Data-driven Optimization for Zero-delay Lossy Source Coding with Side Information

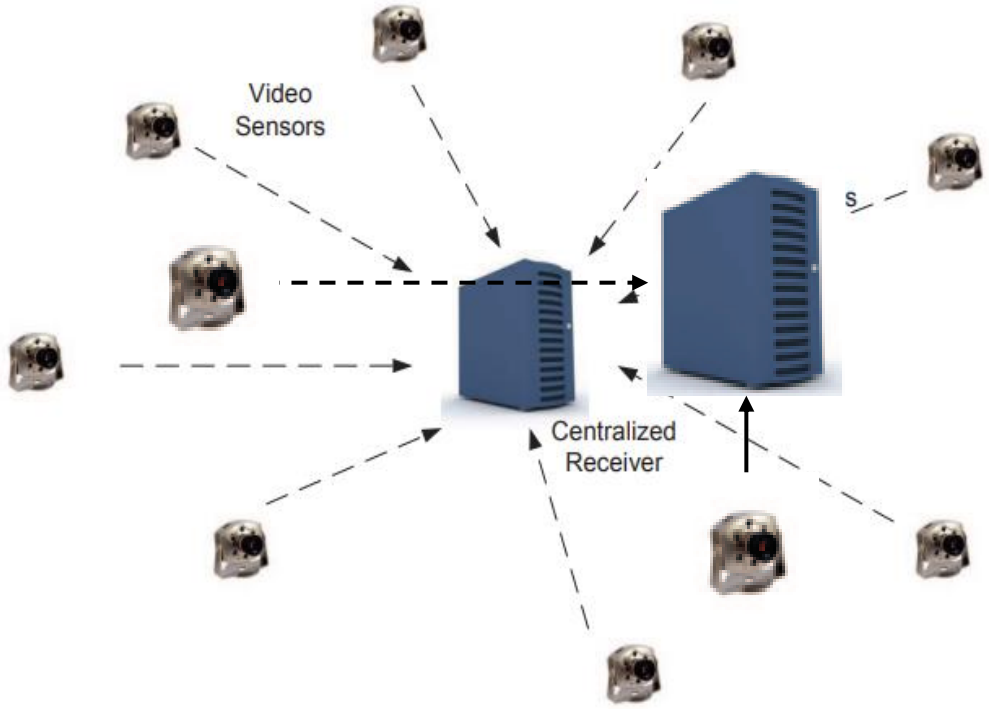
Elad Domanovitz, University of Toronto

Joint work with
Daniel Severo, Ashish Khisti and Wei Yu
University of Toronto

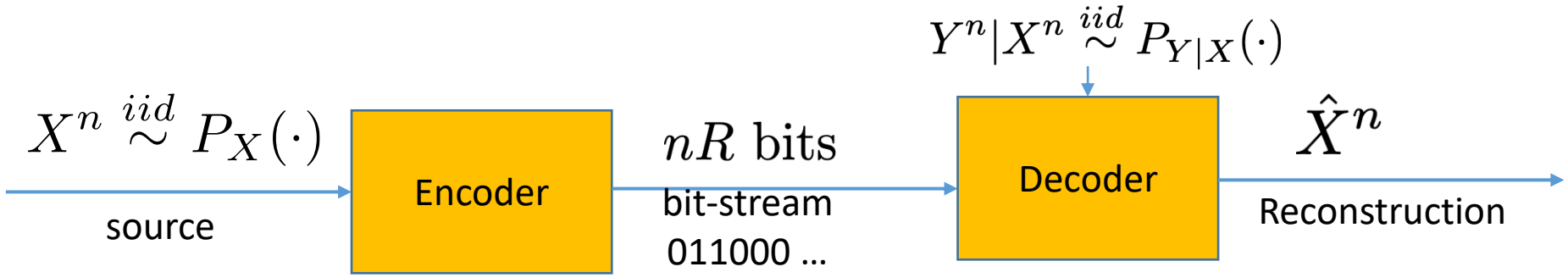
ICASSP 2022

Motivation

- Wireless sensor networks
  - Reduce rate (= reduce power consumption)
  - Delay sensitive applications
    - Anomaly detection
- Can correlation be used to reduce rate?
  - Yes, if side-information (SI) is known at the encoder
- Can correlation be used to reduce rate if SI is not known at the encoder?
  - Yes!
  - Lossless – Slepian, Wolf ‘73
  - Lossy – Wyner, Ziv ‘76
  - Delay ?



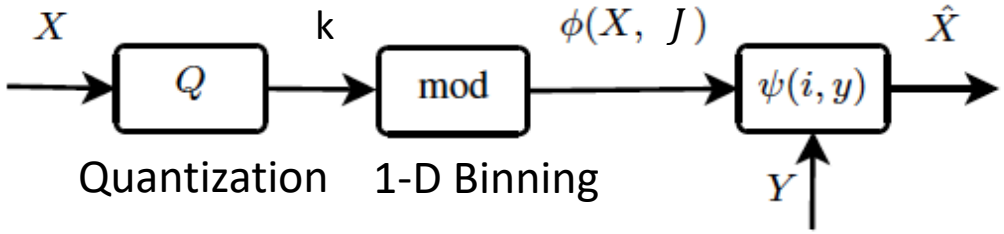
Wyner-Ziv Coding



- Decoder has access to a side information sequence
- Rate-Distortion function was characterized by Wyner and Ziv (1976)
- Gaussian Sources: $R(D) = \frac{1}{2}\log\frac{\sigma^2_{X|Y}}{D}$
- Same rate-distortion function when the side information is known to both the encoder and decoder!
- Drawbacks
  - Asymptotic i.i.d. setting
  - Joint distribution known
  - Random Coding, infinite block length -> infinite delay

