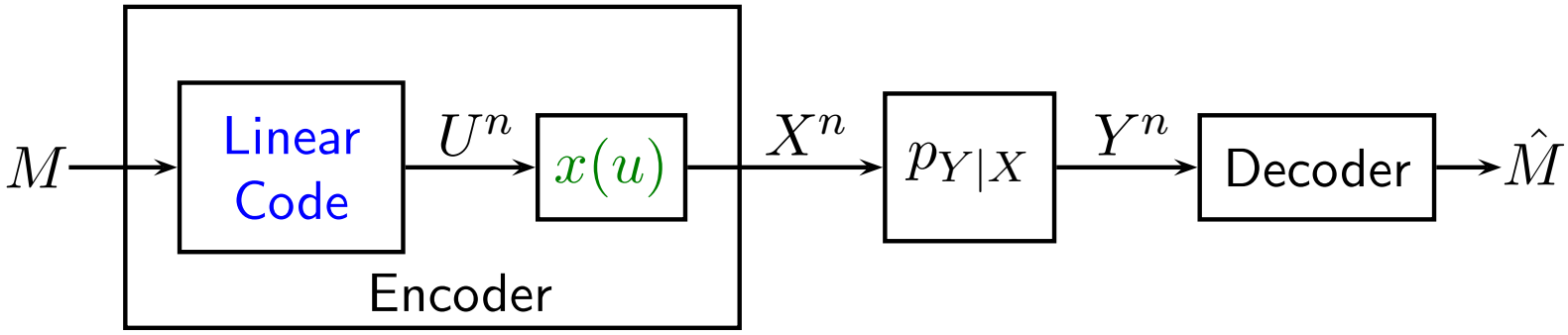


Code Construction:



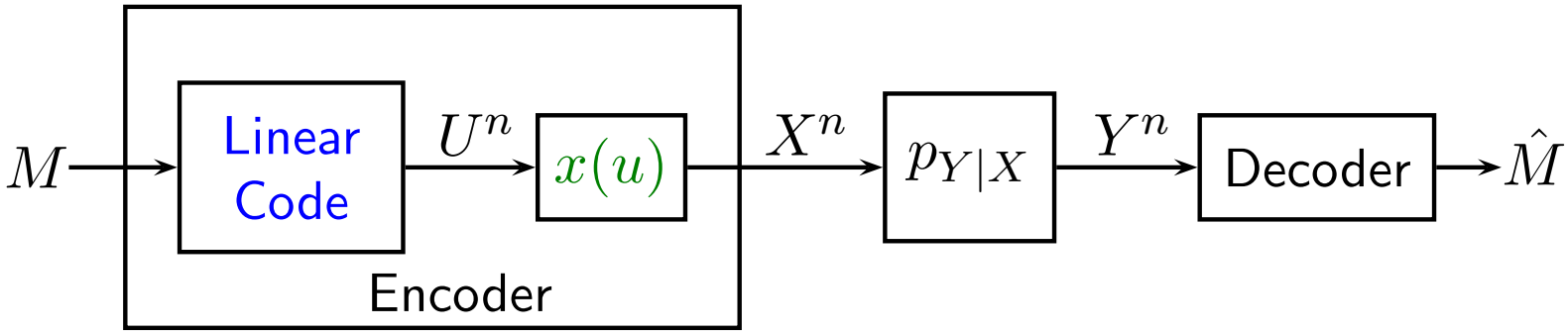
Code Construction:

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



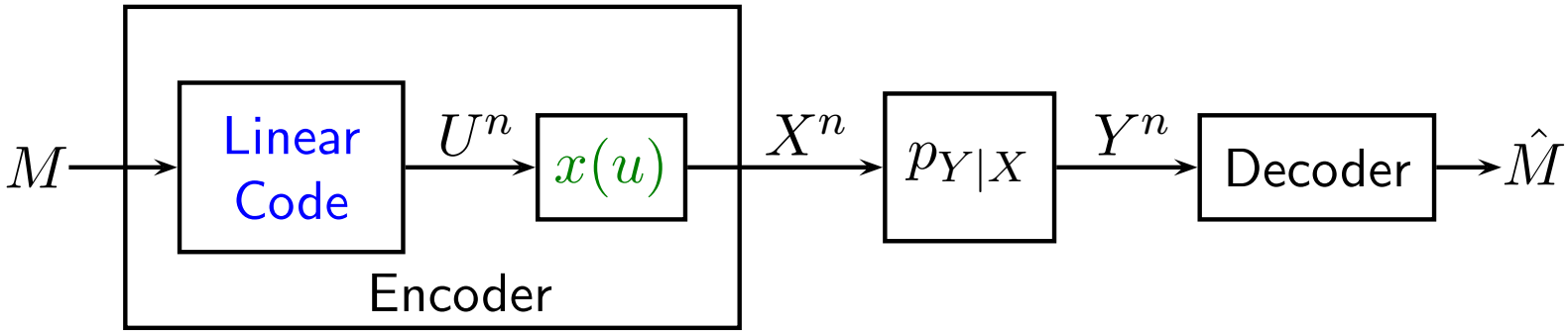
Code Construction:

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



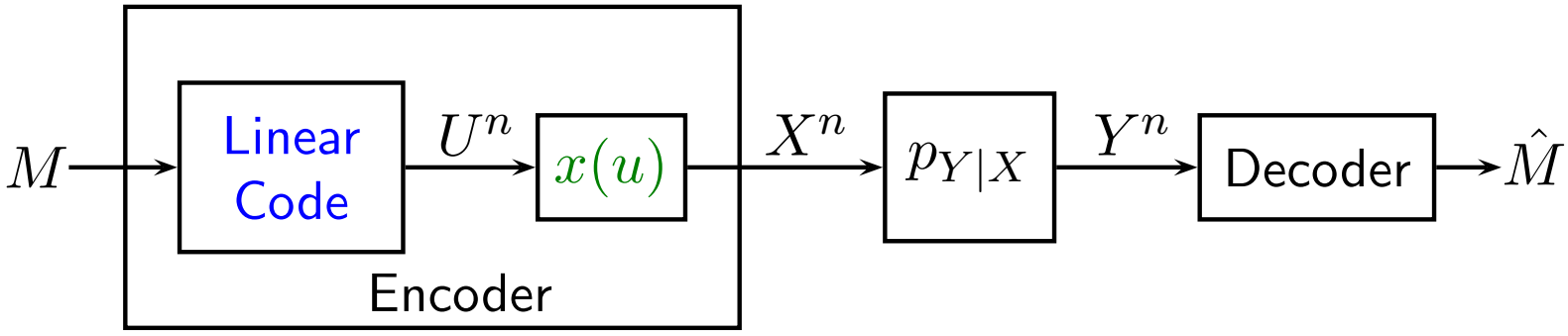
Code Construction:

- 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. $\mathrm{Unif}(\mathbb{F}_\mathsf{q})$. Let G be a realization.



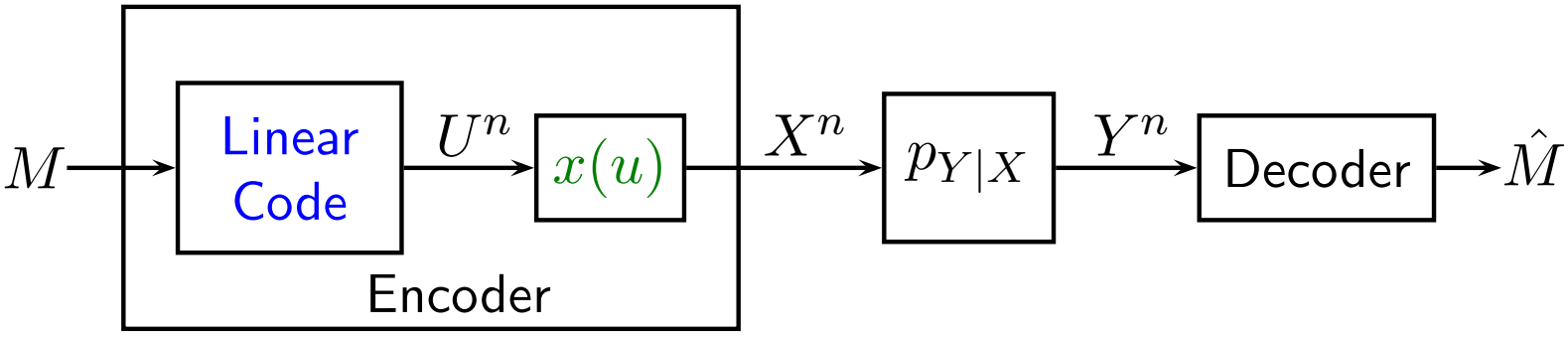
Code Construction:

- 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.



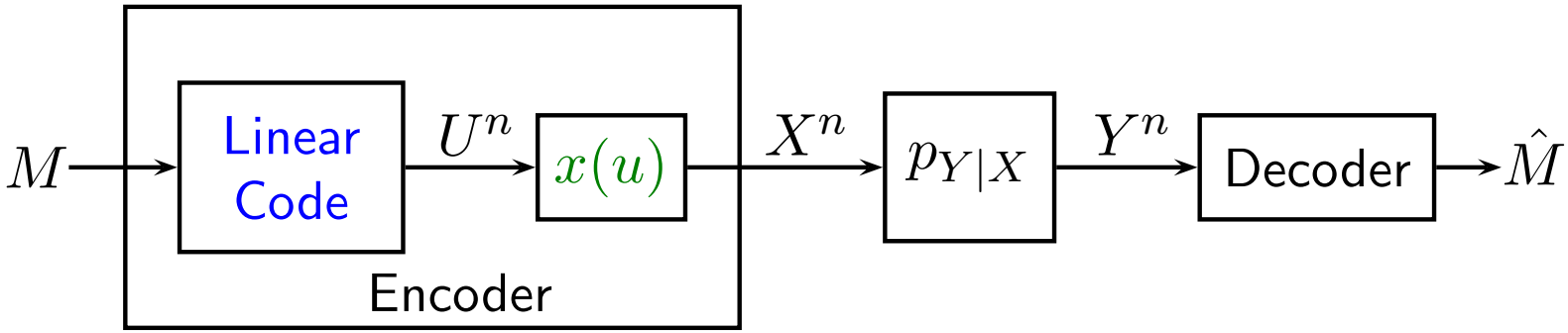
Code Construction:

- 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:

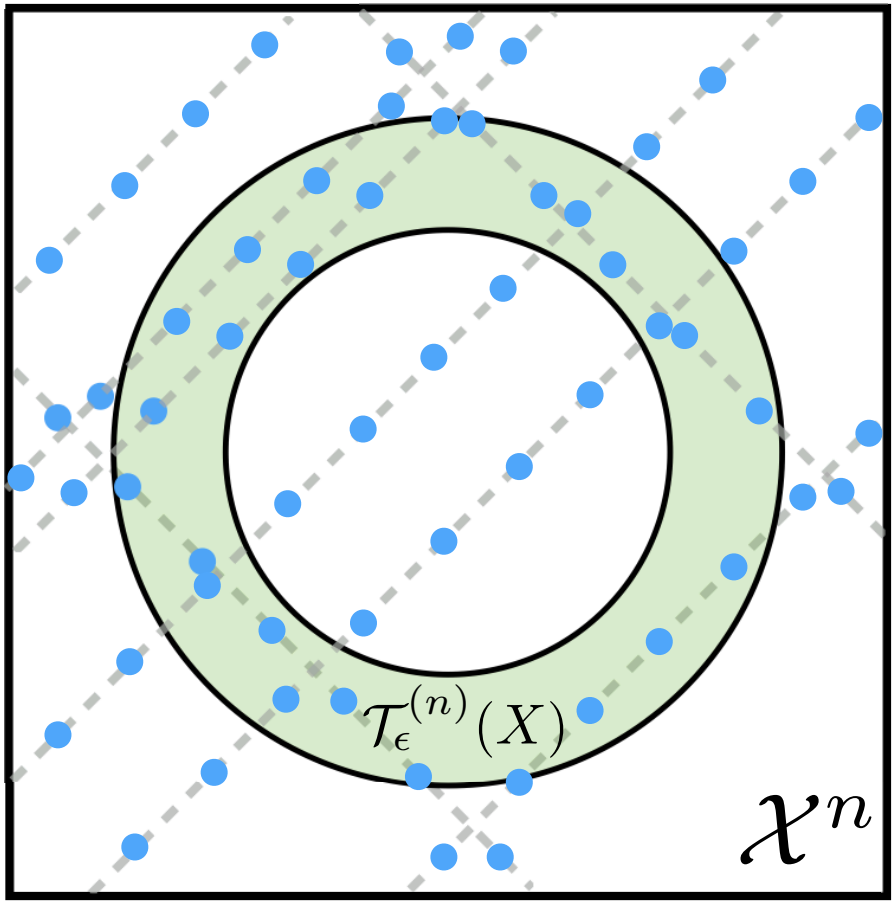
- Pick a finite field $\mathbb{F}_{\mathsf{q}}$ and a symbol mapping $x : \mathbb{F}_{\mathsf{q}} \to \mathcal{X}$.
- Set $\kappa = nR/\log(\mathsf{q})$.
- Draw a random generator matrix $\mathbf{G} \in \mathbb{F}_{\mathsf{q}}^{\kappa \times n}$ elementwise i.i.d. $\mathrm{Unif}(\mathbb{F}_{\mathsf{q}})$.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

