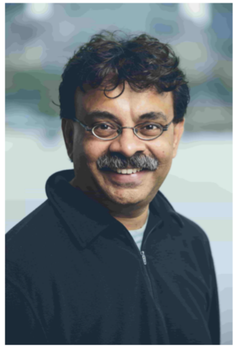
How to Compress Hidden Markov Sources

Preetum Nakkiran
Harvard University

Joint works with:
Venkatesan Guruswami, Madhu Sudan + Jarosław Błasiok, Atri Rudra

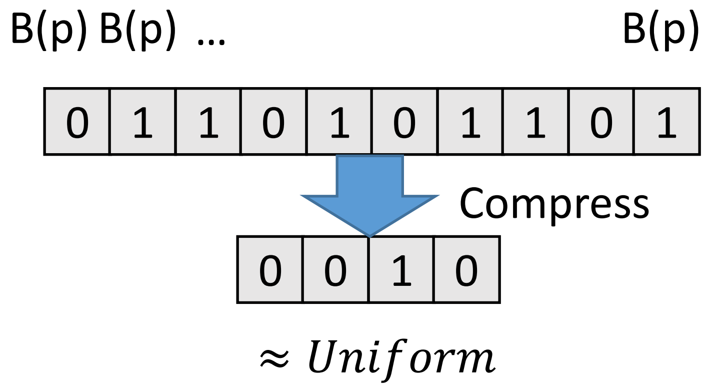



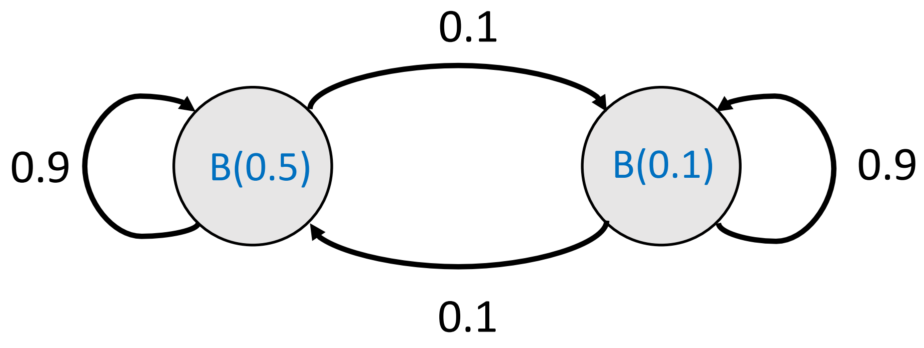
Compression

- **Problem:** Given n symbols from a probabilistic source, compress down to $< n$ symbols (ideally to "entropy" of the source)

  (s.t. decompression succeeds with high probability)

- Sources: Usually iid. This talk: Hidden-Markov Model





(Symbol alphabet can be arbitrary)

Organization

1. Goal: Compressing Symbols
    - What/why
2. Polarization & Polar Codes (for iid sources)
3. Polar codes for Markov Sources

Compression: Main Questions

For a source distribution on $(X_1, X_2, \dots, X_n)$:

1. How much can we compress?
   - [Shannon '48]: Down to the **entropy** $H(X_1, X_2, \dots X_n)$ [non-explicit]

     E.g. for iid Bernoulli(p): entropy = $nH(p)$.
2. Efficiency?
   - Efficiency of algorithms: compression/decompression (n)
   - **Efficiency of code:** Quickly approach the entropy rate

     $n\ symbols \mapsto nH(p) + n^{1-\delta}\ \ symbols$ vs. $n\ symbols \mapsto nH(p) + o(n)$

     Achieves within ϵ of entropy rate ($n\ symbols \mapsto n[H(p) + \epsilon]$) at blocklength $n \geq poly(\frac{1}{\epsilon})$
3. Linearity?
   - Useful for channel coding (as we will see)

Our Scheme: Compressing HMM sources

Compression/decompression algorithms which, given the HMM source, achieve:

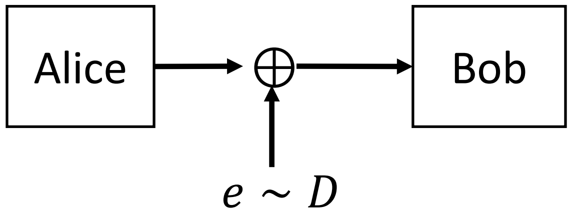
1. Poly-time compression/decompression
2. Linear
3. Rapidly approach entropy rate: For $X^n := (X_1, X_2, \dots X_n)$ from source

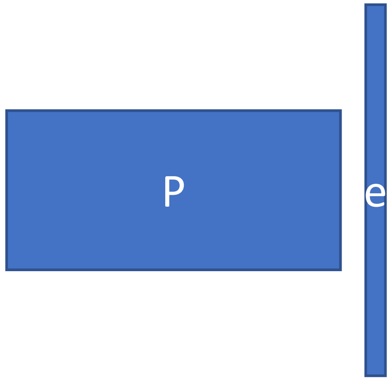
$$n \text{ symbols} \mapsto H(X^n) + \tau^{O(1)} \cdot n^{1-\delta} \text{ symbols}$$ (for HMM with mixing time τ)

- Previously unknown how to achieve all 3 above.
  Non-explicit: $n \mapsto H(X^n) + \sqrt{n}$
  [Lempel-Ziv]: $n \mapsto H(X^n) + o(n)$. Nonlinear. But, works for unknown HMM.
- Our result: Enabled by **Polar Codes**

Detour: Compression ⇒ Error-Correction

- Given a source D, corresponding Additive Channel:
  Alice sends $\boldsymbol{x} \in \mathbb{F}_q^n$
  Bob receives $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$ for $e = (e_1, e_2, \ldots e_n) \sim D$
- **Linear compression** scheme for $e \sim D$ ⇒ Linear error-correcting code for D-channel:
  - Let $P: \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$ be compression matrix. **Pe** can be decoded to **e** whp when $e \sim D$
  - Alice encodes into **nullspace(P):** $\boldsymbol{x} \in \boldsymbol{Null}(\boldsymbol{P})$
    - Bob receives $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{e}$
    - Bob computes $\boldsymbol{Py} = \boldsymbol{Px} + \boldsymbol{Pe} = \boldsymbol{Pe}$, and recovers the error **e**

