

Computationally Efficient Codes for Adversarial Binary-Erasure Channels

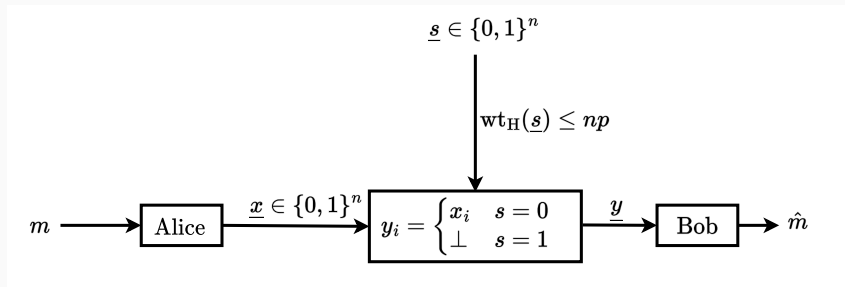
ISIT 2023 (Taipei, Taiwan)

¹Sijie Li ²Prasad Krishnan ³Sidharth Jaggi ⁴Michael Langberg ⁵Anand D. Sarwate

26 June 2023

¹UT at Austin ²IIT Hyderabad ³University of Bristol ⁴University at Buffalo ⁵Rutgers University

Our good friend the erasure channel

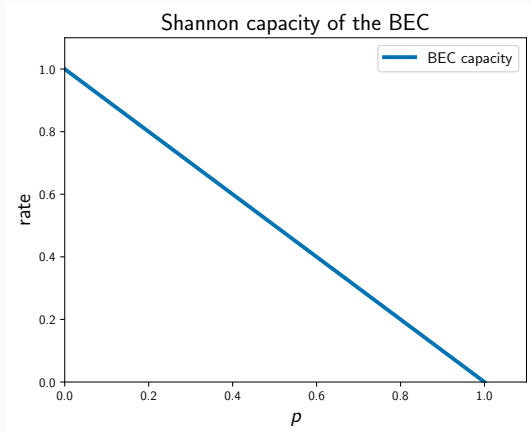


The erasure channel almost needs no introduction...

- Binary input, ternary output.
- Think of erasures as a state sequence \underline{s} where 1 means “erase.”
- Fraction of erased bits upper bounded by p , either exactly (coding theory) or with high probability (Shannon theory).

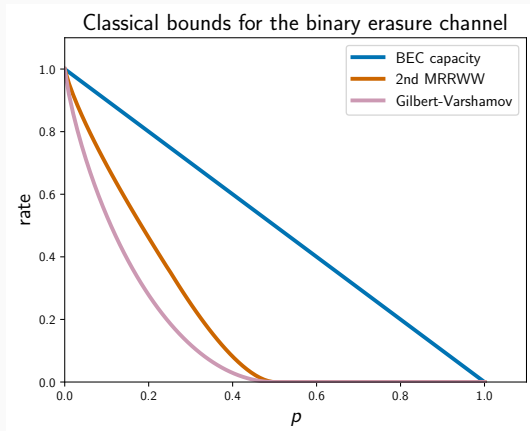
Optimism versus pessimism

Optimism versus pessimism



Optimistic: *Random* erasures (BEC):
erasures are i.i.d. Bernoulli. Shannon
capacity is $1 - p$.

Optimism versus pessimism



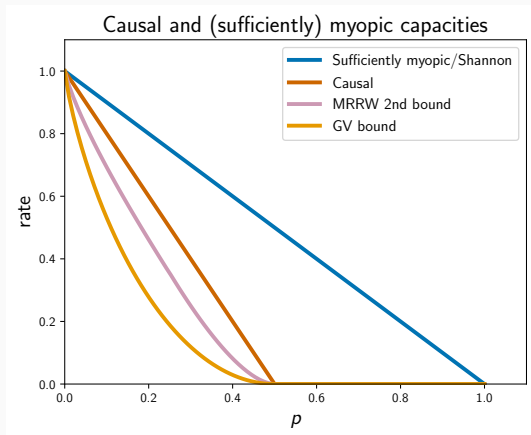
Optimistic: *Random* erasures (BEC): erasures are i.i.d. Bernoulli. Shannon capacity is $1 - p$.

Pessimistic: *Adversarial* erasures: erasures can depend on transmitted codeword.

Capacity unknown!