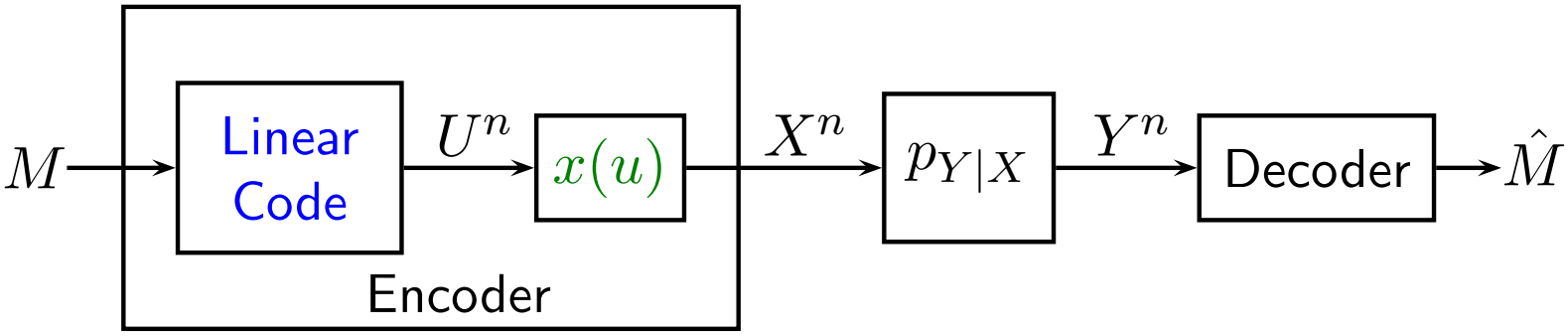
Random i.i.d. Codebooks



Random Linear Codes

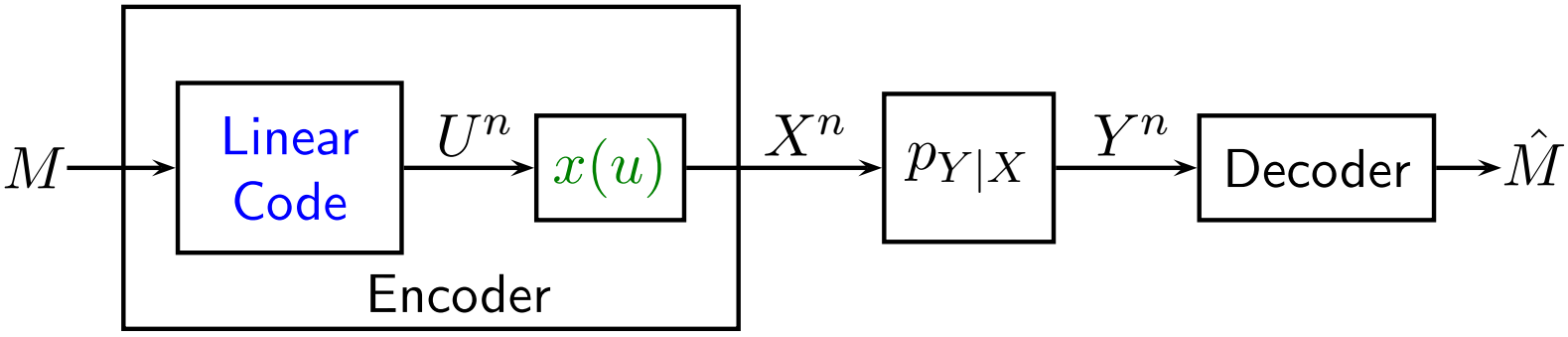
- Codewords are pairwise independent of one another.
- Codewords are uniformly distributed over $\mathbb{F}_q^n$.

Point-to-Point Channels: Linear Codes



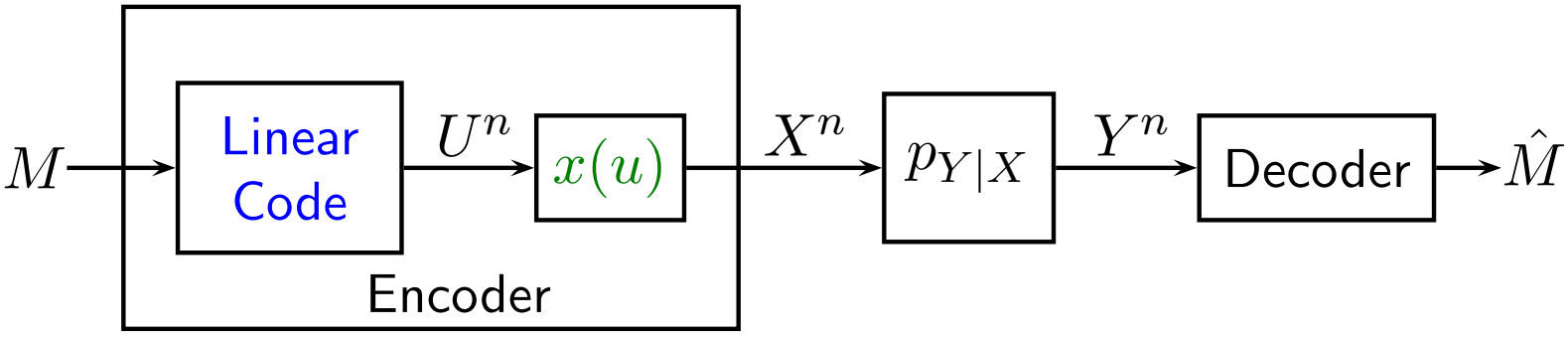
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**

Point-to-Point Channels: Linear Codes



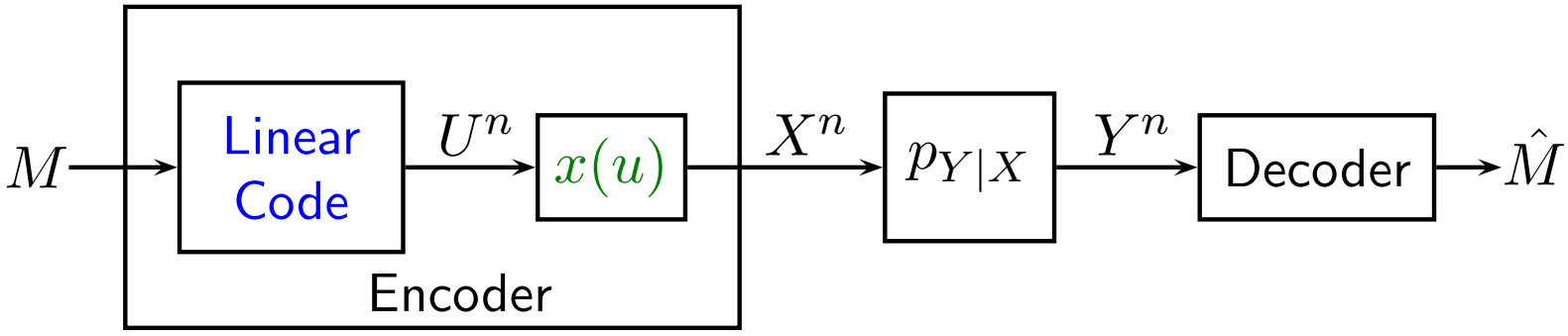
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the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

Point-to-Point Channels: Linear Codes



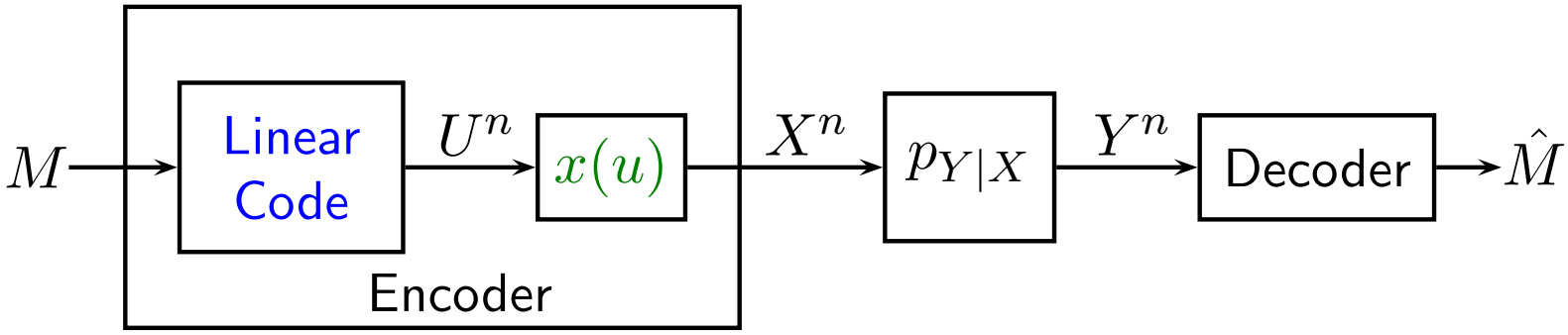
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the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

Point-to-Point Channels: Linear Codes



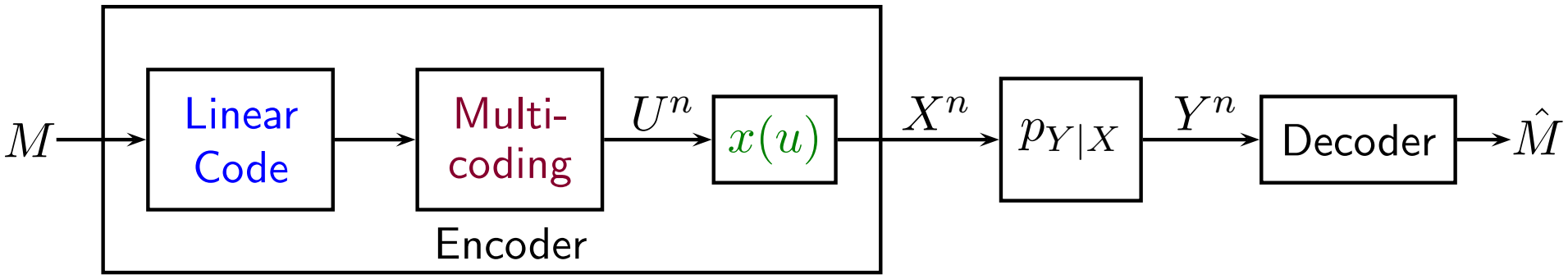
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the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

Point-to-Point Channels: Linear Codes



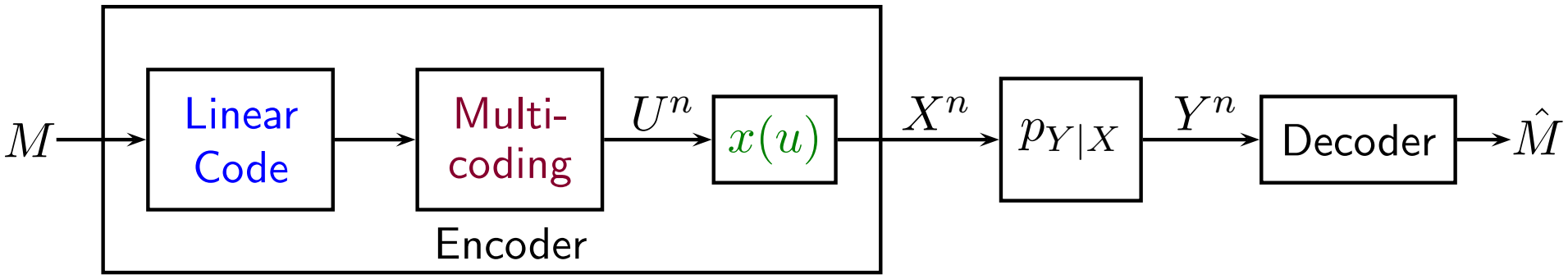
- Well known that a direct application of linear coding is not sufficient to reach the point-to-point capacity, **Ahlswede '71.**
- **Gallager '68:** Pick $\mathbb{F}_q$ with $q \gg \mathcal{X}$ and choose symbol mapping $x(u)$ to reach c.a.i.d. from $\mathrm{Unif}(\mathbb{F}_q)$. This can attain the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

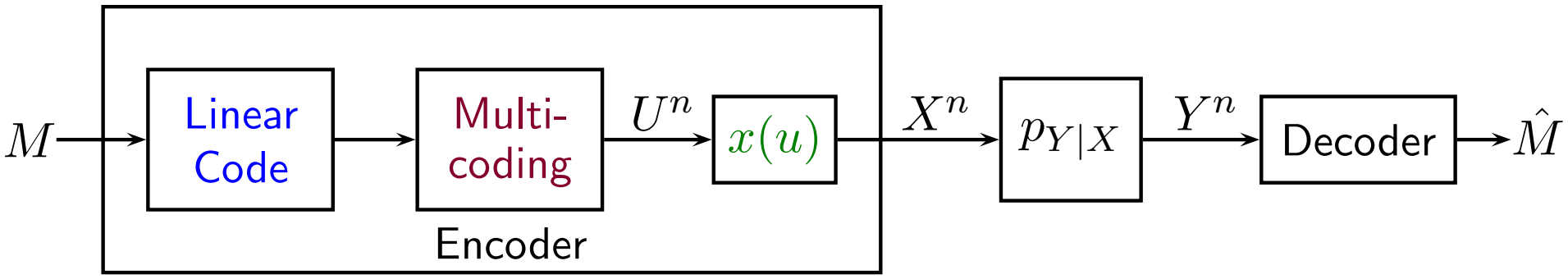
Point-to-Point Channels: Linear Codes + Multicoding