Zero delay (Scalar) Wyner-Ziv Coding

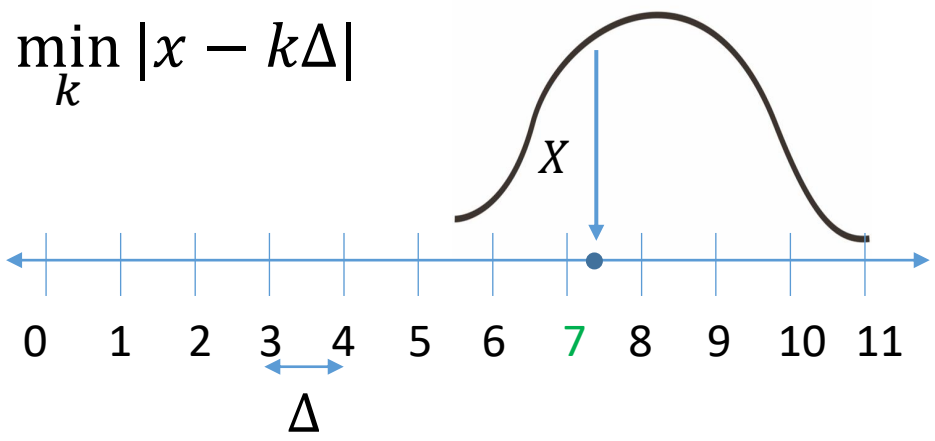
- Zamir, Erez, Shamai TIT'02



Encoding:

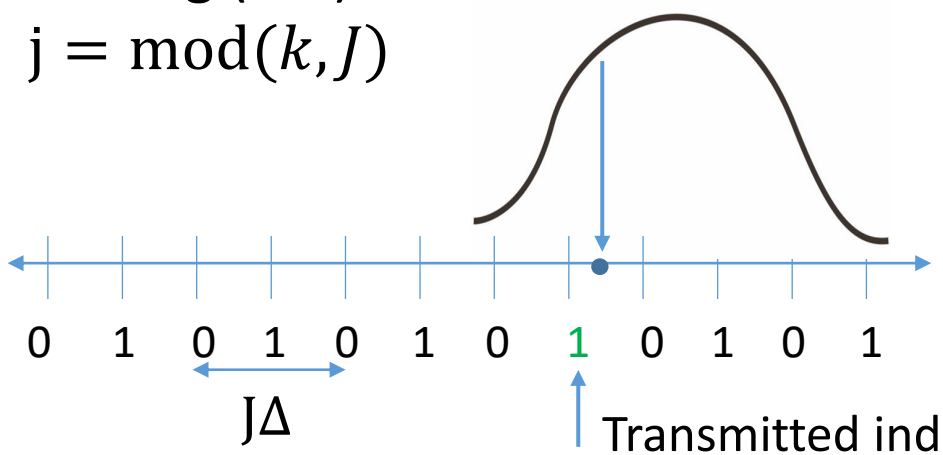
Uniform Scalar Quantizer:

$$\mathrm{k} = \min_{k} |x - k\Delta|$$

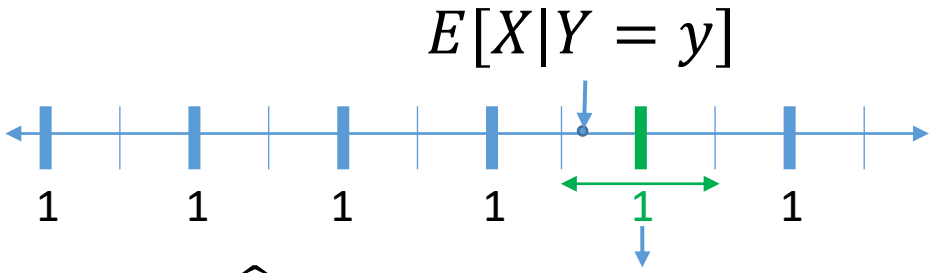


Binning (J=2):

$$\mathrm{j} = \mathrm{mod}(k, J)$$



Decoding:



Zero delay (Scalar) Wyner-Ziv Coding

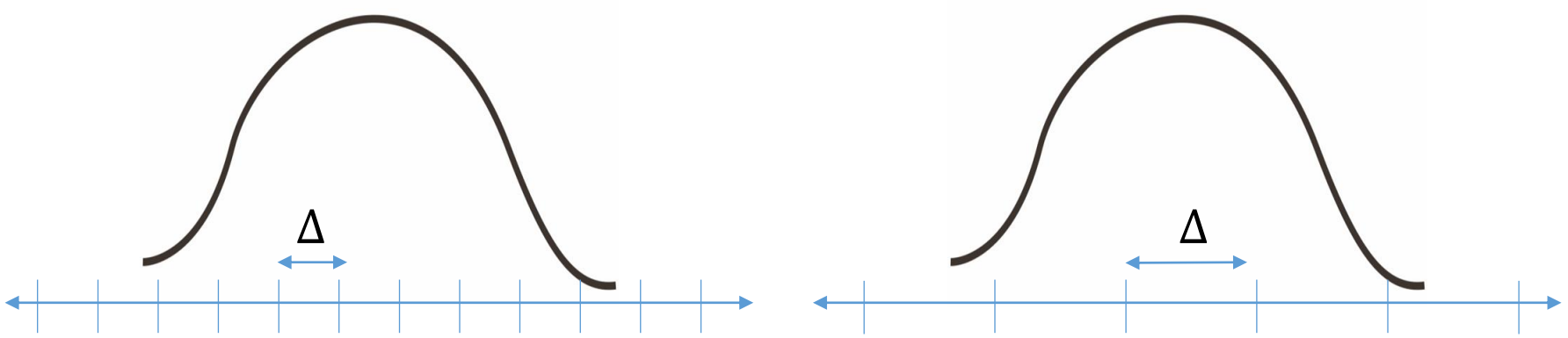
- Encoding:
  - Uniform Scalar Quantization $Q: \mathbb{R} \longrightarrow \mathbb{Z}$, $\mathrm{k} = \min_{k} |x - k\Delta|$
  - Reconstruction Points: $k \rightarrow \tilde{x}_k$
  - Binning (one dimensional modulo): $\Phi_{k,J} = k \bmod \mathrm{J} \in \{0,1, \dots, J-1\}$
    - $R = \log_2 J$
- Decoding:
  - Given $\Phi_{k,J} = j$ and $Y = y$, decode $\tilde{x}_k$:

$$\psi(j, \mathrm{E[X|Y=y]}) = \hat{\tilde{x}}_k = \underset{\mathrm{k}:\Phi_{k,J}=j}{\operatorname{argmin}} || k\Delta - E[X|Y=y] ||$$

  - Distortion: $D = E\left[\left(X - \hat{\tilde{X}}_k\right)^2\right]$ composed of
    - D_g =Granular distortion: fine quantizer granularity
    - D_o =“Overload” distortion: distortion resulting from deciding on wrong index

Zero delay (Scalar) Wyner-Ziv Coding: Analysis

- Chen, Tuncel T-SP'11:
  - X and Y jointly (memoryless) Gaussian with correlation ρ
  - Tradeoff in quantization interval



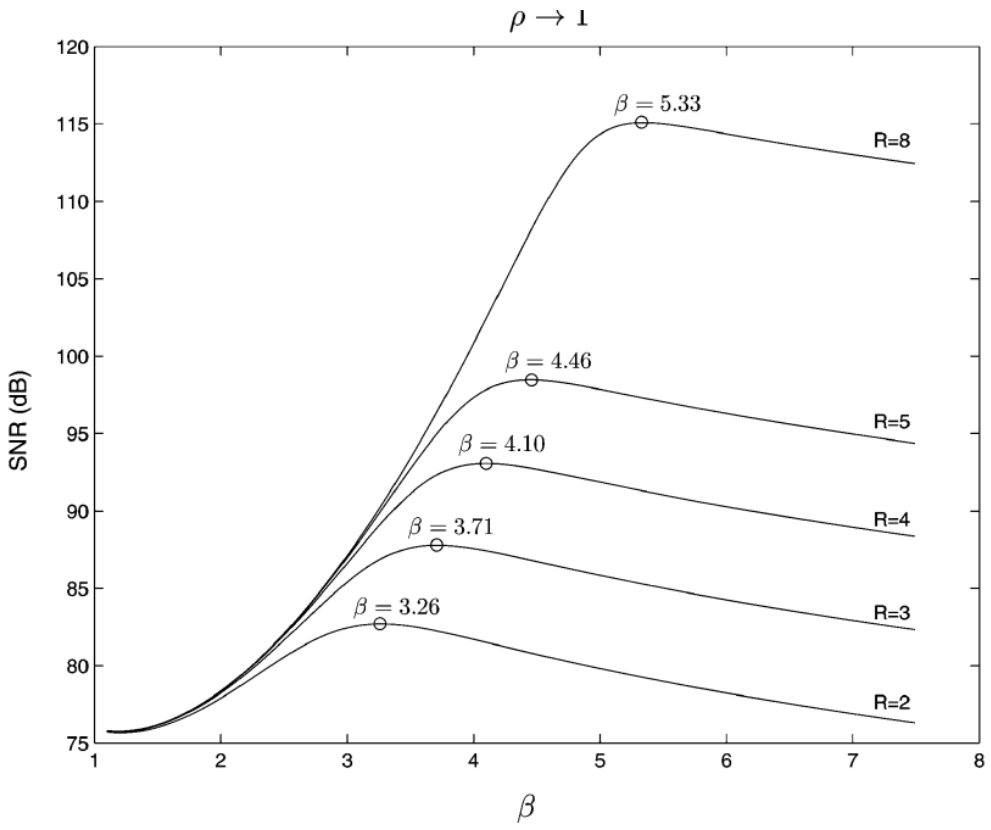
Dense quantizer= $D_g \downarrow, D_o \uparrow$ Sparse quantizer= $D_g \uparrow, D_o \downarrow$

- Assuming periodic quantizer:

$$\Delta = \frac{\beta}{J}\sqrt{1-\rho^2}, D = 4\sigma_x^2(1-\rho^2)\left(\frac{\beta}{J}\right)^2\left[\frac{2^{-2R}}{12} + 1 - \mathrm{erf}\left(\frac{\beta}{12J}\right)\right]$$