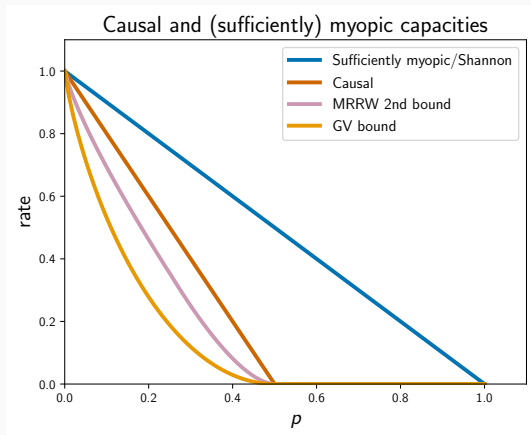
- Lower bound: Gilbert-Varshamov (and **linear codes** work.)
- Upper bound from linear programming (MRRW “LP” bound).

Models in the middle: causal and myopic



(Chen, Jaggi, Langberg 2015) (Sarwate 2010) (Dey et al. 2016) (Dey et al. 2019)

Models in the middle: causal and myopic

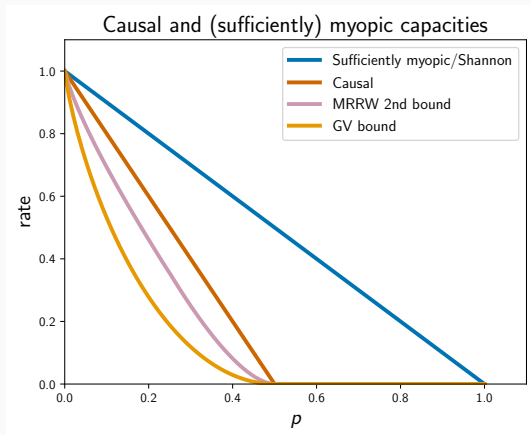


Causal adversaries: erasures can depend on the current and past input only.

Capacity is $1 - 2p$.

(Chen, Jaggi, Langberg 2015) (Sarwate 2010) (Dey et al. 2016) (Dey et al. 2019)

Models in the middle: causal and myopic



Causal adversaries: erasures can depend on the current and past input only.

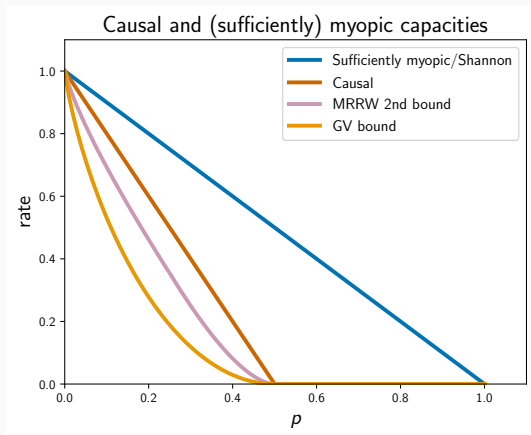
Capacity is $1 - 2p$.

Myopic adversaries: erasures can depend on a noisy (erased by $\text{BEC}(q)$) version of the codeword.

- If $p < q$, capacity is $1 - p$.
- If $p > q$... see the paper.

(Chen, Jaggi, Langberg 2015) (Sarwate 2010) (Dey et al. 2016) (Dey et al. 2019)

Models in the middle: causal and myopic



Causal adversaries: erasures can depend on the current and past input only.

Capacity is $1 - 2p$.

Myopic adversaries: erasures can depend on a noisy (erased by $\text{BEC}(q)$) version of the codeword.

- If $p < q$, capacity is $1 - p$.
- If $p > q$... see the paper.

Achievability arguments use **stochastic encoding** and **list decoding** with **nonlinear codes**.

(Chen, Jaggi, Langberg 2015) (Sarwate 2010) (Dey et al. 2016) (Dey et al. 2019)

Can we design efficient codes for causal and myopic models?

By **efficient** we mean that they take **polynomial time** to encode, decode, and store.

Can we design efficient codes for causal and myopic models?

By **efficient** we mean that they take **polynomial time** to encode, decode, and store.

- **random codes** are inefficient to decode but **linear codes** are too easy jam!
→ use a **library of linear codebooks**.

Can we design efficient codes for causal and myopic models?

By **efficient** we mean that they take **polynomial time** to encode, decode, and store.