Code Construction:

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

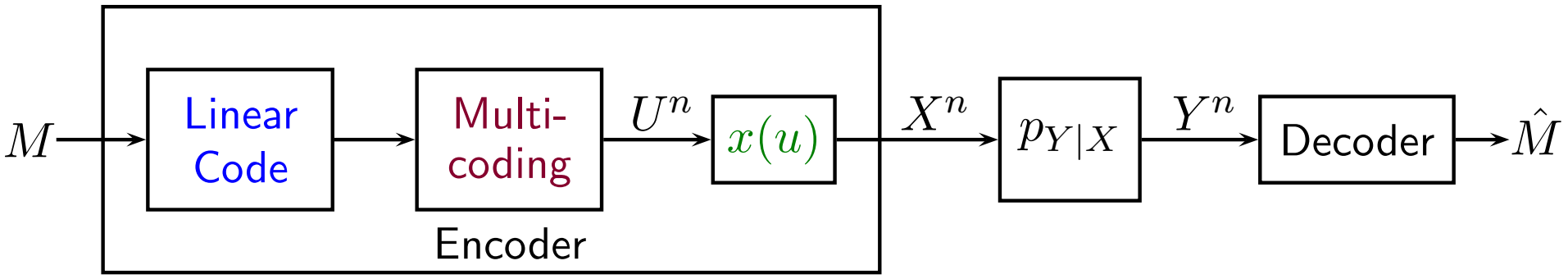
Point-to-Point Channels: Linear Codes + Multicoding



Code Construction:

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

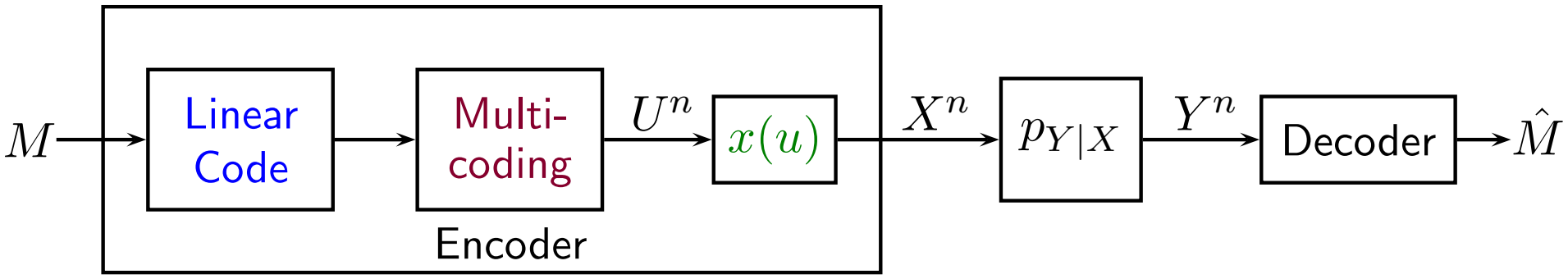
Point-to-Point Channels: Linear Codes + Multicoding



Code Construction:

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

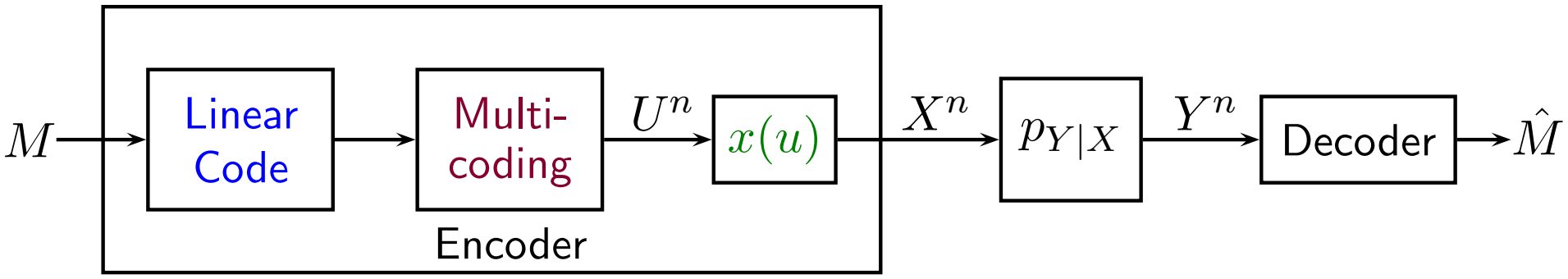
Point-to-Point Channels: Linear Codes + Multicoding



Code Construction:

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

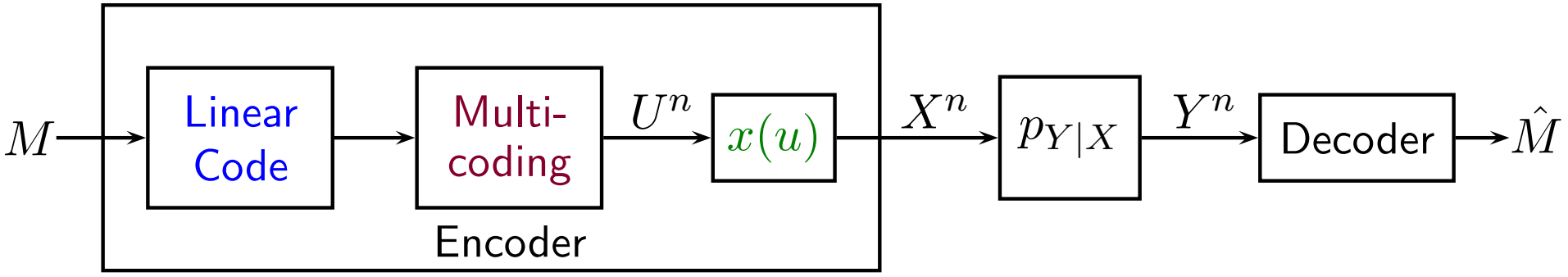
Point-to-Point Channels: Linear Codes + Multicoding



Code Construction:

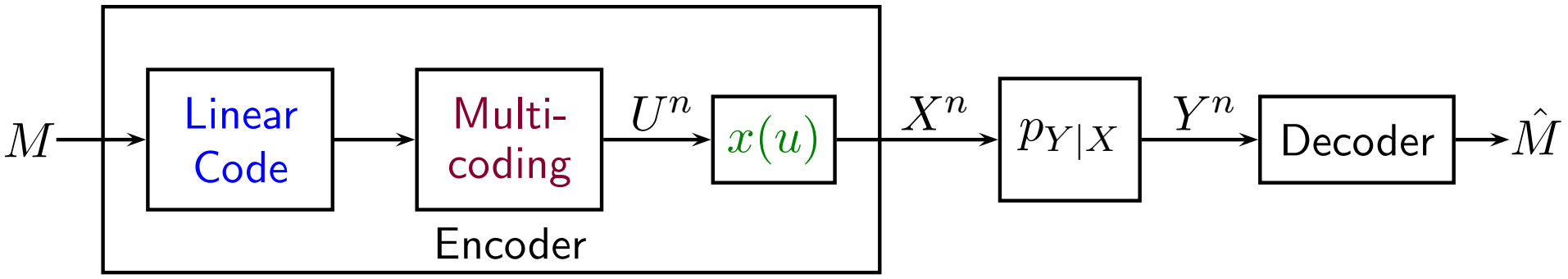
- Messages $m \in [2^{nR}]$ and auxiliary indices $l \in [2^{n\hat{R}}]$.
- Set $\kappa = n(R + \hat{R})/\log(\mathsf{q})$.
- Pick generator matrix G and dither d^n as before.
- Take q-ary expansions $\left[\boldsymbol{\nu}(m)\ \boldsymbol{\nu}(l)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$.
- Linear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

Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

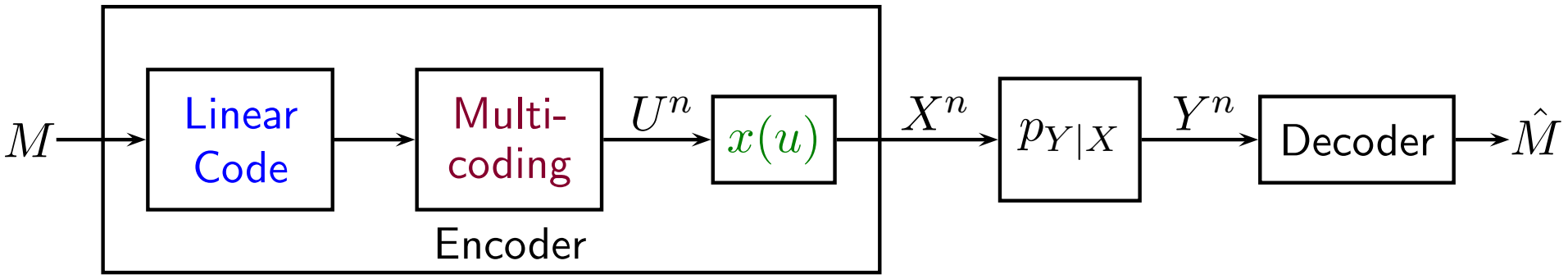
Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

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

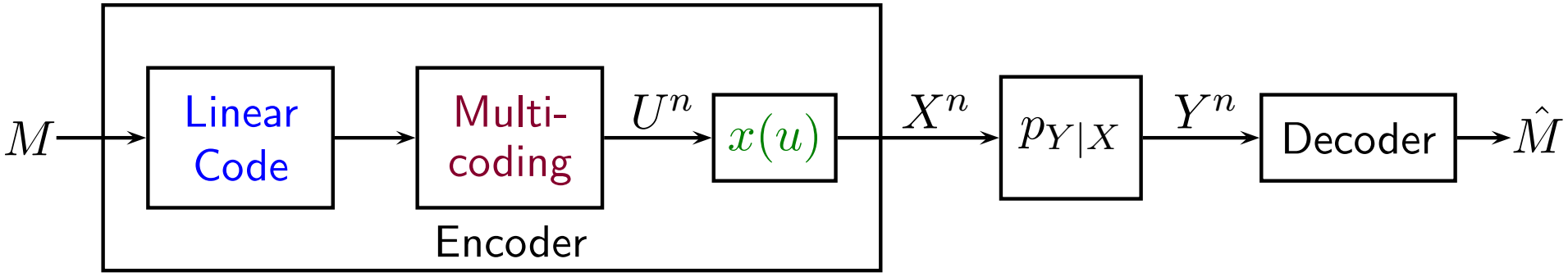
Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

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

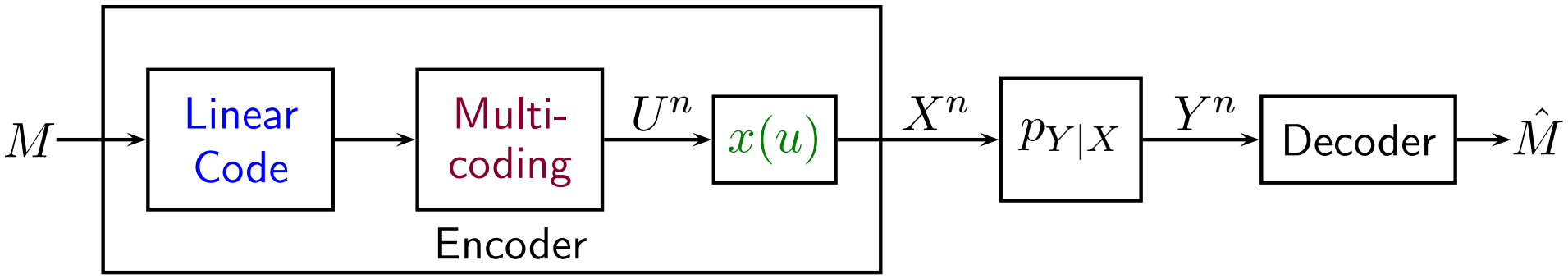
Point-to-Point Channels: Linear Codes + Multicoding



Encoding:

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

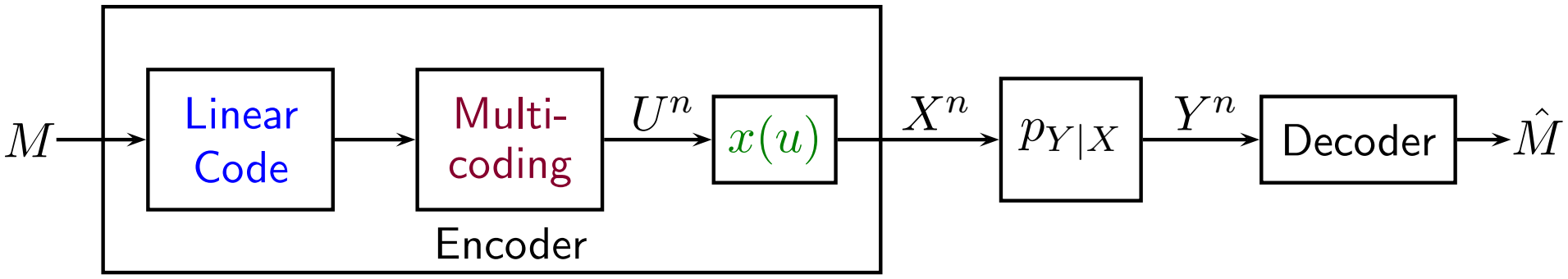
Point-to-Point Channels: Linear Codes + Multicoding



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 $x_i = x\big(u_i(m,l)\big)$.