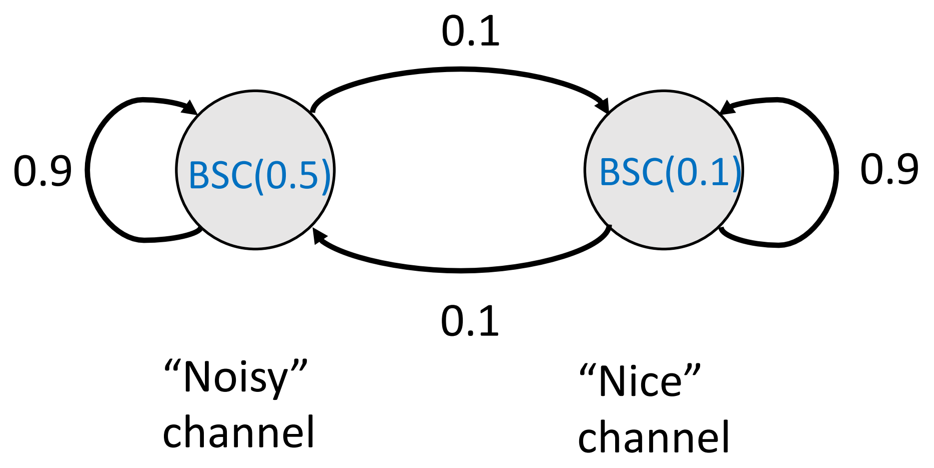




Efficiency: compression which rapidly approaches entropy rate
⇒ code which rapidly approaches capacity

Application: Correcting Markovian Errors

- Our result yields efficient error-correcting codes for Markovian errors.
- Eg: Channel has two states, “noisy” and “nice”, and transitions between them.



Remainder of this talk

- Focus on compressing Hidden-Markov Sources
- For simplicity, alphabet = $\mathbb{F}_2$

The plan:

1. Polar codes for compressing iid Bernoulli(p) bits.
2. Reduce HMM to iid case

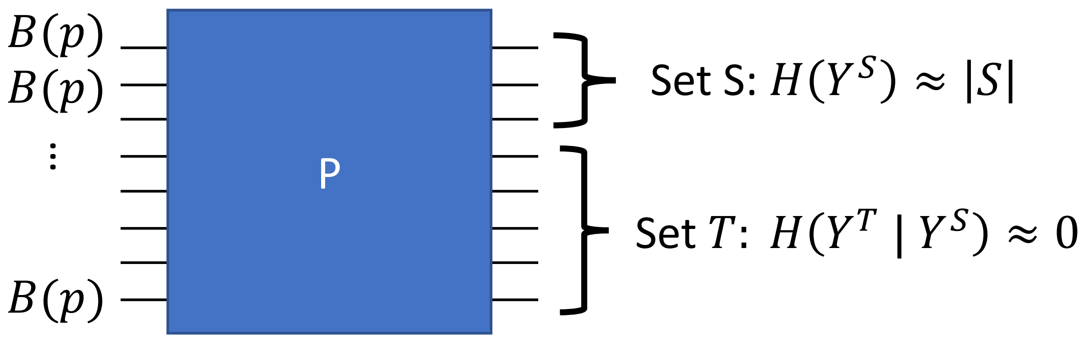
Polar Codes

- Linear compression / error-correcting codes
- Introduced by [Arikan ‘08], efficiency first analyzed in [Guruswami-Xia ’13], extended in [BG**N**RS ’18]
- Efficiency: First error-correcting codes to ``achieve capacity at polynomial blocklengths’’: within ϵ of capacity at blocklengths $n \geq poly(\frac{1}{\epsilon})$
- **Simple, elegant, purely information-theoretic construction**

Compression via Polarization

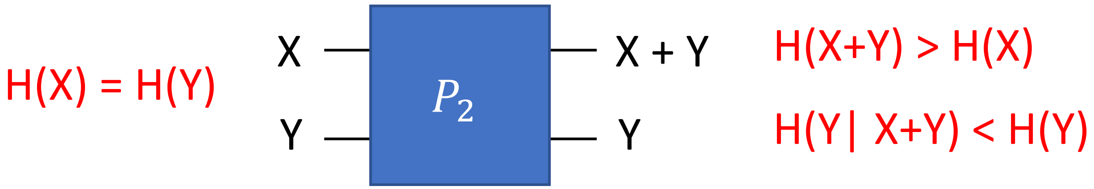
- **Goal:** Compress **n** iid Bernoulli(p) bits
- Polarization ⇒ Compression:
  - Suppose we have invertible transform P such that, on input $B(p)^n$, first block of outputs (set S) have ≈ full entropy

  - **Compression:** Output Y^S.
  - **Decompression:** Since $H(Y^T \mid Y^S) \approx 0$, can guess Y^T whp, then invert P to decompress.

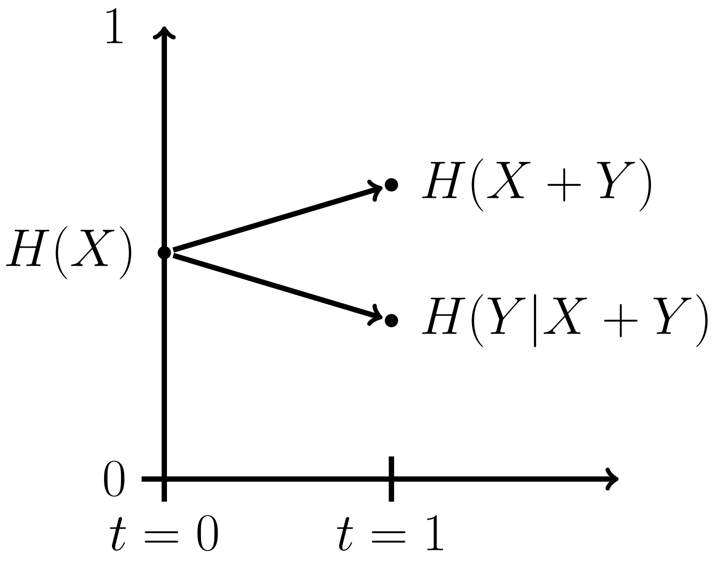


Polar Transform

- The following 2x2 transform over $\mathbb{F}_2$ "polarizes" entropies:

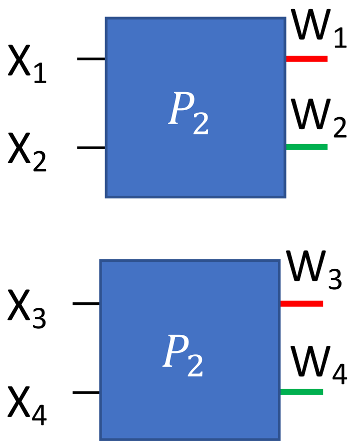


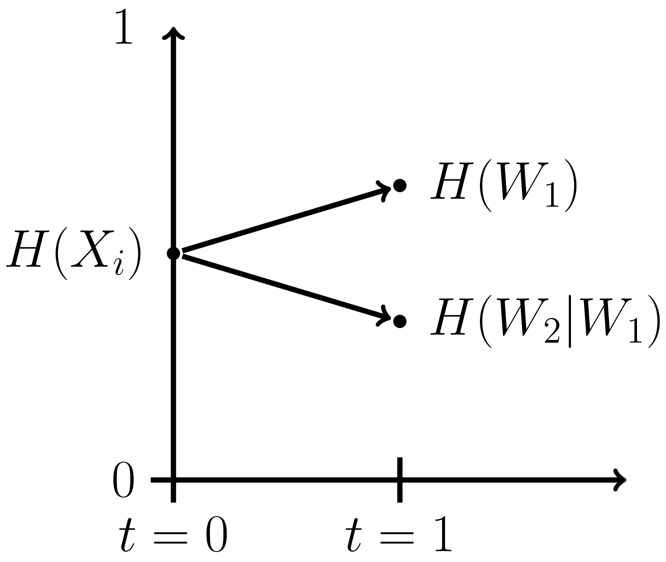
- Consider X, Y iid B(p), for $p \in (0,1)$
- P_2 invertible $\Longrightarrow H(X,Y) = H(X+Y, Y)$
- **H(X + Y) > H(X)**
- Thus, **H(Y | X+Y) < H(Y)**
- **Now recurse!**



Polar Transform

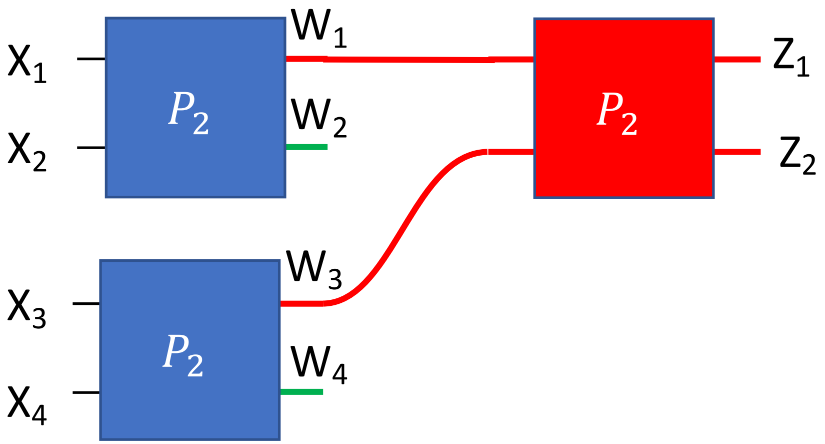
Consider X_i iid B(p), for $p \in (0,1)$

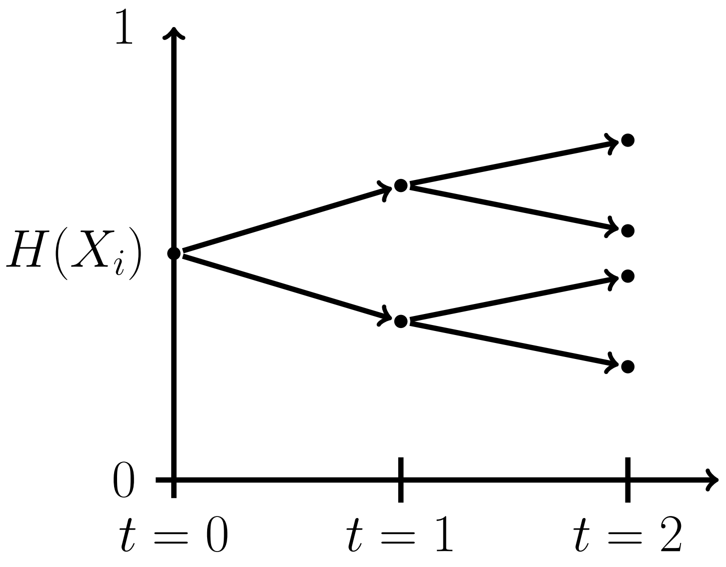




Polar Transform

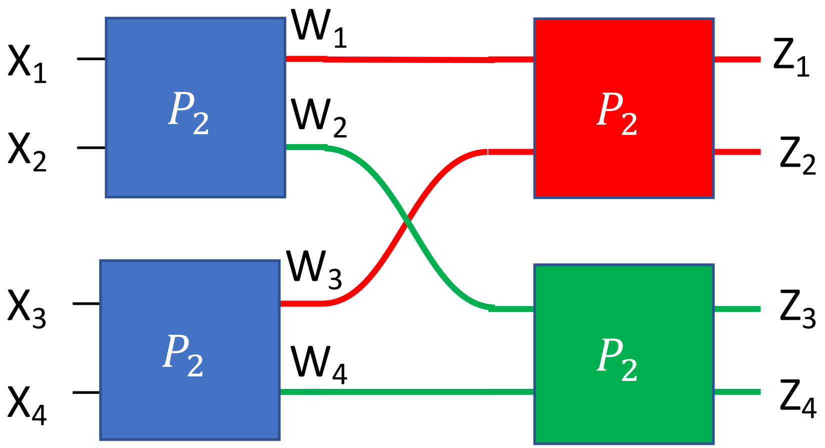
Consider X_i iid B(p), for $p \in (0,1)$