- **random codes** are inefficient to decode but **linear codes** are too easy jam!
→ use a **library of linear codebooks**.
- **common randomness** is unrealistic.
→ use **limited encoder randomization** to **confuse the adversary**.

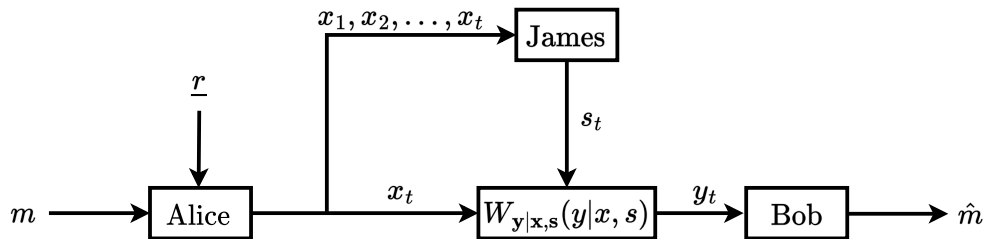
Can we design efficient codes for causal and myopic models?

By **efficient** we mean that they take **polynomial time** to encode, decode, and store.

- **random codes** are inefficient to decode but **linear codes** are too easy jam!
→ use a **library of linear codebooks**.
- **common randomness** is unrealistic.
→ use **limited encoder randomization** to **confuse the adversary**.
- **minimum distance coding** is not efficient in general.
→ use **list decoding** to permit **efficient decoding**.

Causal and Myopic Adversaries

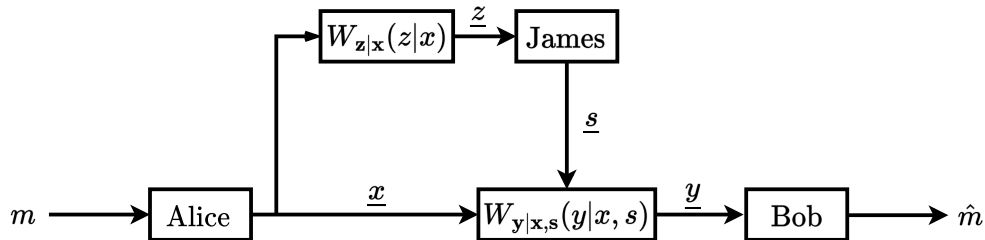
Causal/Online Adversaries



A **causal adversary** can eavesdrop **noiselessly and in real time** on the channel inputs:

- Decision on whether to erase at time t can depend on (x_1, x_2, \dots, x_t) .
- **Adversary's budget:** at most np erasures in codeword of length n .

Myopic adversaries



A **myopic adversary** can eavesdrop **noisily and noncausally** on the channel inputs:

- Decision on whether to erase at time t can depend on \underline{z} formed by passing \underline{x} through a BEC with erasure probability q
- **Adversary's budget:** at most np erasures in codeword of length n .

“Efficient” coding schemes

To get **polynomial complexity**, use

- **a small amount of randomization** to select from a
- **library of random linear codes** and
- uses **list decoding** to reduce the search space

There are different types of complexity we would like to control:

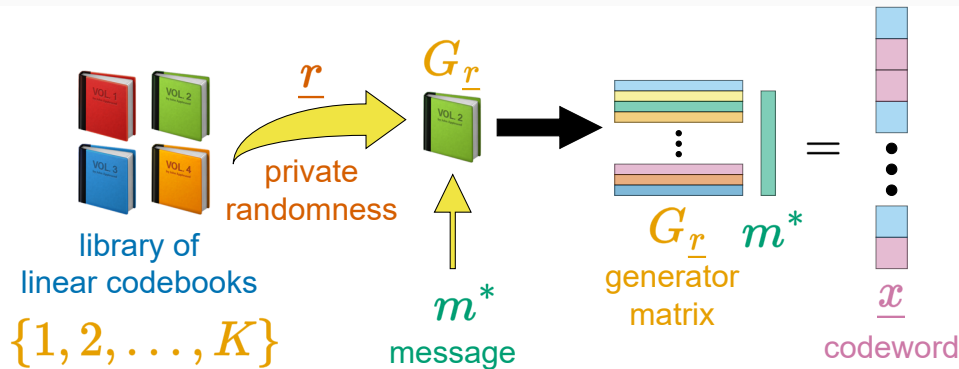
- **Design**: how many bits do we need to generate the code?
- **Storage**: how many bits do we need to store the code?
- **Encoding**: how many operations are needed to encode a message?
- **Decoding**: how many operations are needed to decode the message?