Point-to-Point Channels: Linear Codes + Multicoding

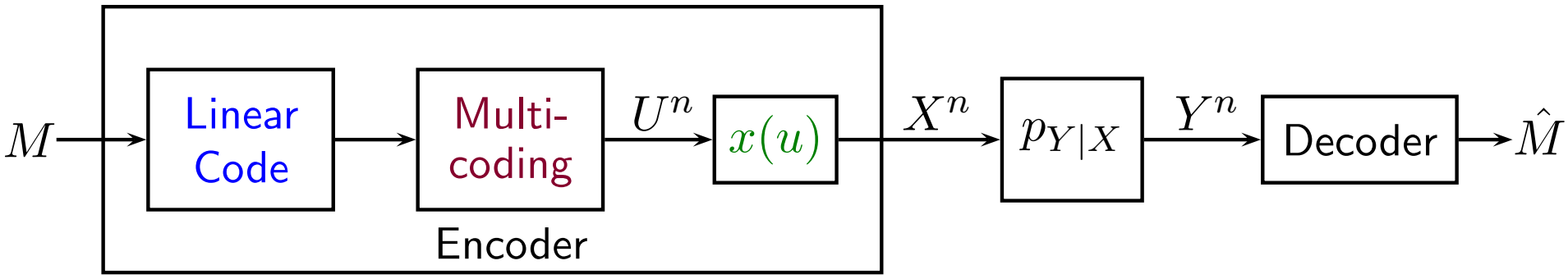


Encoding:

- 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:

Point-to-Point Channels: Linear Codes + Multicoding



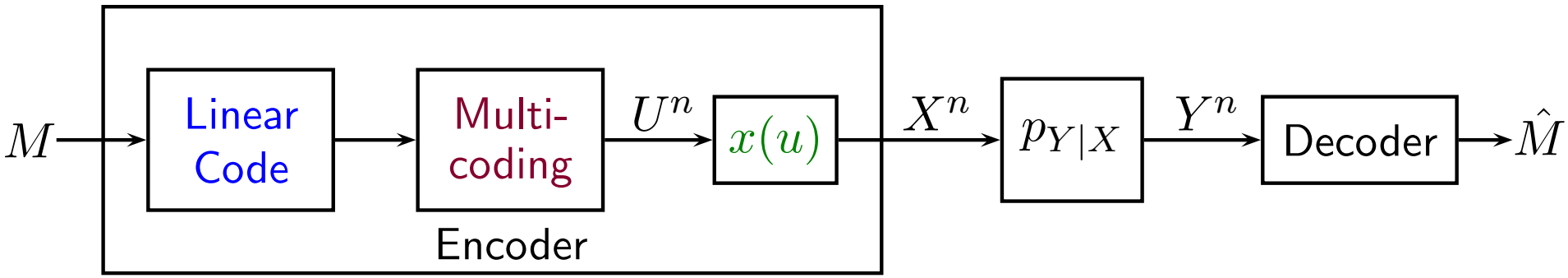
Encoding:

- 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



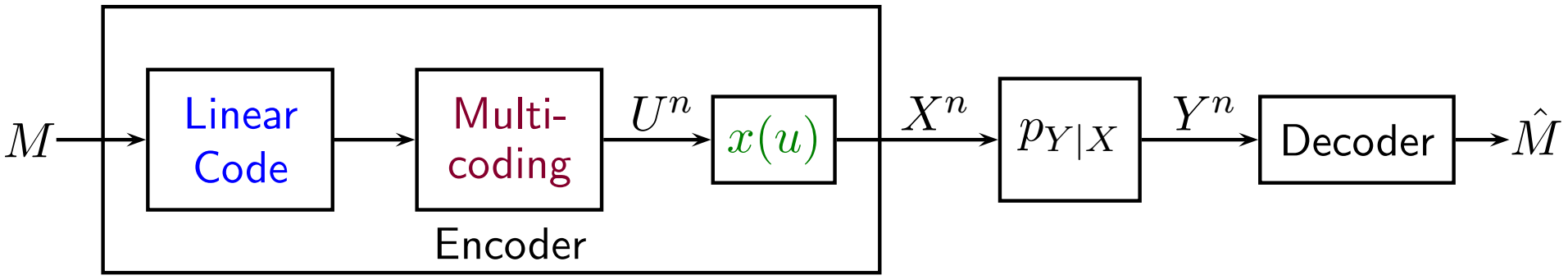
Encoding:

- 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



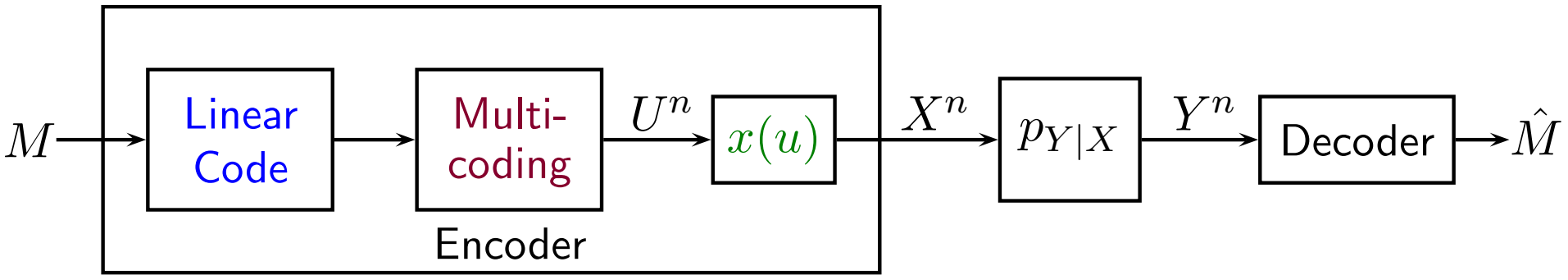
Theorem (Padakandla-Pradhan '13)

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)

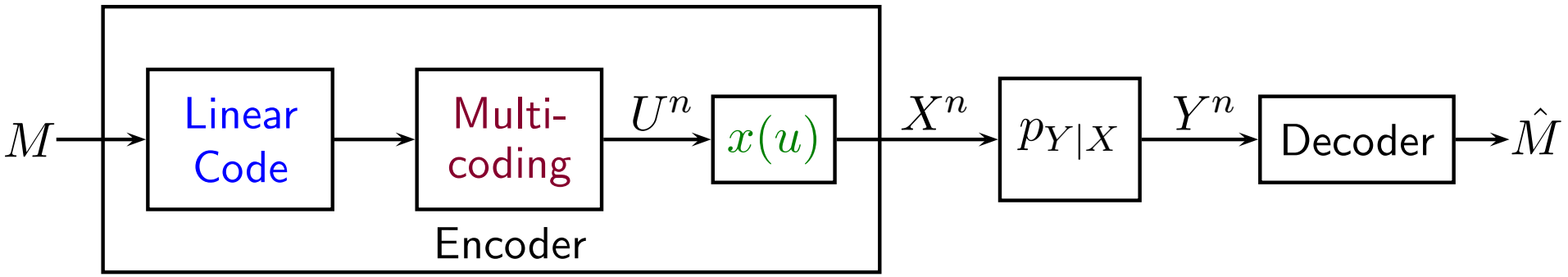
Any rate R satisfying

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

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)

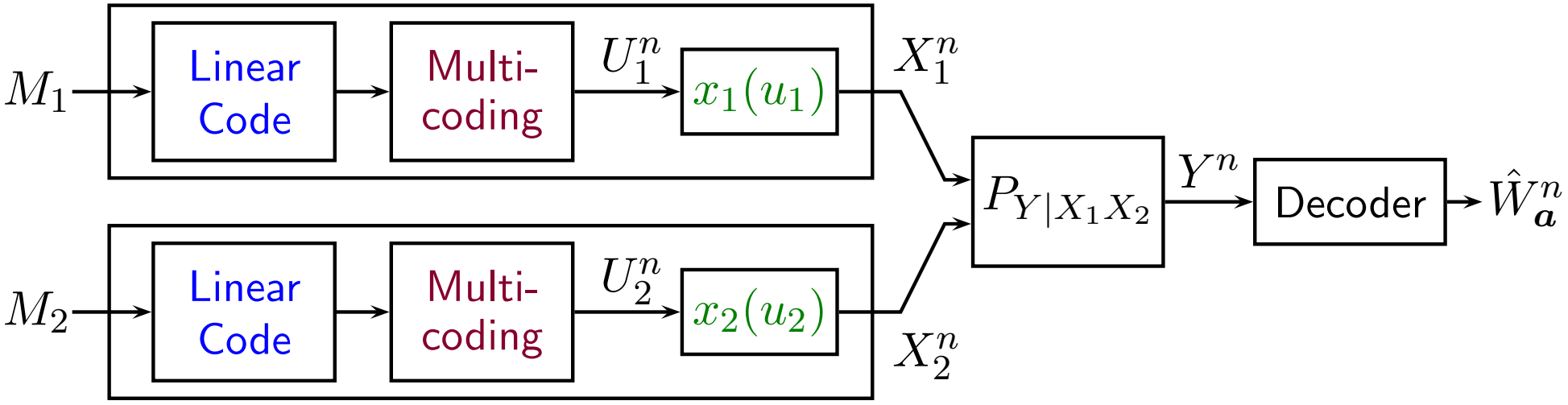
Any rate R satisfying

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

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.

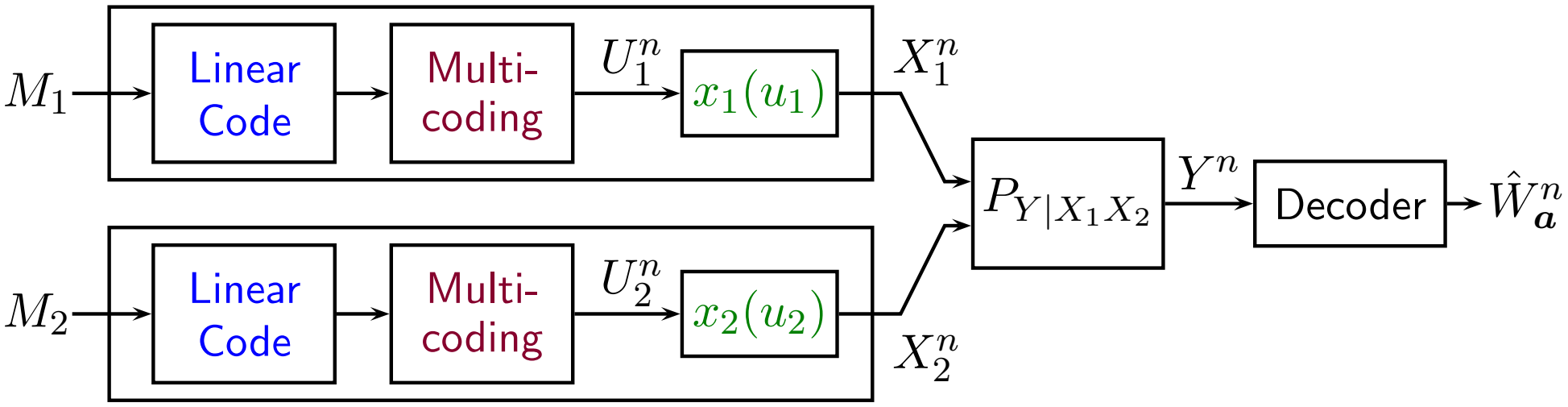
Nested Linear Coding Architecture



Code Construction:

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

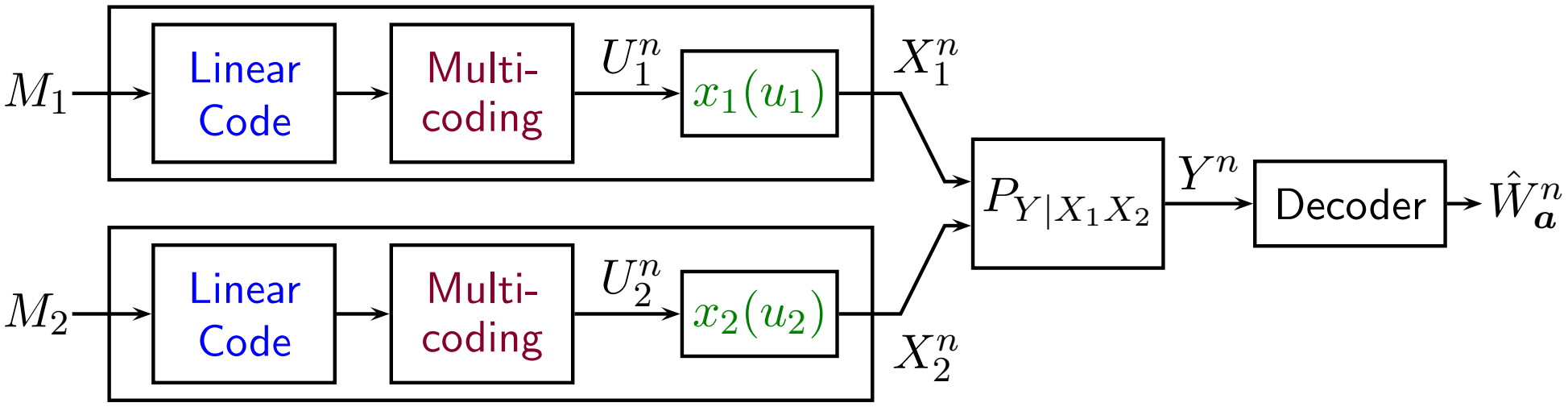
Nested Linear Coding Architecture



Code Construction:

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

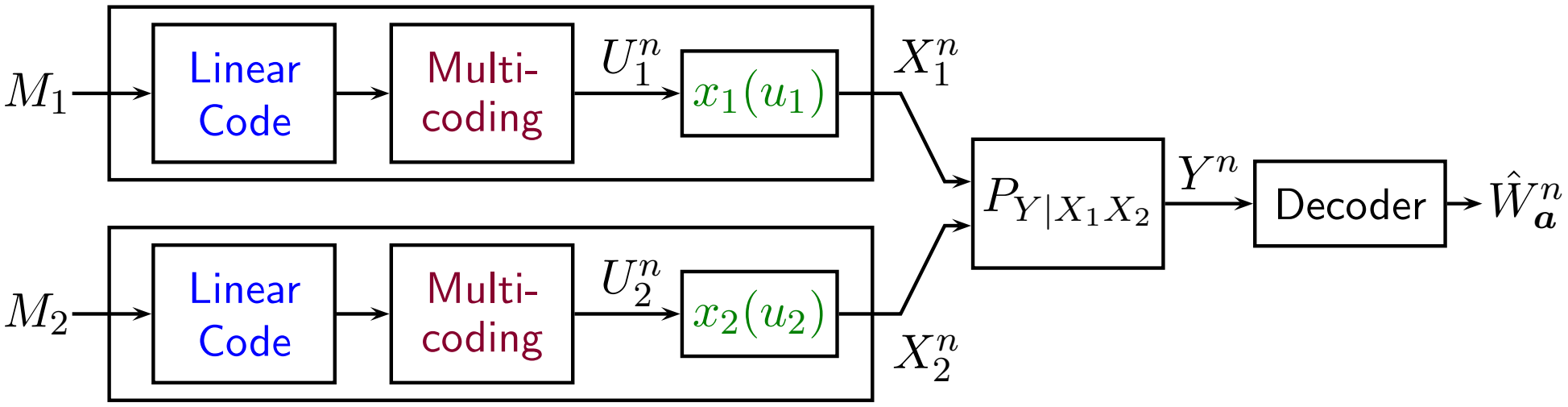
Nested Linear Coding Architecture



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.

Nested Linear Coding Architecture

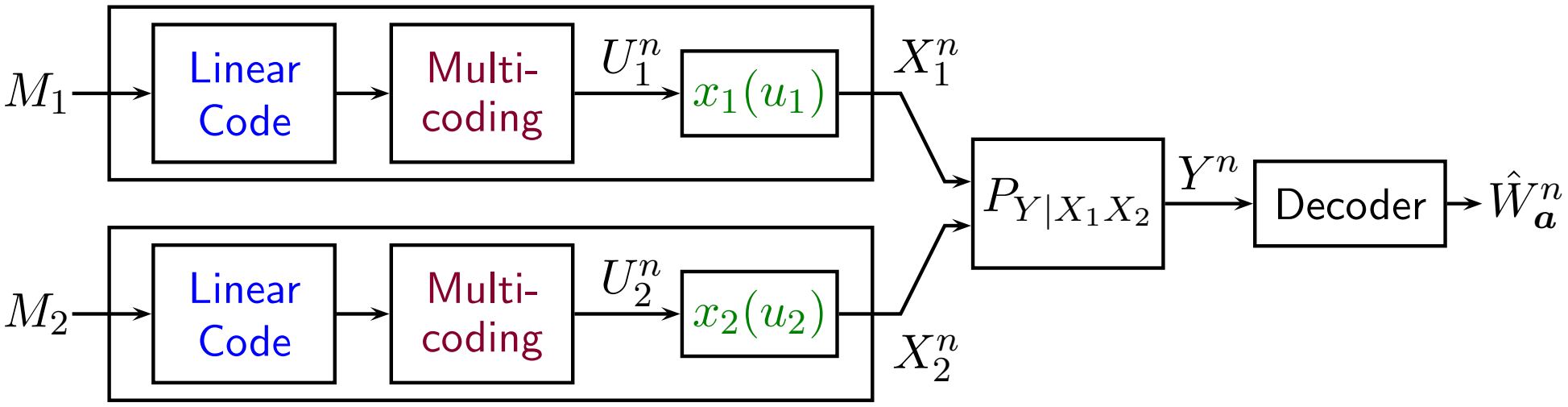


Code Construction:

- 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 $\begin{bmatrix}\boldsymbol{\nu}(m_1) & \boldsymbol{\nu}(l_1)\end{bmatrix} \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

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

Nested Linear Coding Architecture

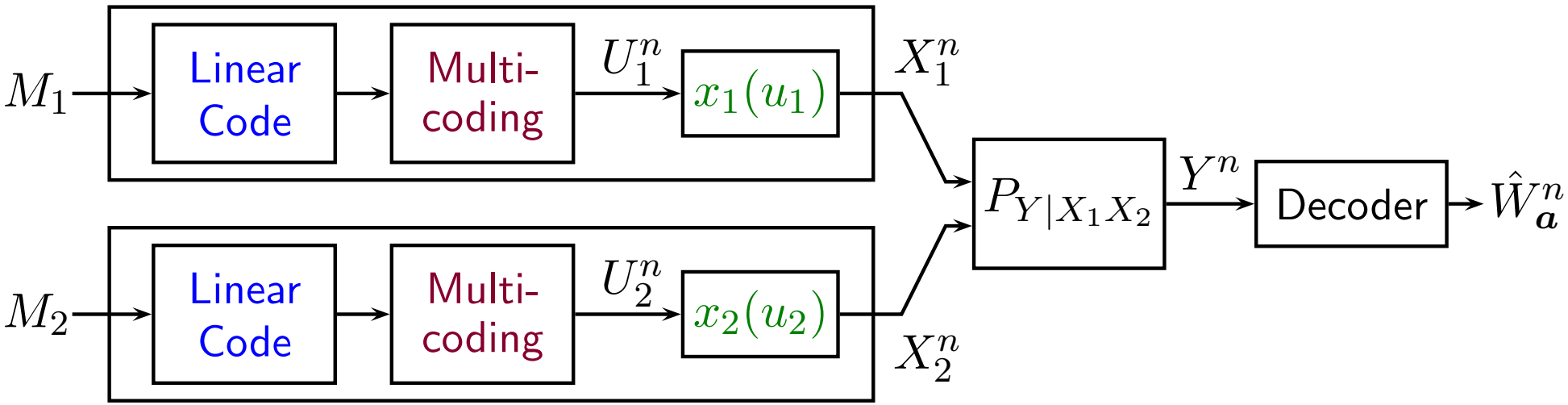


Code Construction:

- 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 $\quad \left[\boldsymbol{\eta}(m_1, l_1)\right] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

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

Nested Linear Coding Architecture



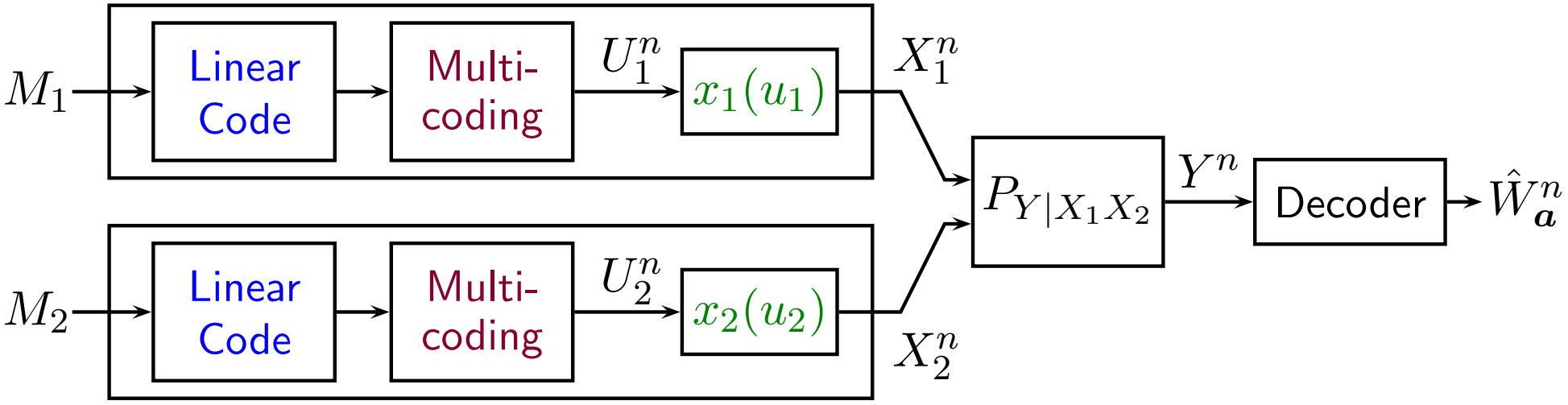
Code Construction:

- 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 $\quad \big[\boldsymbol{\eta}(m_1, l_1)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$

  $\big[\boldsymbol{\eta}(m_2, l_2)\big] \in \mathbb{F}_{\mathsf{q}}^{\kappa}$
- Linear codewords: $\quad u_1^n(m_1, l_1) = \boldsymbol{\eta}(m_1, l_1)\mathsf{G} \oplus d_1^n$

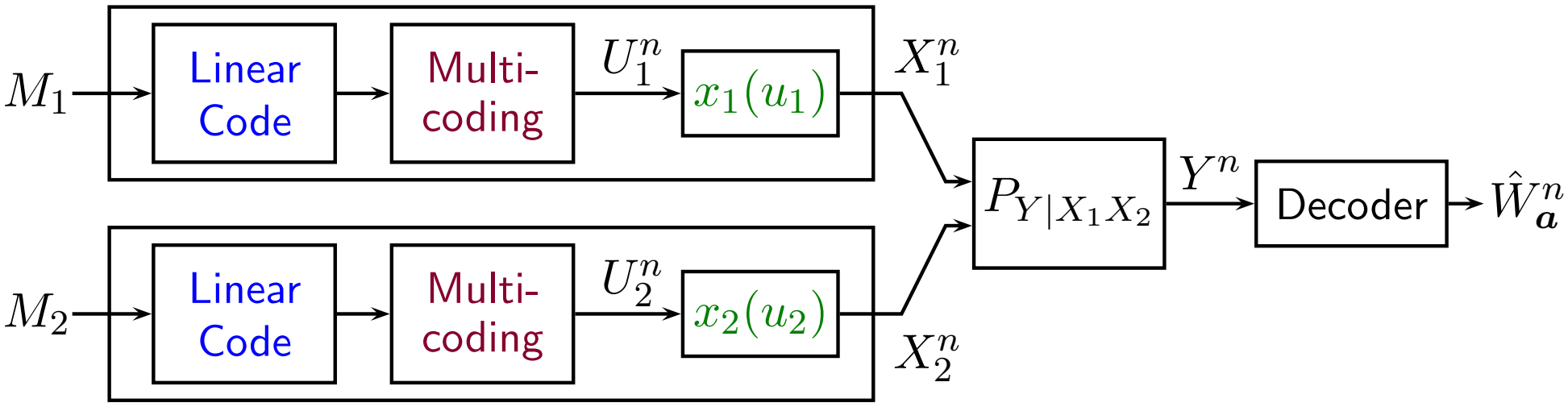
  $u_2^n(m_2, l_2) = \boldsymbol{\eta}(m_2, l_2)\mathsf{G} \oplus d_2^n$

Nested Linear Coding Architecture



Encoding:

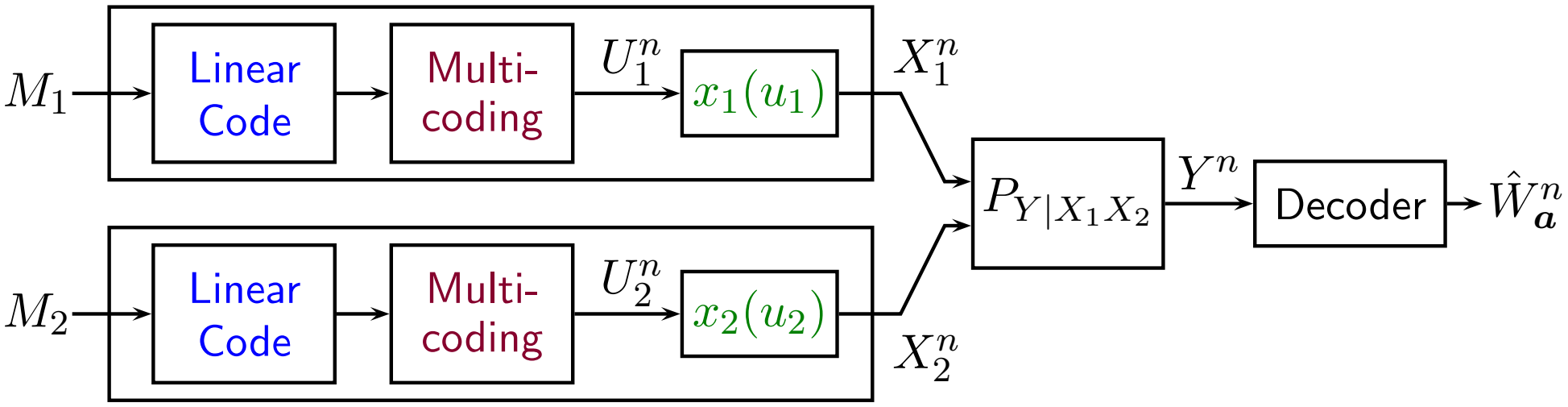
Nested Linear Coding Architecture



Encoding:

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

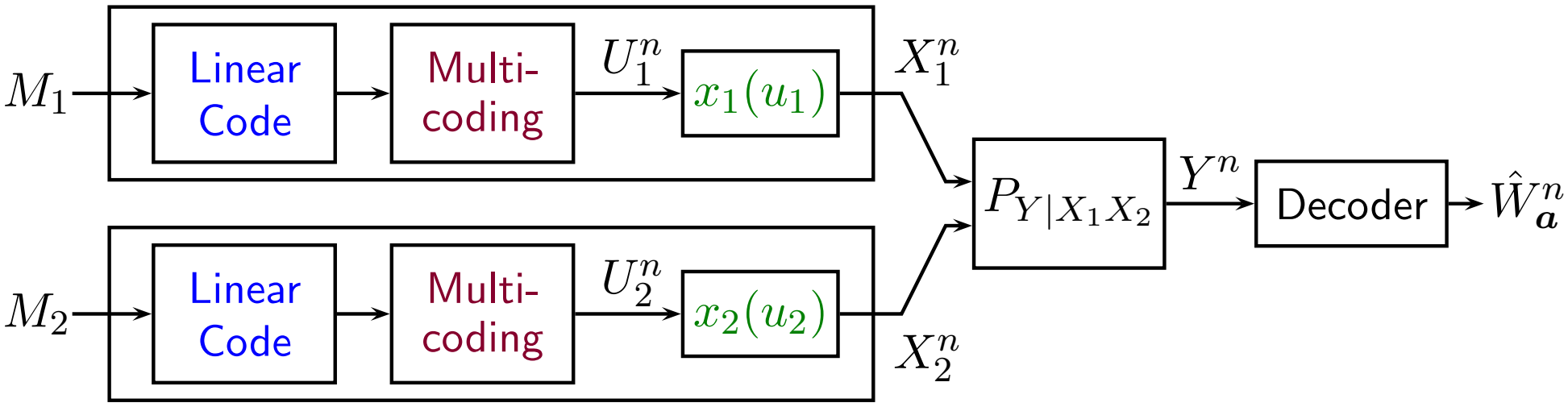
Nested Linear Coding Architecture



Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)

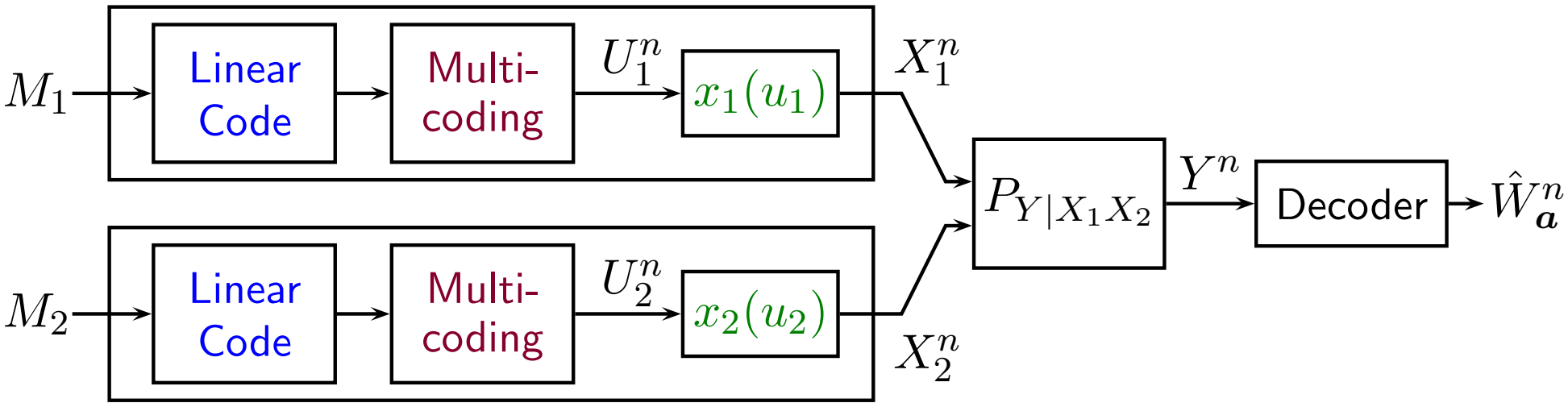
Nested Linear Coding Architecture



Encoding:

- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big), \;\; i = 1, \ldots, n.$

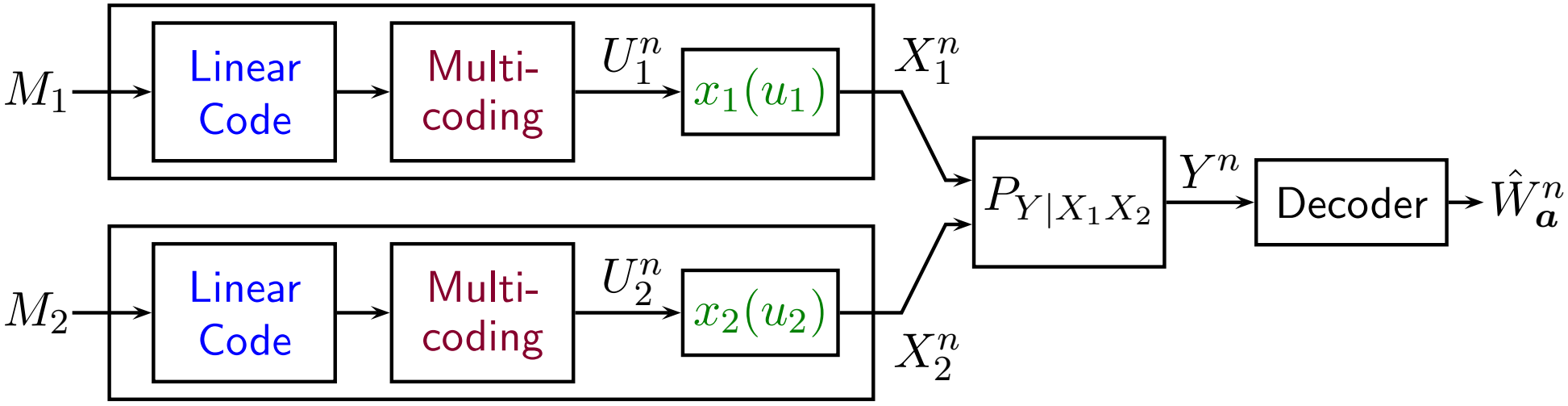
Nested Linear Coding Architecture



Encoding:

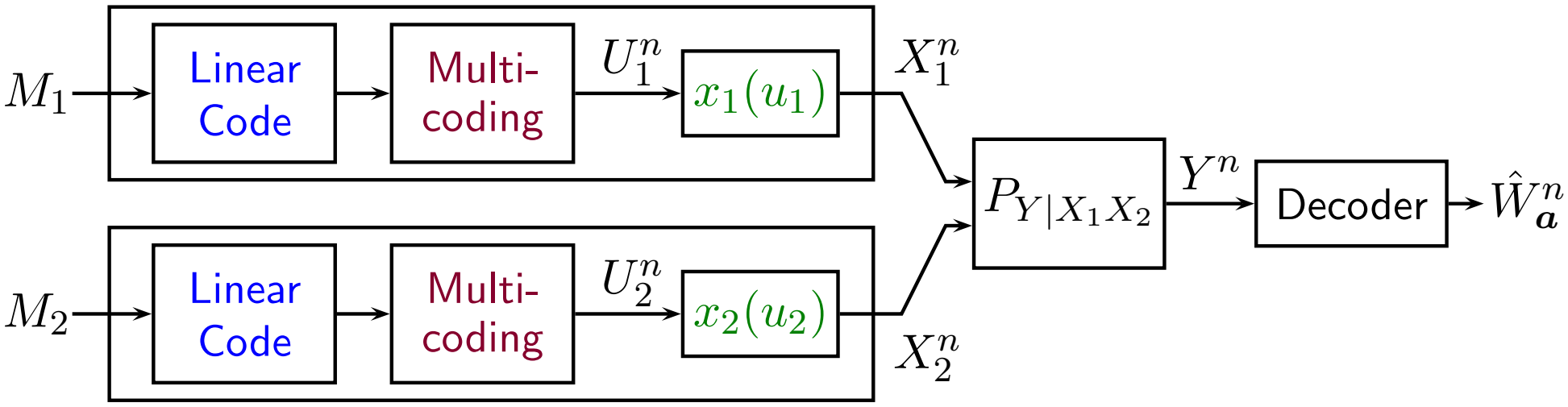
- Fix pmfs $p(u_1)$, $p(u_2)$ and mappings $x_1(u_1)$, and $x_2(u_2)$.
- Multicoding: For each m_k, find an index l_k such that $u_k^n(m_k, l_k) \in \mathcal{T}_{\epsilon'}^{(n)}(U_k)$. (If no such l_k, pick one randomly.)
- Transmit $x_{ki} = x_k\big(u_{ki}(m_k, l_k)\big), \;\; i = 1, \ldots, n.$

Nested Linear Coding Architecture



Computation Problem:

Nested Linear Coding Architecture



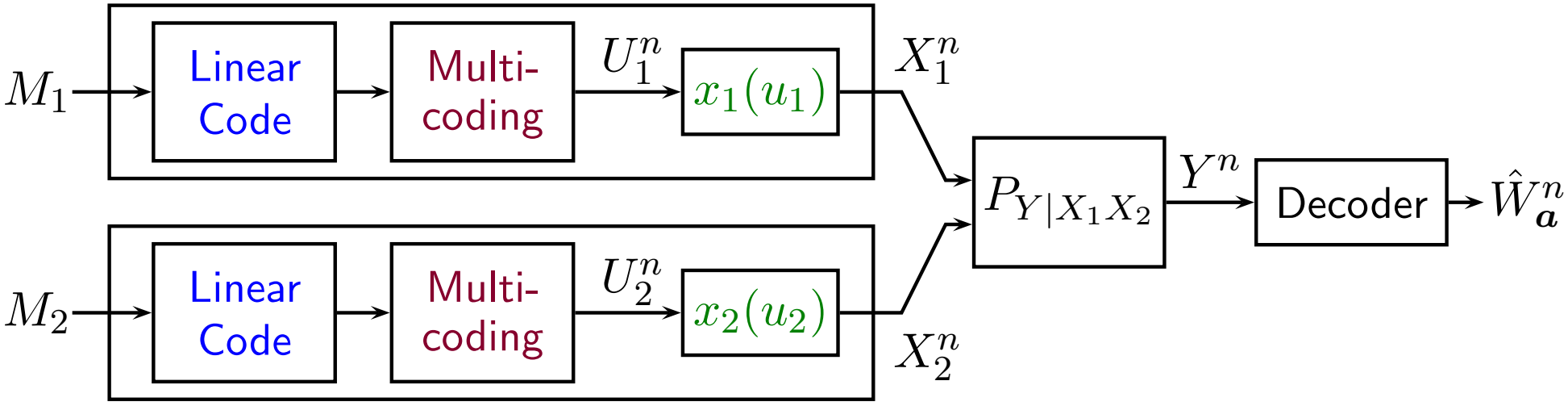
Computation Problem:

- For $m_k \in [2^{nR_k}]$, $l_k \in [2^{n\hat{R}_k}]$, we can express the linear combination of codewords as

$$\begin{aligned} w_{\boldsymbol{a}}^n &= a_1 u_1^n(m_1, l_1) \oplus a_2 u_2^n(m_2, l_2) \\ &= \left[a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\right] \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \\ &= \boldsymbol{\nu}(s_{\boldsymbol{a}}) \mathsf{G} \oplus a_1 d_1^n \oplus a_2 d_2^n \end{aligned}$$

where $s_{\boldsymbol{a}} \in \left[2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}\right]$.