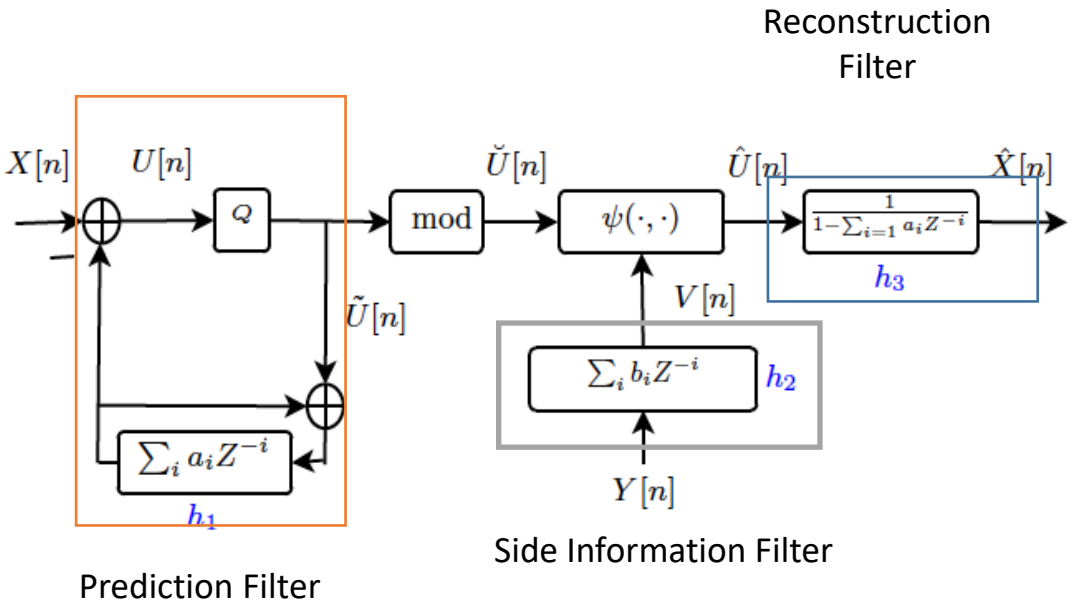
Zero delay (Scalar) Wyner-Ziv Coding: Analysis

- Chen, Tuncel T-SP'11:
  - X and Y jointly (memoryless) Gaussian with correlation ρ
  - Optimized Δ: $10\ dB$ gap in SQNR from ideal Wyner-Ziv limit
  - No closed-form solution for optimal β



WZ Coding: Sources with Memory

- Chen, Tuncel T-SP'11:
  - DPCM with
    - Prediction Filter (Feedback): $A(z)$
    - Reconstruction Filter: $\mathrm{A}^{-1}(z)$
    - SI Filter: $B(z) = 1 + bz^{-1}$
  - Exhaustive search over filter coefficients
  - Only considers Gaussian AR(1) processes
  - 10 dB SQNR gap with respect to ideal Wyner-Ziv limit

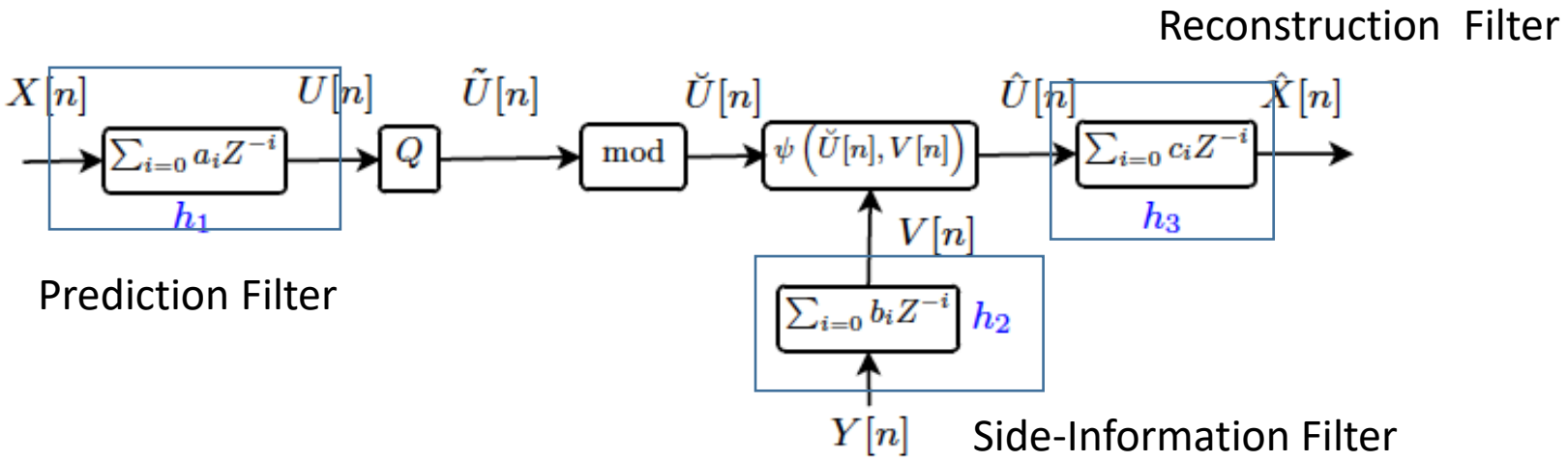


Model vs Data Driven Approach

	Model Driven Approach	Data Driven Approach
Requires	Joint Distribution	Training samples
Method of optimization	Exhaustive search of filter coefficients	Stochastic gradient descent
Complexity	Scales exponentially with memory	Does not scale exponentially
Implemented	Only for Gaussian AR(1) processes	Easily implemented for sources with larger memory