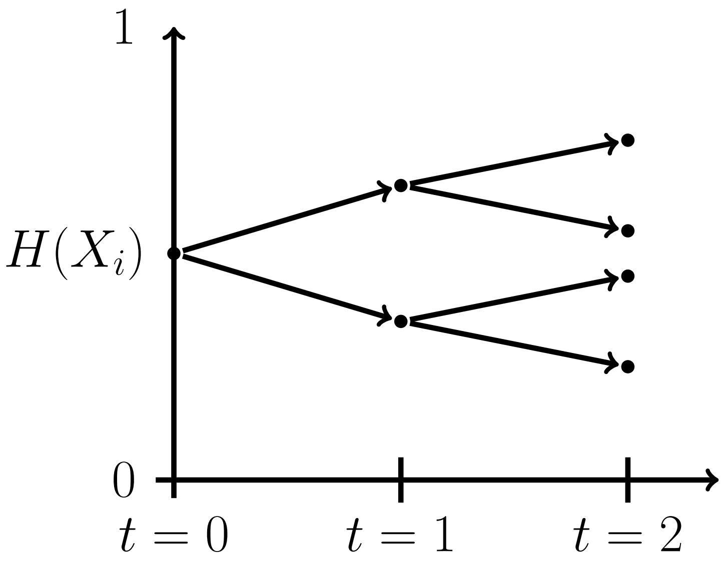




Polar Transform

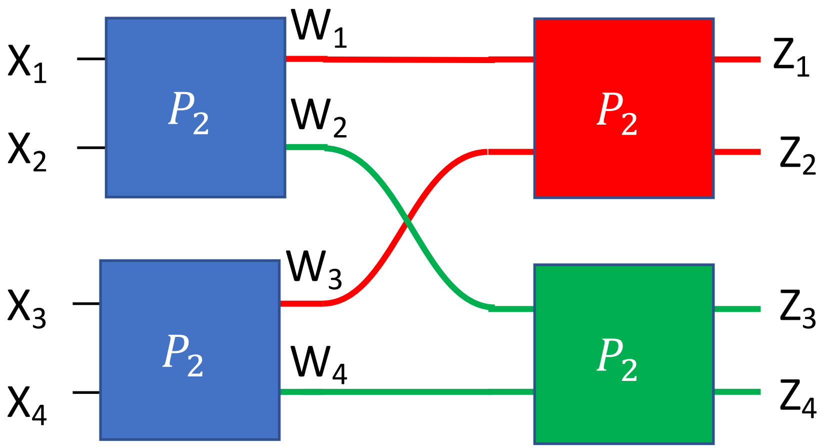
Consider X_i iid B(p), for $p \in (0, 1)$



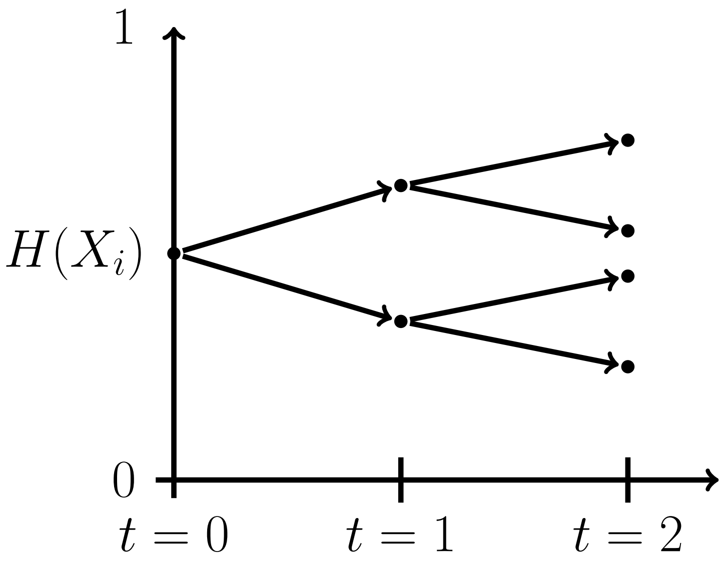


Polar Transform

Consider X_i iid B(p), for $p \in (0, 1)$



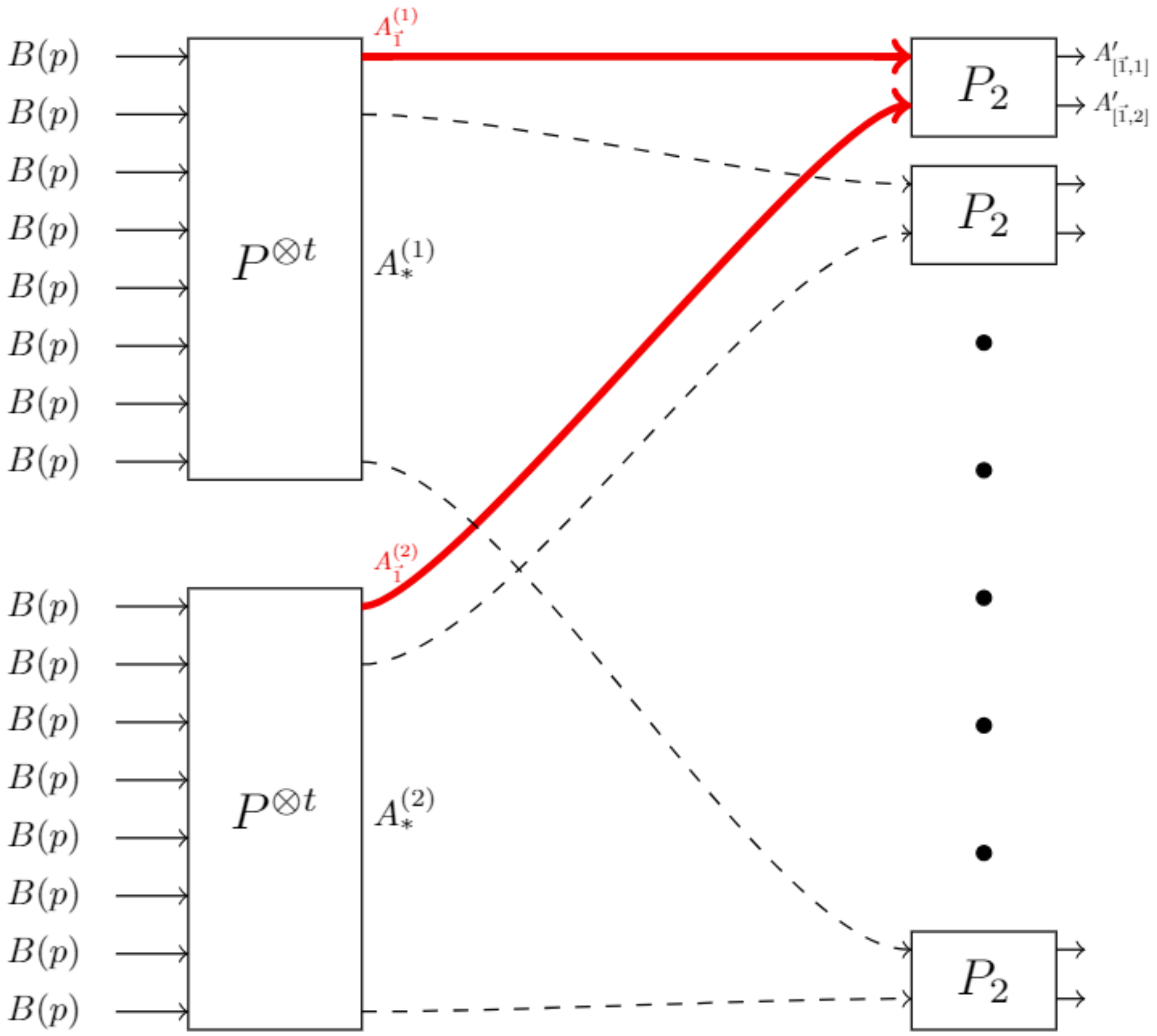
Consider $H(A_i | A^{<i})$:



Hope: most of these entropies eventually close to 0 or 1

Polar Transform

- In general, the recursion is:



Equivalent to: $P_{2^t} \stackrel{\text{def}}{=} P_2^{\otimes t}$

Analysis: Arikan Martingale

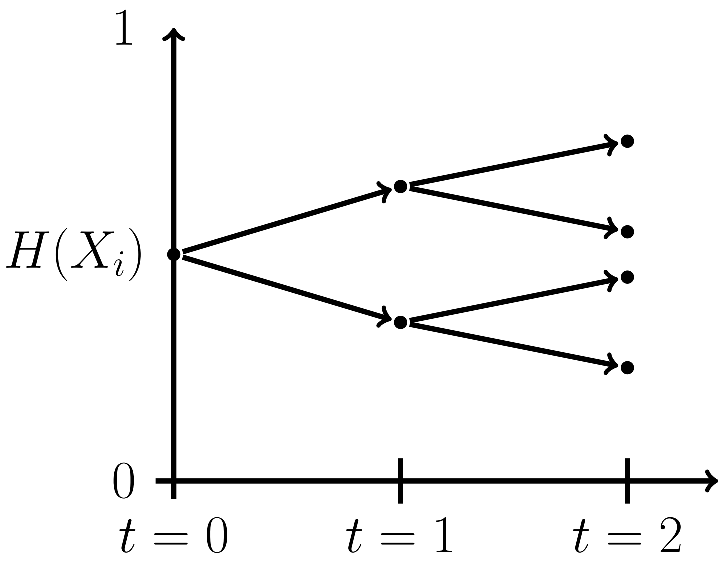
- Let Z_t be entropy of a random wire conditioned on wires above it:

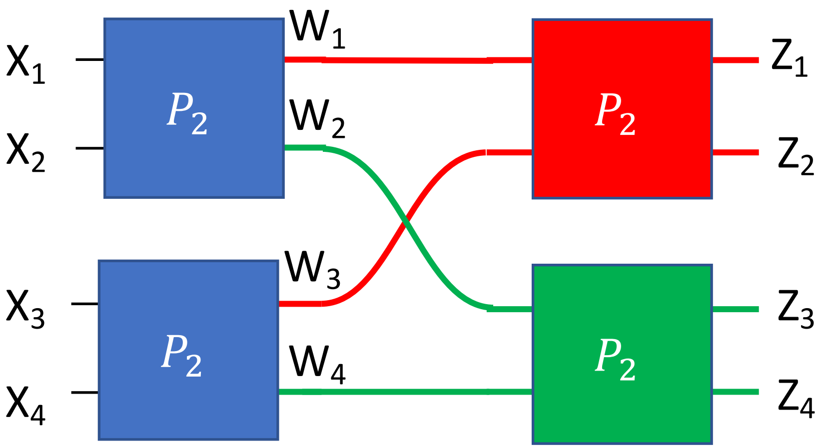
$$Z_t = H(A_t[i] \mid A_t[< i])$$

- **$\boldsymbol{Z_t}$ forms a martingale**

$$\mathbb{E}[Z_{t+1} \mid Z_t] = Z_t$$

because entropy conserved



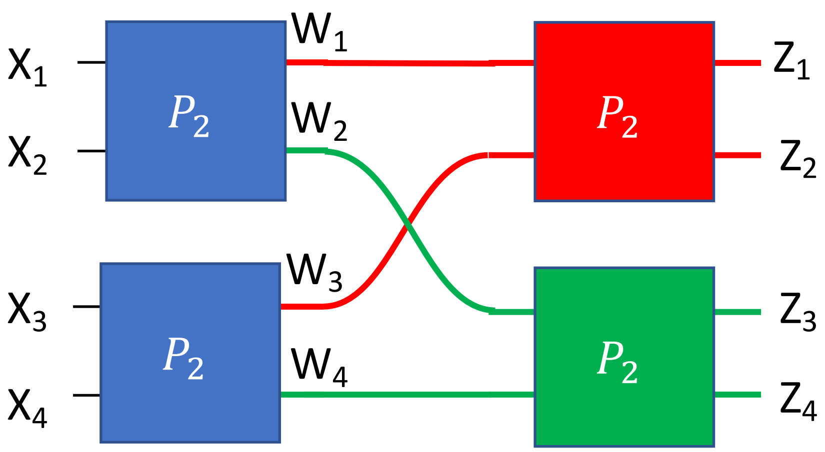


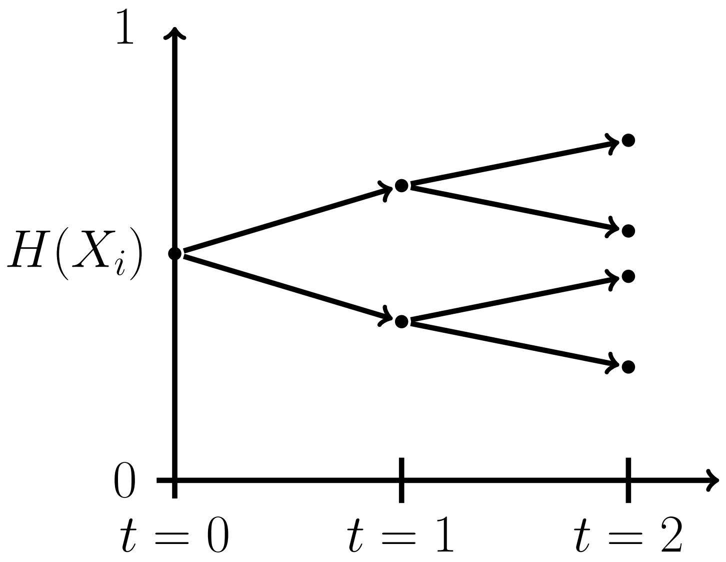
Analysis: Arikan Martingale

We want **fast convergence**: To achieve ϵ-close to entropy rate efficiently, ie with blocklength $n = 2^t = poly(\frac{1}{\epsilon})$, we need:

$$t \geq \Omega(\log(1/\epsilon)) \implies \Pr[Z_t \notin (4^{-t}, 1 - 4^{-t})] \leq \epsilon$$