Main results

Model rate	Randomness	Enc/Storage	Decoding	P_{error}
Myopic $p < q$ $1 - p - \epsilon$	$\lambda_{SM} \log(n)$	$O(n^{2+\lambda_{SM}})$	$O(n^{3+\lambda_{SM}})$	$O(n^{-\lambda_{SM}})$
Myopic $q < q$ small rate	$O(n \log \log n)$	$O(n^2 \log \log n)$	$O(n^3 \log \log n)$	$O(n^{-4/5})$
Causal $1 - 2p - \epsilon$	$O\left(\frac{\gamma \log n}{\epsilon}\right)$	$O(n^3 \log \log n)$	$O(n^{32/\epsilon})$	$O(n^{-(\gamma-1)})$

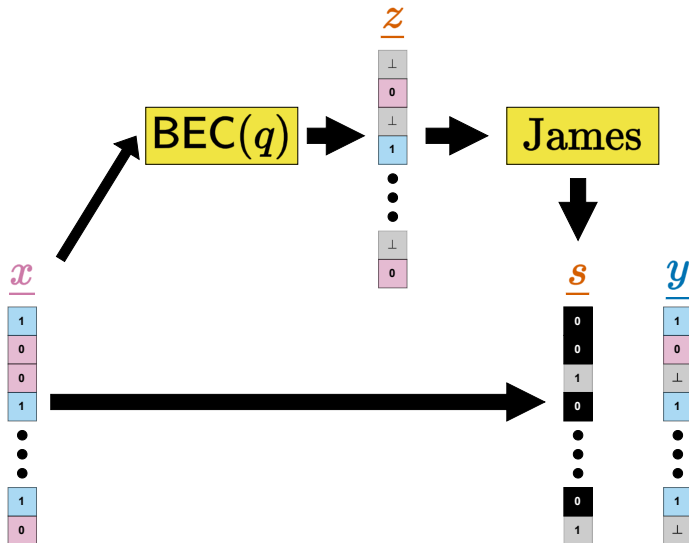
Sufficiently myopic adversaries

Encoding uses a library of linear codebooks

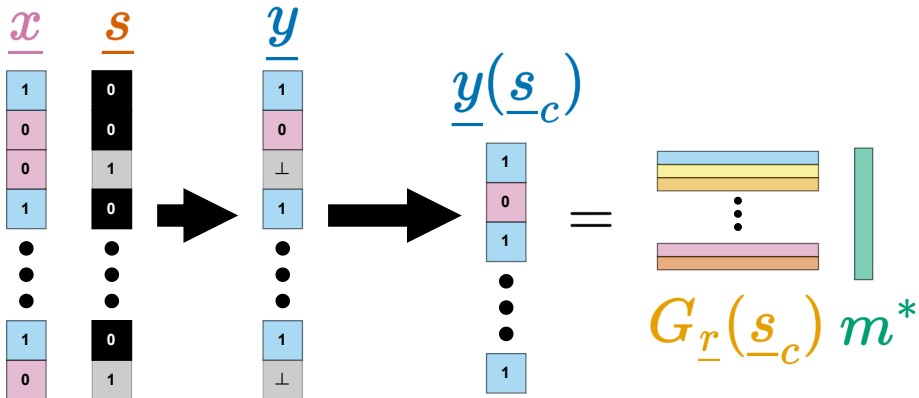


Generating random linear codes: $K = 2^{n\lambda_{SM}}$ generator matrices $G_i \in \mathbb{F}_2^{n \times nR}$ generated i.i.d. Bernoulli(1/2).

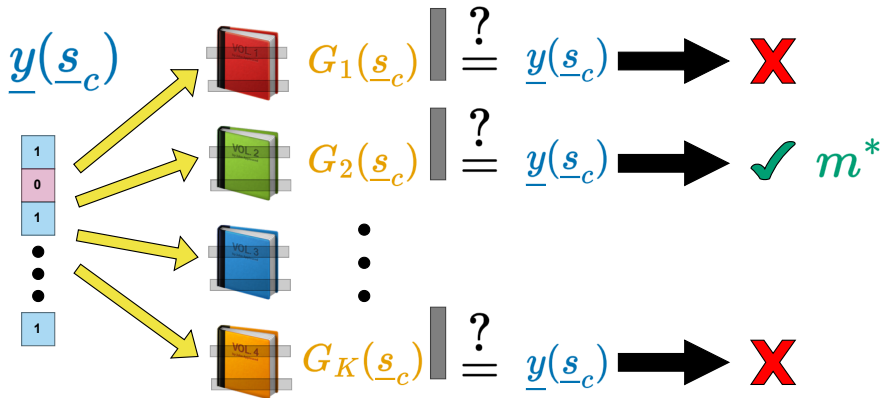
James sees an erased version of the codeword