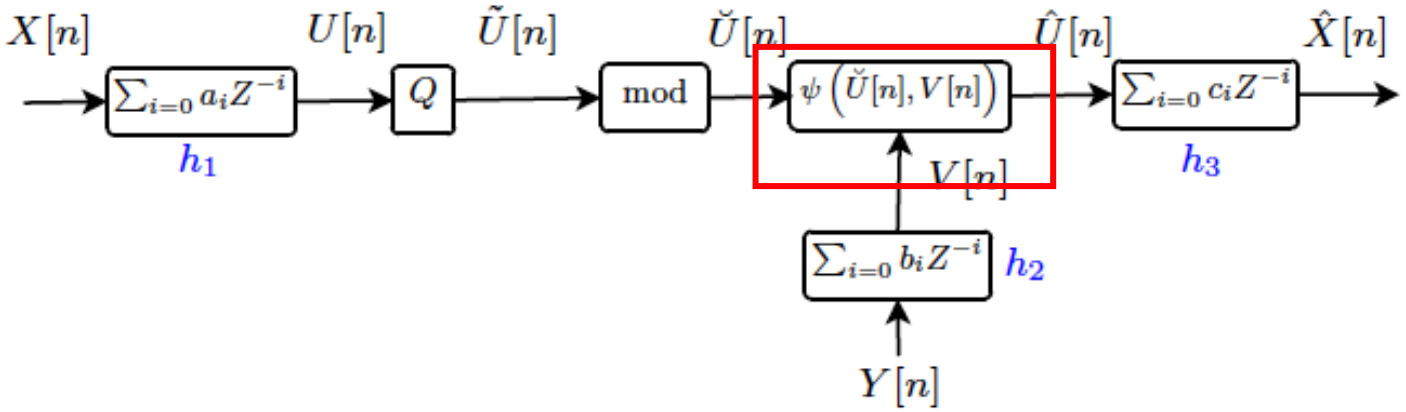
- Prior art:
  - Fleming, Zhao, Effros T-IT‘04
  - Saxena, Rose T-SP‘09
- Data-Driven approaches based on alternating optimization algorithms
- Each block is updated when other are held fixed
  - Coordination
  - Complexity
  - Results presented only for AR(1) processes

Proposed Architecture: Filter Choice



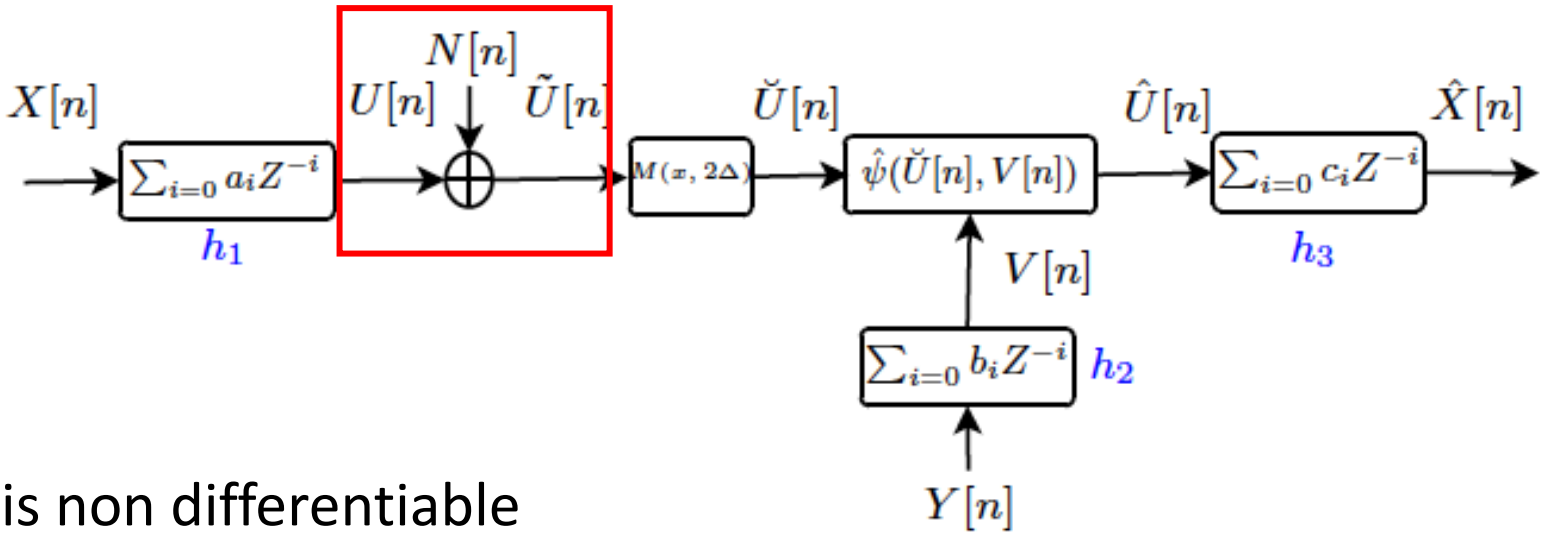
- Prediction filter feedforward: $A(z) \equiv [a_0, \ldots, a_{L_1}]$
- Reconstruction filter: $\mathrm{C}(z) \equiv [c_0, \ldots, c_{L_3}]$
- Side-Information filter: $B(z) \equiv [b_0, \ldots, b_{L_2}]$