$1/n^2$

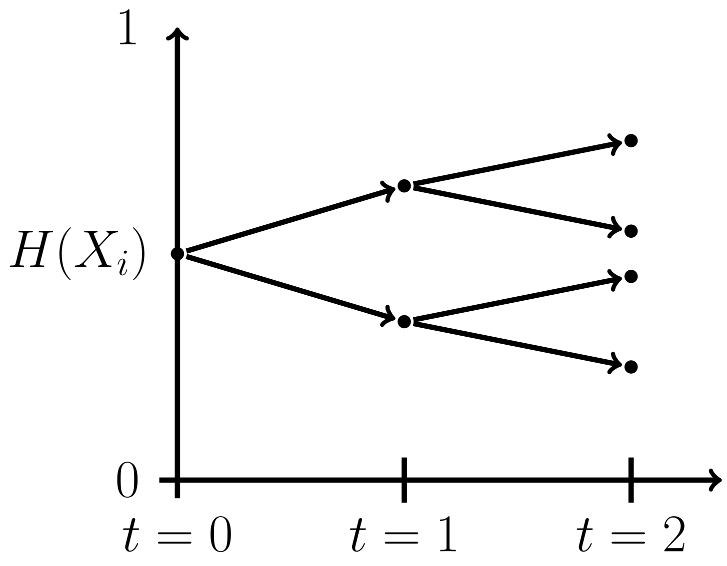




Martingale Convergence

- NOT every [0, 1] martingale converges to 0 or 1:
  - $X_{t+1} = X_t \pm 2^{-t}$
  - $\lim_{t\to\infty} X_t$ converges to Uniform[0, 1]

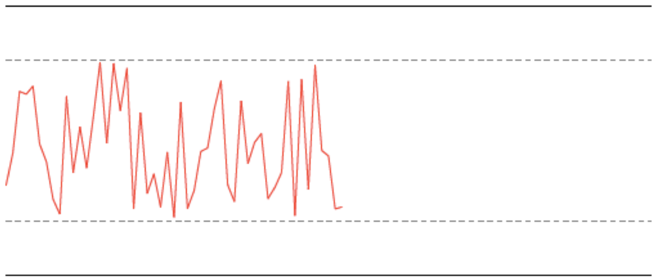
- Will introduce sufficient **local** conditions for fast convergence: ``Local Polarization”



Local Polarization

Properties of the Martingale:

1. Variance in the Middle:

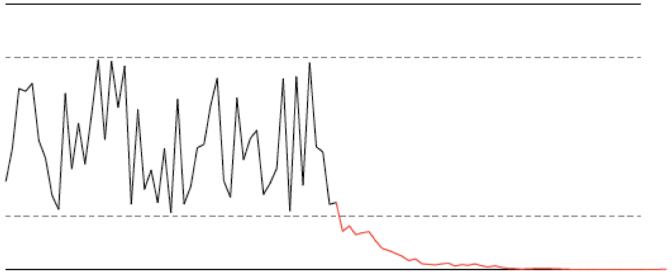


$$\forall \tau, \exists \sigma_\tau \text{ s.t. } Z_t \in (\tau, 1-\tau) \implies Var[Z_{t+1}|Z_t] \geq \sigma_\tau$$

2. Suction at the Ends:

$$\exists \tau \text{ s.t. } Z_t < \tau \implies \Pr[Z_{t+1} < Z_t/100] \geq 1/2$$

and symmetrically for the upper end.



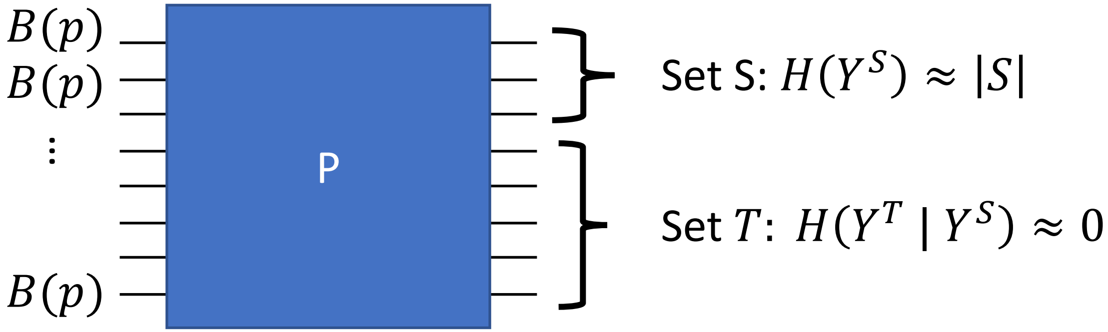
(easy to show these properties)

Recall, we want to show:

$$t \geq \Omega(\log(1/\epsilon)) \implies \Pr[Z_t \notin (4^{-t}, 1-4^{-t})] \leq \epsilon$$

Results of Polarization

- **So far:** After $t = O(\log 1/\epsilon)$ steps of polarization, the resulting polar code of blocklength $n = 2^t = poly\left(\frac{1}{\epsilon}\right)$ has a set T of indices s.t:
  - $\forall i \in T$: $H(Y_i | Y^{<i}) \approx 0$
  - $|T|/n \leq 1 - H(p) + \epsilon$



- Compression: Output Y^S
- Decompression: Guess Y^T given Y^S (ML decoding)

Polar Codes

Inputs Auxiliary Info

Theorem: For every distribution D over (X, W), where $X \in \mathbb{F}_q$,

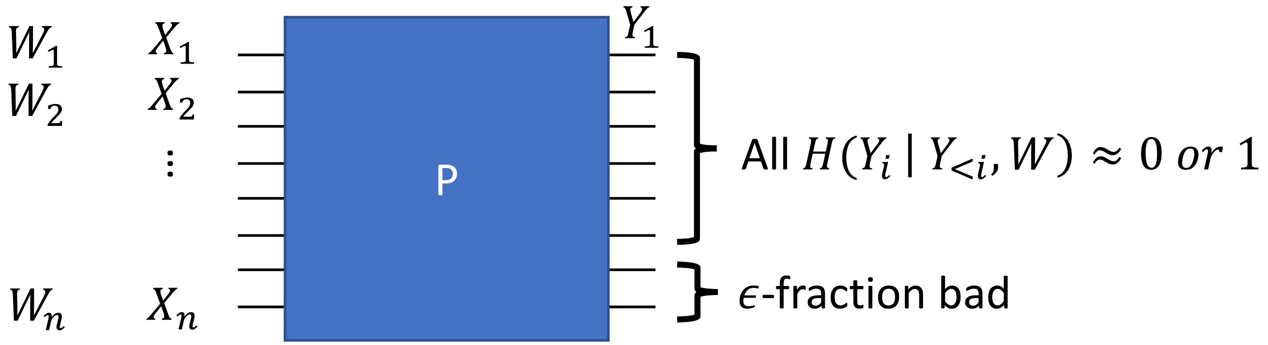
Let $X = (X_1, X_2, \dots X_n)$ and $W = (W_1, W_2, \dots W_n)$ where $(X_i, W_i) \sim D\ iid$