Look at “unerased” rows of codebook

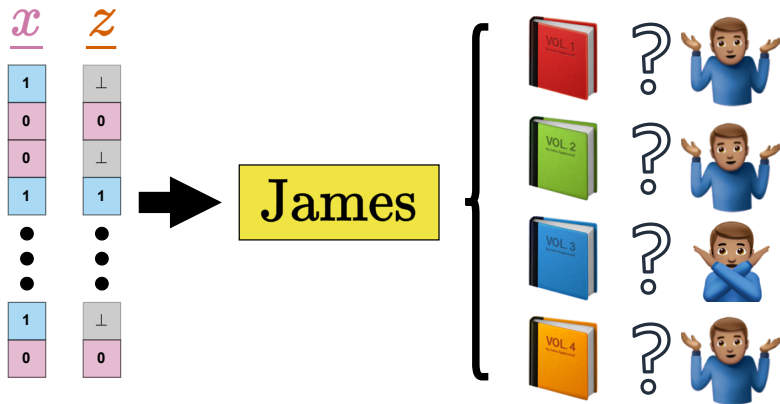


Decoder just tries every codebook



Complexity: n^3 per codebook, $K = n^{n\lambda_{SM}}$ codebooks.

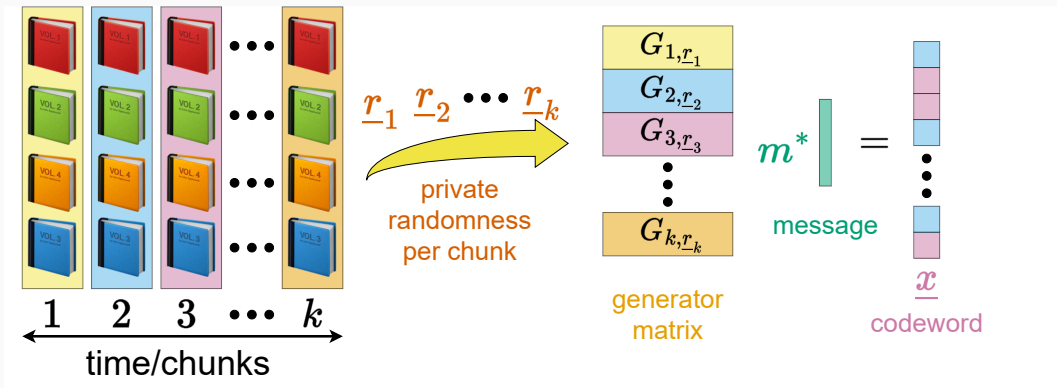
Myopia helps: if $q > p$, James cannot guess the correct codebook



$$P_{\text{error}} = O\left(\mathbf{n}^{-\lambda \mathbf{SM}}\right)$$

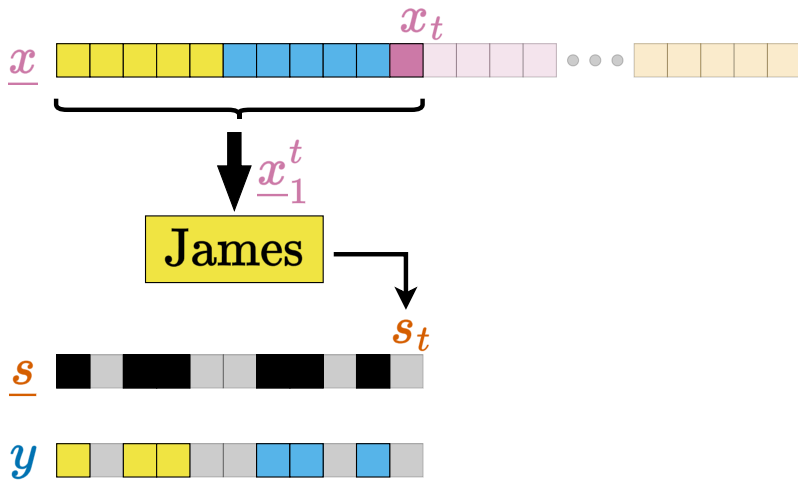
Causal adversaries

Encode splits block into a constant $k = \lceil \frac{n}{\epsilon} \rceil$ chunks