Proposed Architecture: Decoding Function



- Given $\Phi(\widetilde{U}[n])$= j and $V[n] = v$, decode $\tilde{u}_k$ as
  - $\hat{\tilde{u}}_k = \min_{k:\Phi_{k,J}= j} || k\Delta - v||$
- Assuming $E[u[n]|Y[n]] \approx v$ is justified as the filters during training should enforce it

Training Architecture: Quantization

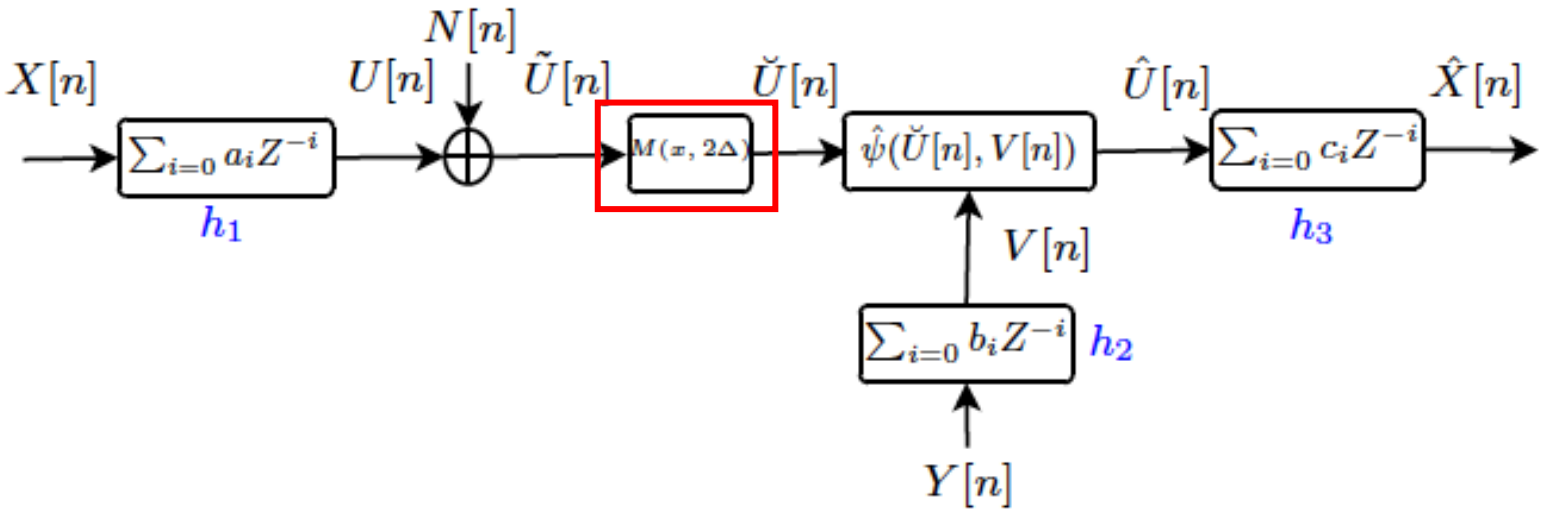


- $Q(U[n])$ is non differentiable
- During training replace with:

$$\widetilde{U}[n] = U[n] + N[n], N[n] = \mathcal{N}\left(0, \frac{\Delta^2}{3} 2^{-2R}\right)$$

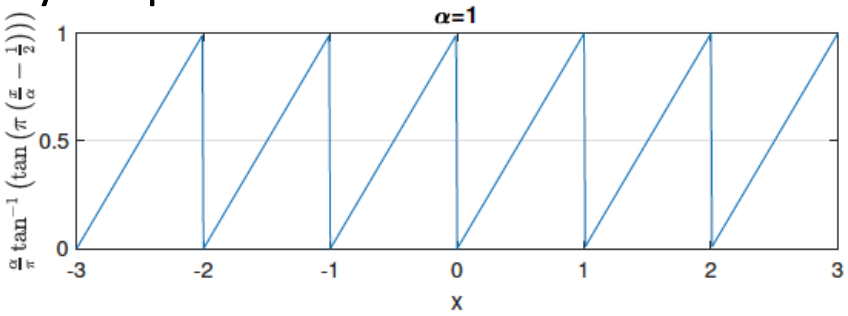
- Similar approaches are commonly used in learned image compression:
  - Balle, Minnen, Singh, Hwang, Johnston ICLR'18
  - Zhang, Qian, Chen, Khisti NeurIPS'21