Then, entropies of $Y := P_n(X)$ are polarized:

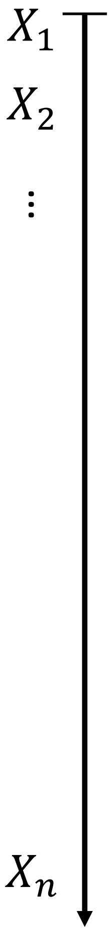
$\forall \epsilon$: if $n \geq poly\left(\frac{1}{\epsilon}\right)$, then all but ϵ-fraction of indices $i \in [n]$

have entropies

$$H(Y_i | Y_{<i}, W) \notin (n^{-4}, 1 - n^{-4})$$

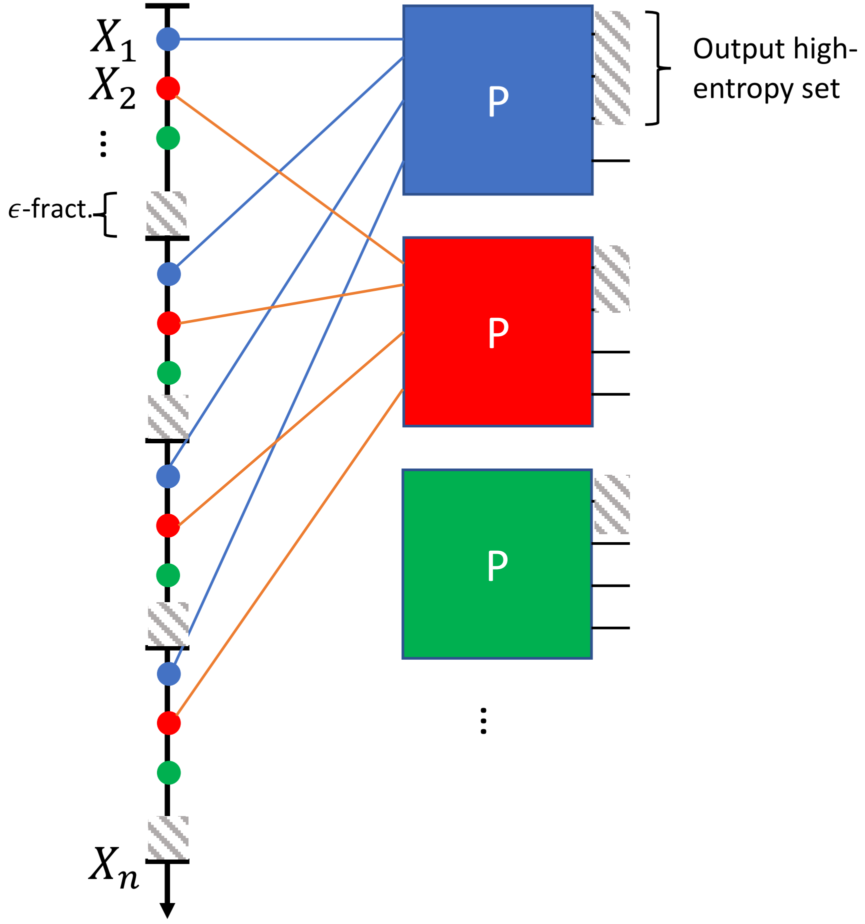


Compressing Hidden Markov Sources



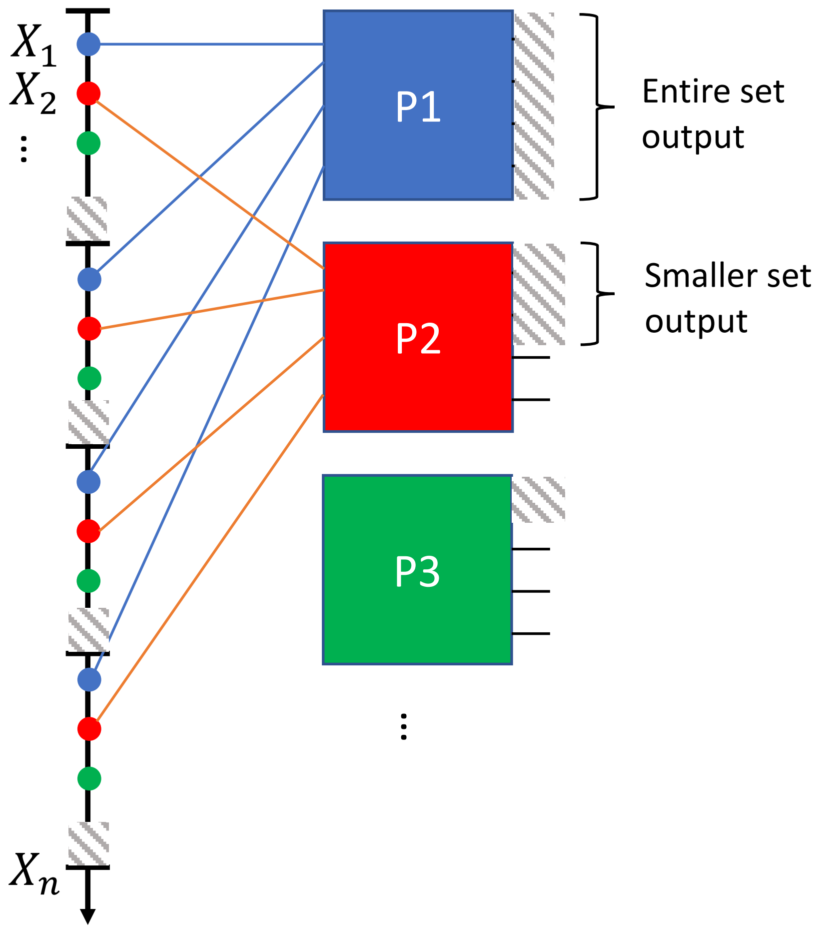
- $X_1, X_2, \ldots\ X_n$ are outputs of a Hidden-Markov Model
  - Not independent: Lots of dependencies between neighboring symbols
- **Goal:** Want to compress to within $H(X^n) + \epsilon n$
- First glance: everything breaks!
  - Polar code analysis (Martingale) relied on input being independent, identical
- But, simple construction works…