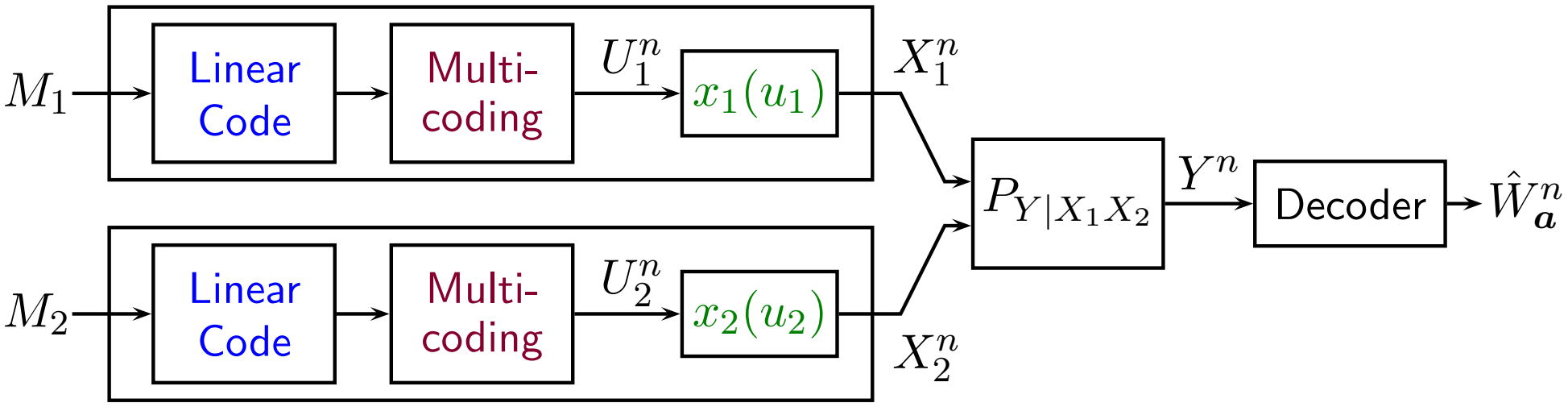
Nested Linear Coding Architecture



Decoding:

- Let $\epsilon' < \epsilon$.

Nested Linear Coding Architecture



Decoding:

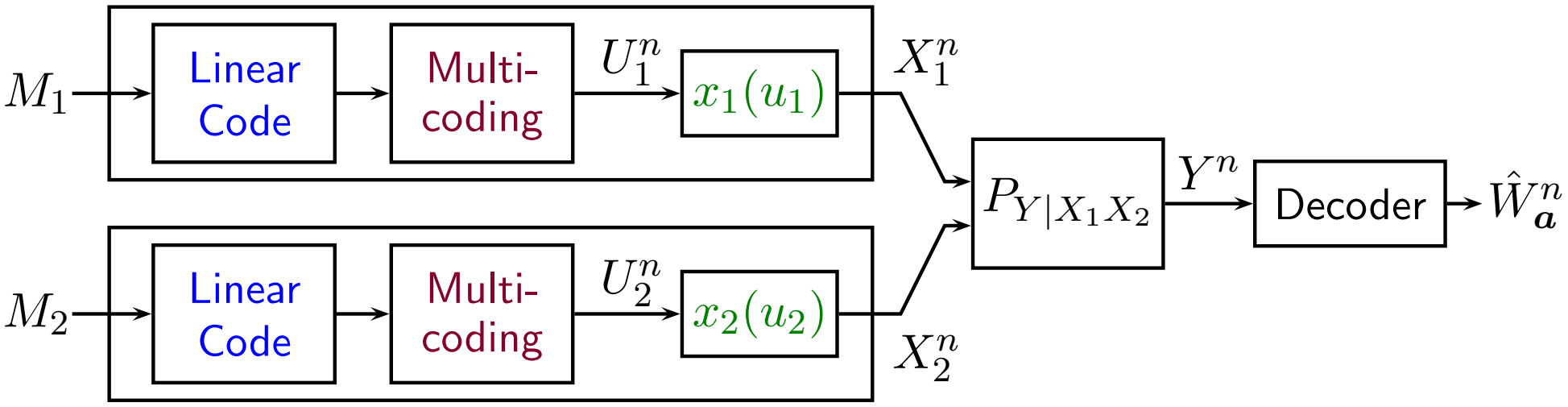
- Let $\epsilon' < \epsilon$.
- Search for a unique index $s_{\boldsymbol{a}} \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$ such that

$$(u_1^n(m_1, l_1), u_2^n(m_2, l_2), y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y),$$

for some $(m_1, l_1, m_2, l_2) \in [2^{nR_1}] \times [2^{n\hat{R}_1}] \times [2^{nR_2}] \times [2^{n\hat{R}_2}]$ such that

$$\boldsymbol{\nu}(s_{\boldsymbol{a}}) = a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2).$$

Nested Linear Coding Architecture



Decoding:

- Let $\epsilon' < \epsilon$.
- Search for a unique index $s_{\boldsymbol{a}} \in [2^{n \max\{R_1+\hat{R}_1, R_2+\hat{R}_2\}}]$ such that

$$(u_1^n(m_1, l_1), u_2^n(m_2, l_2), y^n) \in \mathcal{T}_\epsilon^{(n)}(U_1, U_2, Y),$$

for some $(m_1, l_1, m_2, l_2) \in [2^{nR_1}] \times [2^{n\hat{R}_1}] \times [2^{nR_2}] \times [2^{n\hat{R}_2}]$ such that

$$\boldsymbol{\nu}(s_{\boldsymbol{a}}) = a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2).$$

- If there is no such index, or more than one, the decoder declares an error.

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_{\epsilon}^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

Error Analysis

An error occurs only if one or more of the following events occur,

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_{\epsilon}^{(n)}\}.$$

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_{\epsilon}^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

Then, by the union of events bound,

$$\mathsf{P}\{\hat{W}_{\boldsymbol{a}}^n \neq W_{\boldsymbol{a}}^n\} \leq \mathsf{P}\{\mathcal{E}_1\} + \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} + \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\}.$$

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.
- Intuition: Searching for one of $\approx 2^{nH(U_k)}$ typical sequences out of $2^{n \log q}$ total sequences. Will succeed w.h.p. if $2^{n\hat{R}_k} > 2^{n(\log q - H(U_k))}$.

Error Analysis

- For some message, we cannot find a typical linear codeword:

$$\mathcal{E}_1 = \{U_k^n(m_k, l_k) \notin \mathcal{T}_{\epsilon'}^{(n)} \text{ for all } l_k, \text{ for some } m_k, k = 1, 2\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_1\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- $D(p_{U_k} \| p_{\mathsf{q}}) = \log \mathsf{q} - H(U_k)$.
- Intuition: Searching for one of $\approx 2^{nH(U_k)}$ typical sequences out of $2^{n \log q}$ total sequences. Will succeed w.h.p. if $2^{n\hat{R}_k} > 2^{n(\log q - H(U_k))}$.
- Proof just requires second moment method.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.
- Our proof handles these statistical dependencies by breaking up the possible error events according to the underlying rank of the selected linear codewords. (Markov Lemma for Nested Linear Codes.)

Error Analysis

- The channel inputs and output are not jointly typical:

$$\mathcal{E}_2 = \{(U_1^n(M_1, L_1), U_2^n(M_2, L_2), Y^n) \notin \mathcal{T}_\epsilon^{(n)}\}.$$

- If $\hat{R}_k > D(p_{U_k} \| p_{\mathsf{q}}) + \delta(\epsilon)$, then $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_2 \cap \mathcal{E}_1^c\} = 0$ where $\delta(\epsilon) \to 0$ as $\epsilon \to 0$.
- In a random i.i.d. coding proof, we would just use the fact that the codewords are independent and that the channel is memoryless.
- Here, the linear codewords can be statistically dependent, since the choices of the auxiliary indices L_k is coupled due to the shared nested linear codebook.
- Our proof handles these statistical dependencies by breaking up the possible error events according to the underlying rank of the selected linear codewords. (Markov Lemma for Nested Linear Codes.)
- Prior work by **Padakandla-Pradhan '13** developed a bound that also requires $\hat{R}_k < D(p_{U_k} \| p_{\mathsf{q}}) + 3\delta(\epsilon)$.

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

Error Analysis

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

- We upper bound this event in two ways.

Error Analysis

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

- We upper bound this event in two ways.
  1. "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\left\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\right\}$$

Error Analysis

- There are linear codewords that are jointly typical with the channel output and give the wrong linear combination:

$$\mathcal{E}_3 = \{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)} \text{ for some } (m_1, l_1, m_2, l_2) \text{ such that } \boldsymbol{\nu}(S_{\boldsymbol{a}}) \neq a_1 \boldsymbol{\eta}(m_1, l_1) \oplus a_2 \boldsymbol{\eta}(m_2, l_2)\}.$$

- We upper bound this event in two ways.
  1. "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

  2. "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1, l_1, m_2, l_2) \neq (M_1, L_1, M_2, L_2)\Big\}$$

Error Analysis: "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

Error Analysis: "Direct Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(W_{\boldsymbol{a}}^n(s_{\boldsymbol{a}}), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c,\ s_{\boldsymbol{a}} \neq S_{\boldsymbol{a}}\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$R_1 < I_{\mathsf{CF},1}(\boldsymbol{a}) \triangleq H(U_1) - H(W_{\boldsymbol{a}}|Y)$$
$$R_2 < I_{\mathsf{CF},2}(\boldsymbol{a}) \triangleq H(U_2) - H(W_{\boldsymbol{a}}|Y),$$

  which matches our intuition from earlier.

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1, l_1), U_2^n(m_2, l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1, l_1, m_2, l_2) \neq (M_1, L_1, M_2, L_2)\Big\}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \neq (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$\begin{aligned} R_1 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\},\\ R_2 &< I(X_2;Y|X_1),\\ R_1+R_2 &< I(X_1,X_2;Y) \end{aligned}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \neq (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

$$\begin{aligned} R_1 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\},\\ R_2 &< I(X_2;Y|X_1),\\ R_1+R_2 &< I(X_1,X_2;Y) \end{aligned}$$

OR

$$\begin{aligned} R_1 &< I(X_1;Y|X_2),\\ R_2 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},2}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},1}(\boldsymbol{b})\},\\ R_1+R_2 &< I(X_1,X_2;Y). \end{aligned}$$

Error Analysis: "Multiple-Access Decoding" Bound

$$\mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} \leq \mathsf{P}\Big\{(U_1^n(m_1,l_1), U_2^n(m_2,l_2), Y^n) \in \mathcal{T}_\epsilon^{(n)},\ \mathcal{E}_1^c \text{ for some } (m_1,l_1,m_2,l_2) \neq (M_1,L_1,M_2,L_2)\Big\}$$

- Can show that $\lim_{n\to\infty} \mathsf{P}\{\mathcal{E}_3 \cap \mathcal{E}_1^c\} = 0$ if

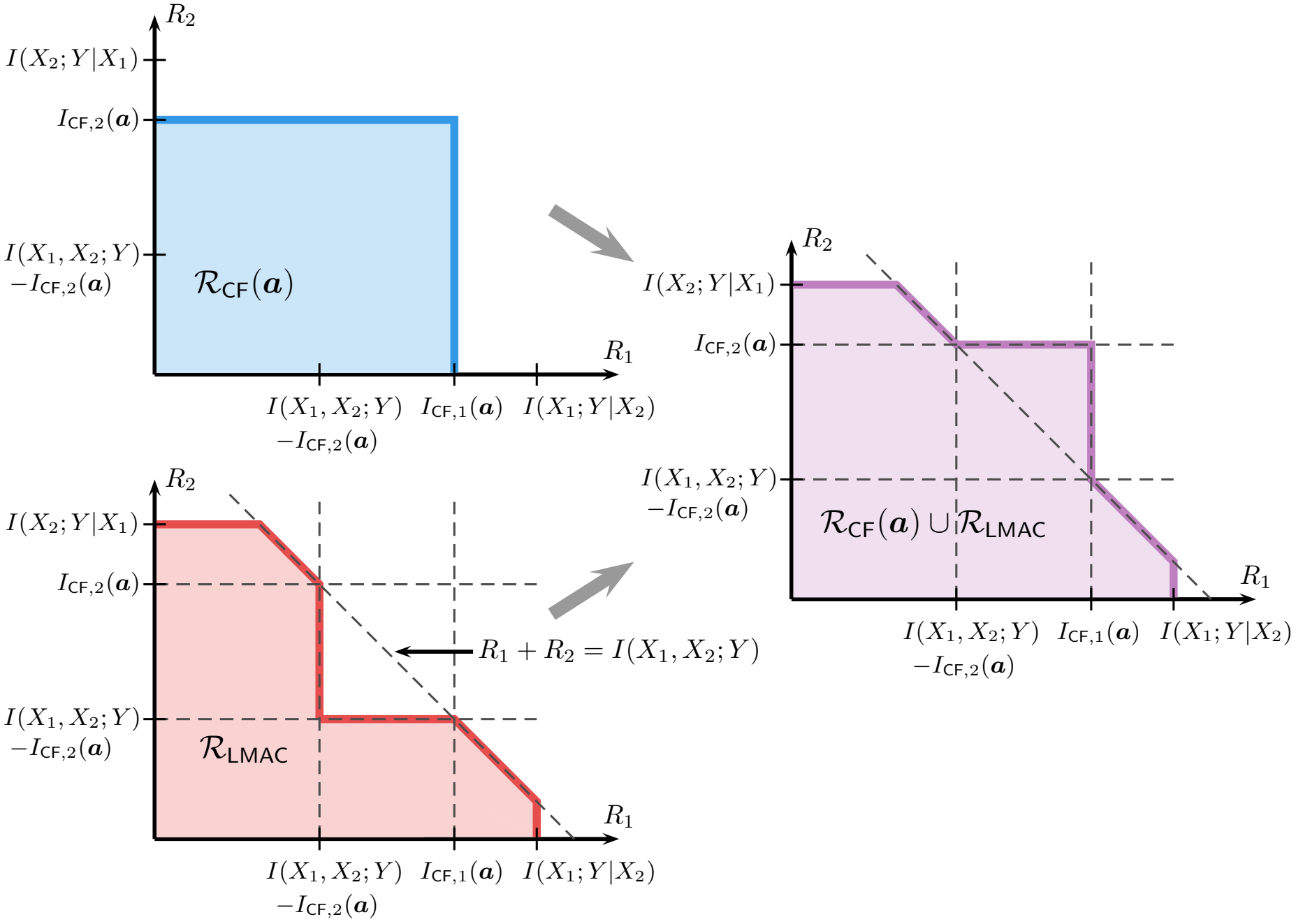
$$\begin{aligned} R_1 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},1}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},2}(\boldsymbol{b})\},\\ R_2 &< I(X_2;Y|X_1),\\ R_1+R_2 &< I(X_1,X_2;Y) \end{aligned}$$