Training Architecture: Modulo Function

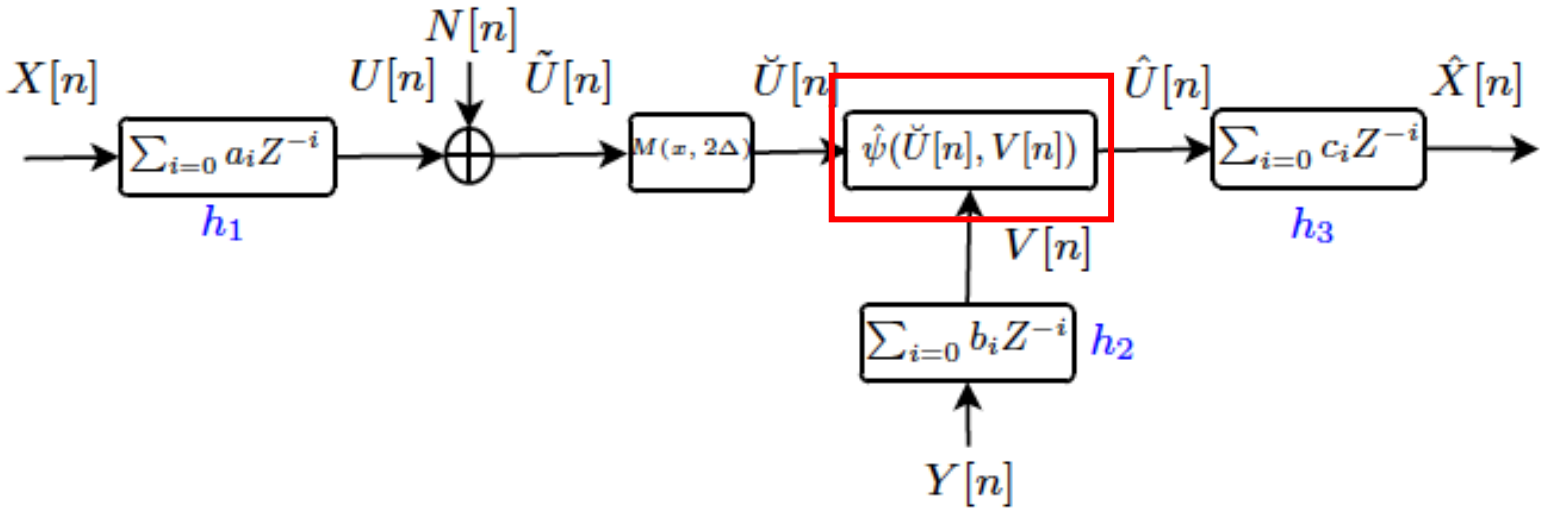


- Desired modulo function: Maps $\tilde{U}[n]$ to $\breve{U}[n] = mod\left(\tilde{U}[n] + \frac{J\Delta}{2}, J\Delta\right) - \frac{J\Delta}{2}$
  - Non differentiable
  - Differentiation with respect to the modulo size not always implemented
  - Replace with:

$$M(x, \alpha) \approx \frac{\alpha}{\pi} \tan^{-1}\left(\tan\left(\pi\left(\frac{x}{\alpha} - \frac{1}{2}\right)\right)\right)$$



Training Architecture: SoftMin Decoder



- $\widehat{U}[n] = \min_j || \breve{U}[n] + jJ\Delta - v||$ is non differentiable
- Replace by a "soft-min function"

$$\breve{\psi}(j, v, K) = \sum_{k=\lfloor \frac{K}{2} \rfloor}^{\lceil \frac{K}{2} \rceil} k \frac{e^{-Tw_k}}{\sum_j e^{-Tw_j}}, \quad w_k = \left(\breve{U}[n] - kJ\Delta - v\right)^2$$

  - T=Temperature

Training Algorithm: SGD

- Loss Function: $\mathcal{L}(a, b, c, \Delta, \mathrm{R}) = \mathbb{E}^{emp}\left[\left\|X[n] - \hat{X}[n]\right\|^2\right]$

- Algorithm (SGD)

Algorithm 1: Data-driven scalar WZ optimization

Result: Coefficients of the prediction filters, Δ
Input: N, L_1, L_2, L_3, LR, Convergence condition
Initialization: $a_0 = b_0 = c_0 = 1$, other coefficients equal zero, random Δ;
while *not converged* **do**
    Calculate the empirical distortion;
    Update coefficients and modulo range using SGD;
end