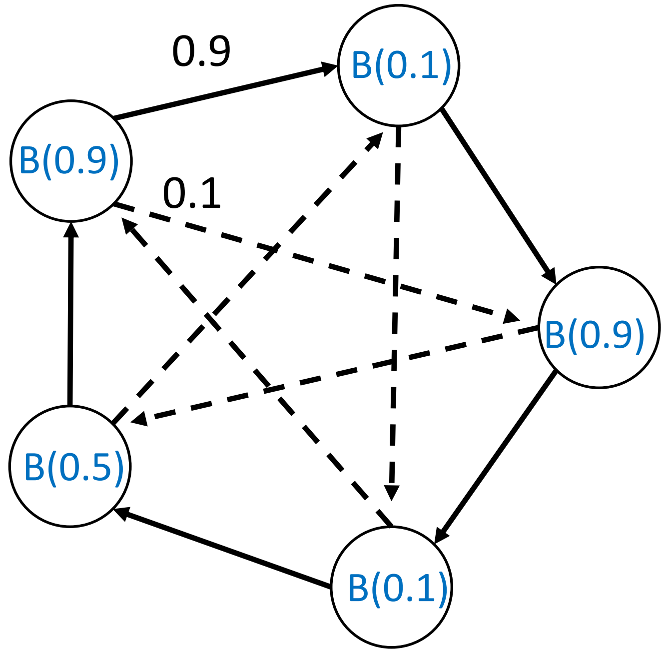
Compression Construction



- $X_1, X_2, \ldots\ X_n$: outputs of a stationary HMM
  - Mixing time $\ll n$
- Break input into $\sqrt{n}$ blocks of $\sqrt{n}$.
- Polarize the 1st symbols of each block.
  - These are approx. independent!
- Then Polarize the 2nd symbols
  - Polarizing ● , **conditioned on** ●
  - Joint distribution of all {(●,●)} is approx. independent across blocks
- Output last ϵ-fraction of each block in the clear

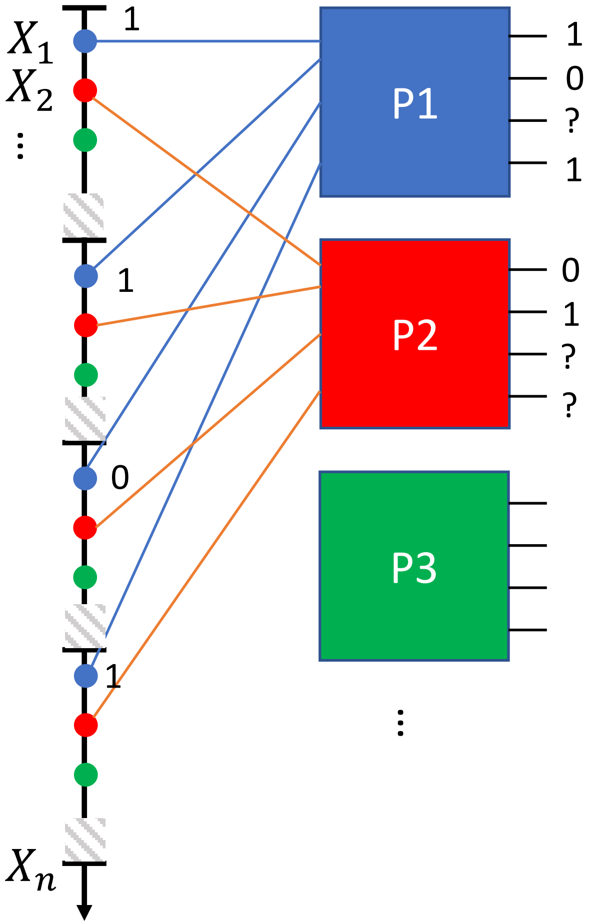
Example





- HMM: Marginally, X_i is uniform bit
- P1: inputs have full entropy
- P2: inputs have lower entropy, conditioned on P1

Decompression



Polar-decoder Black Box:

Input:

- Product distribution on inputs
- Setting of high-entropy polarization outputs

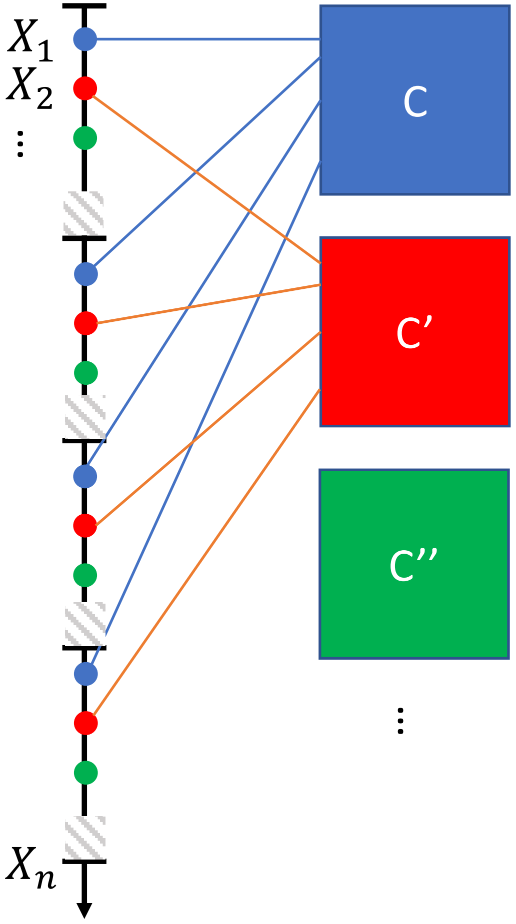
Output:

- Estimate of input

Markov decoding:

1. Decompress P1 outputs
2. Compute distribution of P2 inputs, conditioned on P1
3. Decompress P2 outputs
4. …

Decompression: Extras



<u>Note:</u>

Could have done this with any black-box compression scheme for independent, non-identically distributed symbols.

But: non-linear (and messy)

- Linear compression black-box for every fixed distribution on symbols $\not\Rightarrow$ overall linear compression

Polar codes are particularly suited for this