OR

$$\begin{aligned} R_1 &< I(X_1;Y|X_2),\\ R_2 &< \max_{\boldsymbol{b}\in\mathbb{A}^2\setminus\{\mathbf{0}\}} \min\{I_{\mathsf{CF},2}(\boldsymbol{b}), I(X_1,X_2;Y) - I_{\mathsf{CF},1}(\boldsymbol{b})\},\\ R_1+R_2 &< I(X_1,X_2;Y). \end{aligned}$$

- The $I_{\mathsf{CF},2}(\boldsymbol{b})$ term plays a key role in handling the dependencies between competing pairs of linear codewords.

Rate Region


R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a})$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$
R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{CF}}(\boldsymbol{a}) \cup \mathcal{R}_{\mathsf{LMAC}}$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$
R_2
$I(X_2;Y|X_1)$
$I_{\mathsf{CF},2}(\boldsymbol{a})$
$R_1 + R_2 = I(X_1,X_2;Y)$
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$\mathcal{R}_{\mathsf{LMAC}}$
R_1
$I(X_1,X_2;Y)$
$-I_{\mathsf{CF},2}(\boldsymbol{a})$
$I_{\mathsf{CF},1}(\boldsymbol{a})$
$I(X_1;Y|X_2)$

Compute-and-Forward over a Gaussian MAC

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

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

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

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,

$$W^n_{\boldsymbol{a}_1} = \bigoplus_{k=1}^{K} a_{1,k} U^n_k(M_k, L_k)$$
$$W^n_{\boldsymbol{a}_2} = \bigoplus_{k=1}^{K} a_{2,k} u^n_k(M_k, L_k) \ ,$$

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,

$$W^n_{\boldsymbol{a}_1} = \bigoplus_{k=1}^{K} a_{1,k} U^n_k(M_k, L_k)$$
$$W^n_{\boldsymbol{a}_2} = \bigoplus_{k=1}^{K} a_{2,k} u^n_k(M_k, L_k) \ ,$$

  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

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

$$R_k < \max_{\boldsymbol{b} \in \mathbb{F}_q^2 \setminus \{\mathbf{0}\}} \min\{H(U_k) - H(V_{\boldsymbol{b}}|Y), H(U_k) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}|Y, V_{\boldsymbol{b}})\}$$

$$R_j < I(W_{\boldsymbol{a}_2}; Y, W_{\boldsymbol{a}_1}) - H(W_{\boldsymbol{a}_2}) + H(U_j)$$

$$R_k + R_j < I(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}; Y) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}) + H(U_k) + H(U_j)$$

or

$$R_k < I(W_{\boldsymbol{a}_1}; Y, W_{\boldsymbol{a}_2}) - H(W_{\boldsymbol{a}_1}) + H(U_k)$$

$$R_j < \max_{\boldsymbol{b} \in \mathbb{F}_q^2 \setminus \{\mathbf{0}\}} \min\{H(U_j) - H(V_{\boldsymbol{b}}|Y), H(U_j) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}|Y, V_{\boldsymbol{b}})\}$$

$$R_k + R_j < I(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}; Y) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}) + H(U_k) + H(U_j)$$

for all j, k such that $a_{1,k} \neq 0$, $a_{2,j} \neq 0$ for some input pmf $\prod_{k=1}^{K} p(u_k)$, mappings $x_k(u_k)$, and $V_{\boldsymbol{b}} = b_1 W_{\boldsymbol{a}_1} \oplus b_2 W_{\boldsymbol{a}_2}$.

Jointly Decoding Two Linear Combinations of K Codewords

Theorem

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

$$R_k < \max_{\boldsymbol{b} \in \mathbb{F}_q^2 \setminus \{\mathbf{0}\}} \min\{H(U_k) - H(V_{\boldsymbol{b}}|Y), H(U_k) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}|Y, V_{\boldsymbol{b}})\}$$

$$R_j < I(W_{\boldsymbol{a}_2}; Y, W_{\boldsymbol{a}_1}) - H(W_{\boldsymbol{a}_2}) + H(U_j)$$

$$R_k + R_j < I(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}; Y) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}) + H(U_k) + H(U_j)$$

or

$$R_k < I(W_{\boldsymbol{a}_1}; Y, W_{\boldsymbol{a}_2}) - H(W_{\boldsymbol{a}_1}) + H(U_k)$$

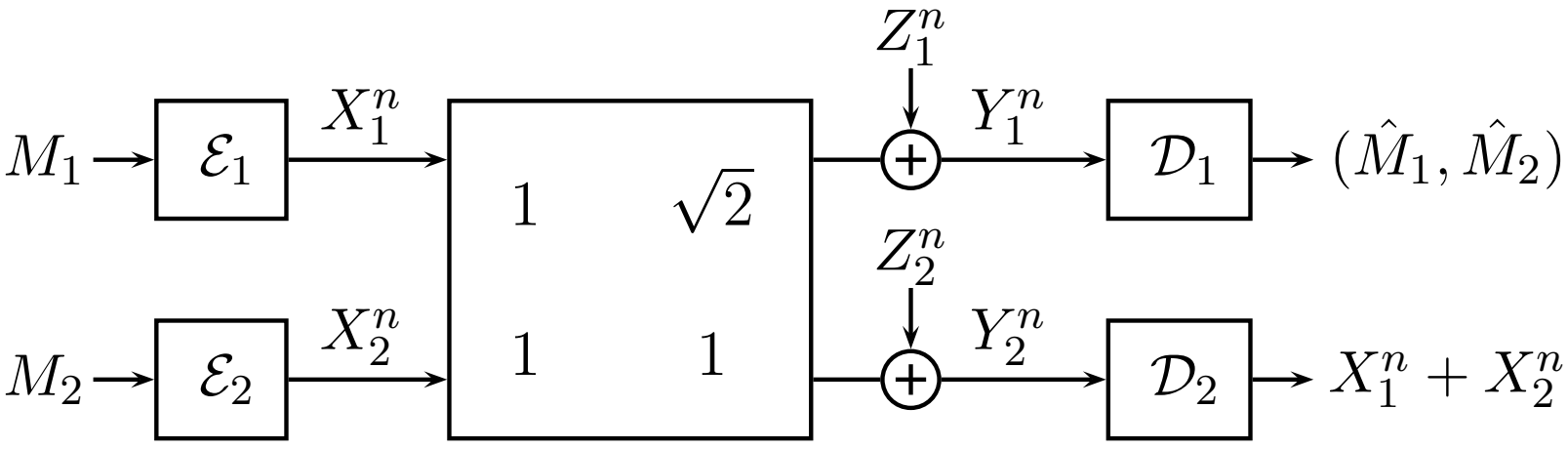
$$R_j < \max_{\boldsymbol{b} \in \mathbb{F}_q^2 \setminus \{\mathbf{0}\}} \min\{H(U_j) - H(V_{\boldsymbol{b}}|Y), H(U_j) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}|Y, V_{\boldsymbol{b}})\}$$

$$R_k + R_j < I(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}; Y) - H(W_{\boldsymbol{a}_1}, W_{\boldsymbol{a}_2}) + H(U_k) + H(U_j)$$

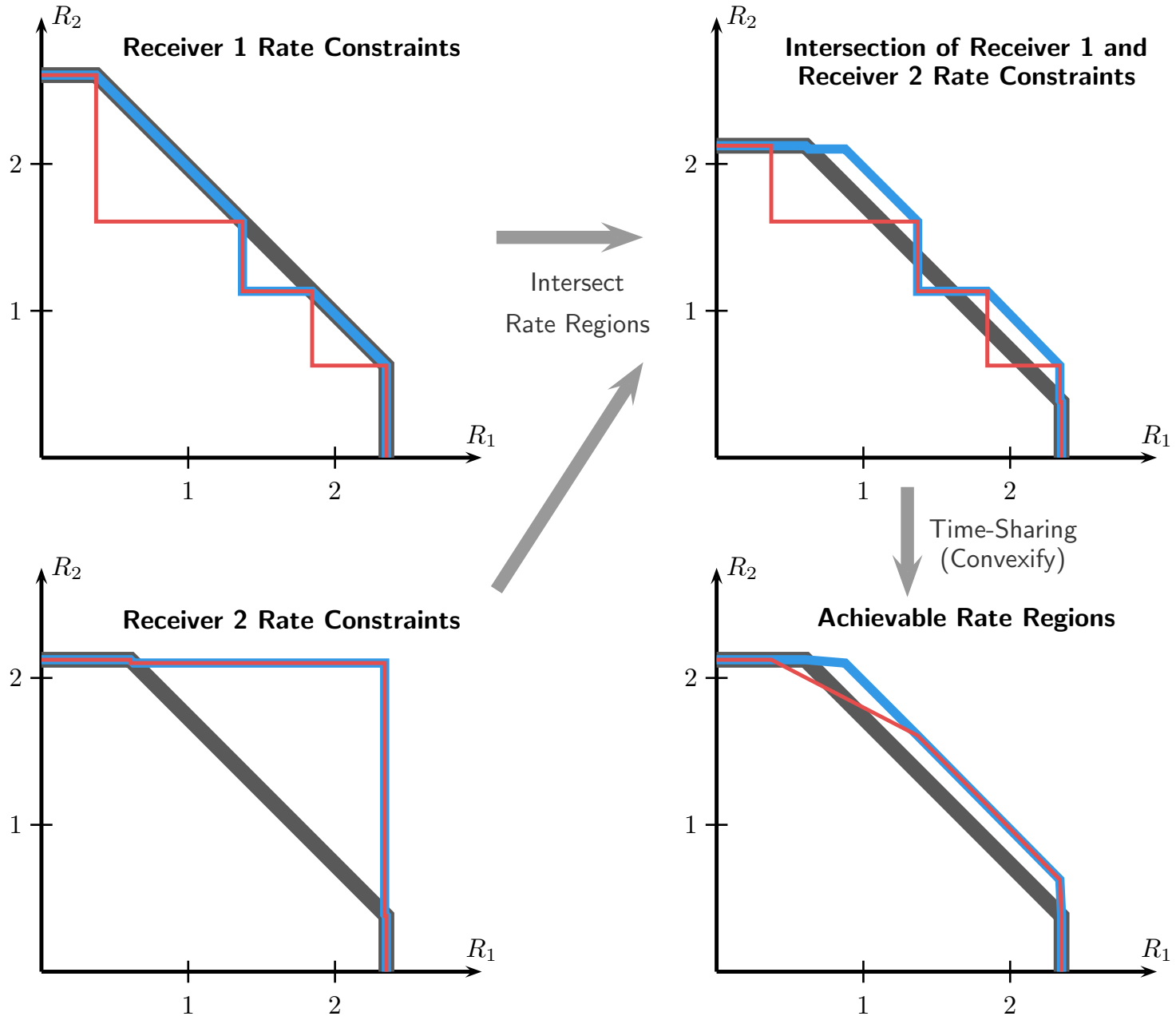
for all j, k such that $a_{1,k} \neq 0$, $a_{2,j} \neq 0$ for some input pmf $\prod_{k=1}^{K} p(u_k)$, mappings $x_k(u_k)$, and $V_{\boldsymbol{b}} = b_1 W_{\boldsymbol{a}_1} \oplus b_2 W_{\boldsymbol{a}_2}$.

- The auxiliary linear combination $V_{\boldsymbol{b}}$ plays a key role in classifying dependent competing pairs in the error analysis.

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.