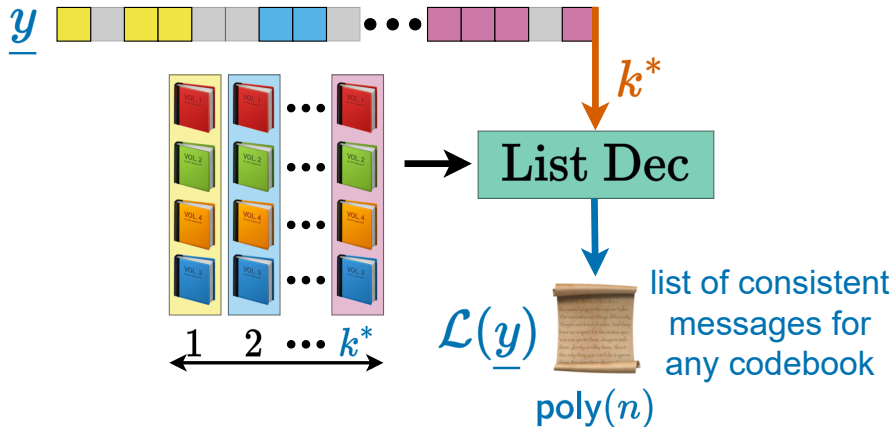


Generate a library of linear codebooks independently for each chunk.

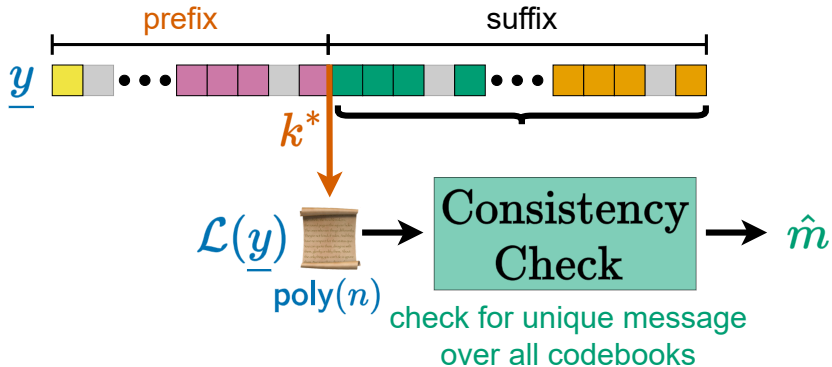
James can erase with causal information only



Bob decodes to a polynomial list after a certain time



Bob uses suffix to disambiguate the list



Why does this work?

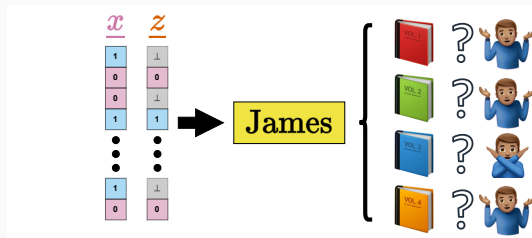
1. Bob can **track James's erasure budget**.
2. List decoding creates **a smaller set of messages** to check for consistency.
3. James has a choice to **make the list larger** (erase more earlier, less later) or **conserve his budget** (erase less earlier, more later).
4. **Poor James, he can't win.**

Recap and next steps

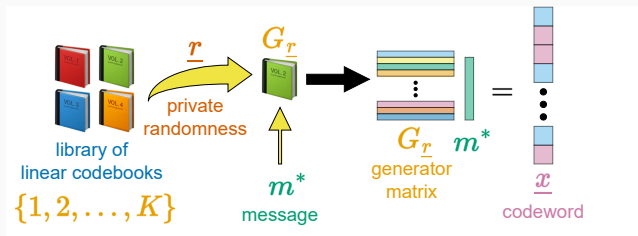
We design efficient (polynomial time) codes for both causal and myopic models.

- Use libraries of linear codebooks for efficient decoding.
- Use limited encoder randomization to confuse the adversary.
- Use list decoding to permit efficient decoding.

Open questions and future directions



- Other adversary structures?
- Better degree for “polynomial”?
- Better error guarantees?
- General AVC models?



Thank you!