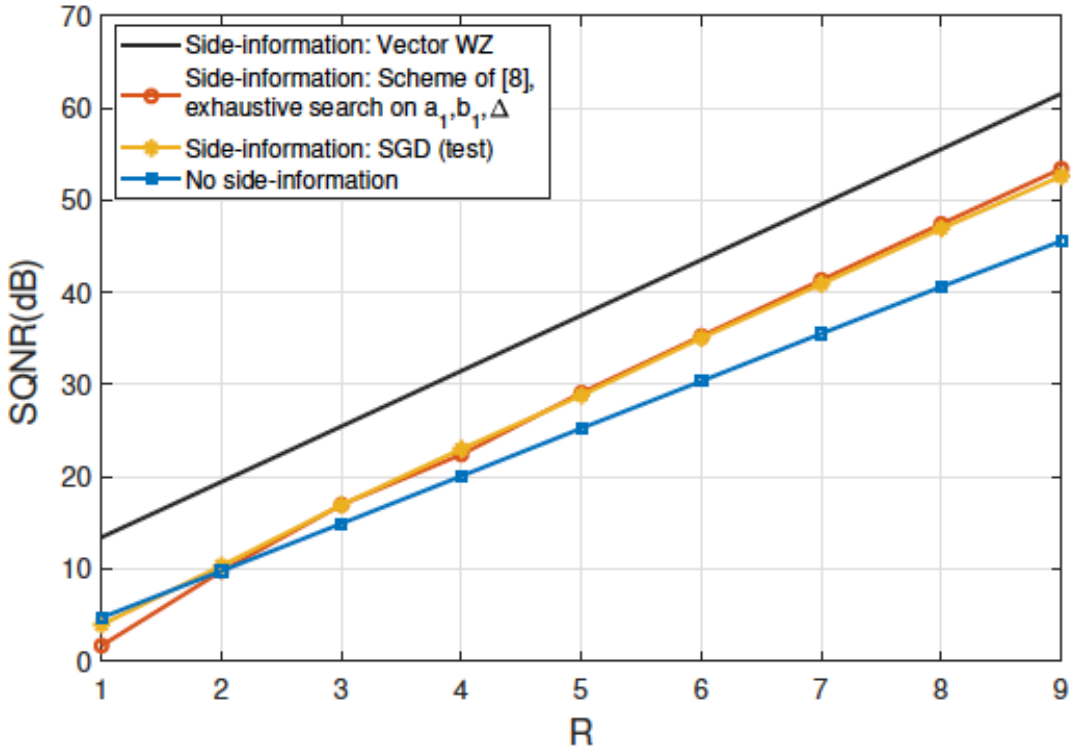
Experimental Results - 1

- First order Gauss Markov (GM) processes (Chen, Tuncel TSP'10)

$$T(n) = \rho T(n-1) + W(n),$$
$$W(n) \overset{iid}{\sim} N(0, 1-\rho^2),\ \sigma_T^2 = 1$$

$$X(n) = T(n) + N_x(n)$$
$$Y(n) = T(n) + N_y(n)$$
$$N_x(n) \overset{iid}{\sim} N(0, \sigma_x^2)$$
$$N_y(n) \overset{iid}{\sim} N(0, \sigma_y^2)$$

$$\rho = 0.7, \sigma_x = \sigma_y = 0.1$$



Experimental Results - 2

- Source: First order GM
- Side Information: Second order GM

$$T_1(n) = \rho_1 T(n-1) + W(n),$$

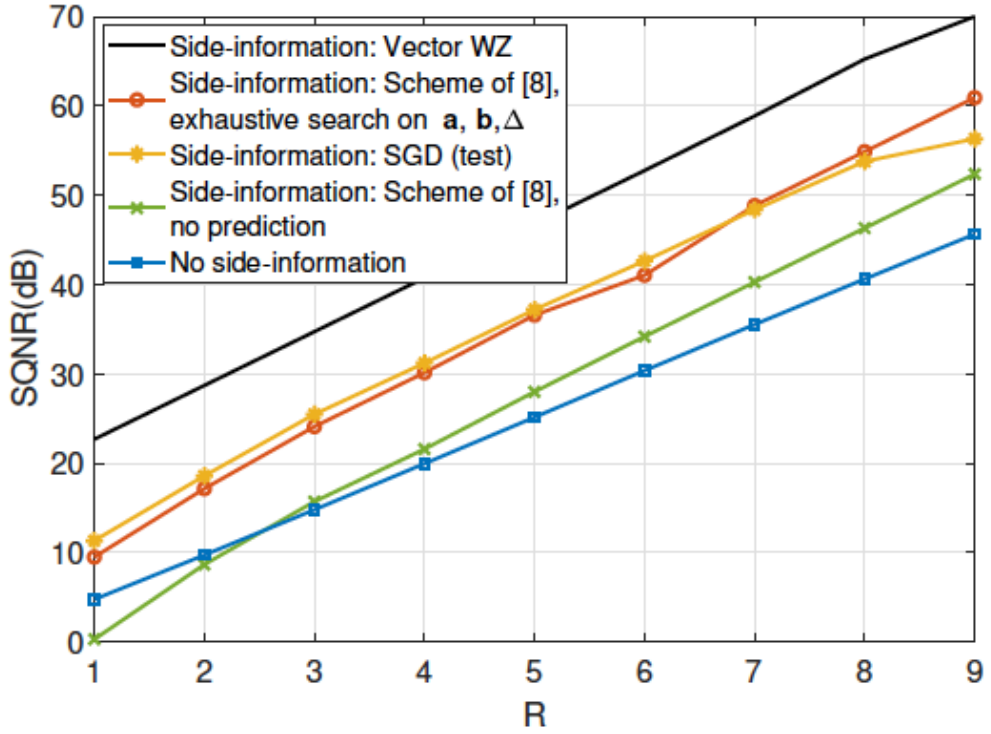
$$T_2(n) = \sum_{j=1}^{2} \rho_{2,j} T(n-j) + W(n)$$

$$X(n) = T_1(n) + N_x(n)$$

$$Y(n) = T_2(n) + N_y(n)$$

$$\rho_1 = 0.51, \rho_{2,1} = 0.05, \rho_{2,2} = 0.5$$

$$\sigma_x = \sigma_y = 0.1, \sigma_W = 0.86$$



Experimental Results - 3

- Source: Third order GM
- Side Information: Second order GM

$$T_1(n) = \sum_{j=1}^{3} \rho_{1,j} T(n-j) + W(n),$$

$$T_2(n) = \sum_{j=1}^{2} \rho_{2,j} T(n-j) + W(n)$$

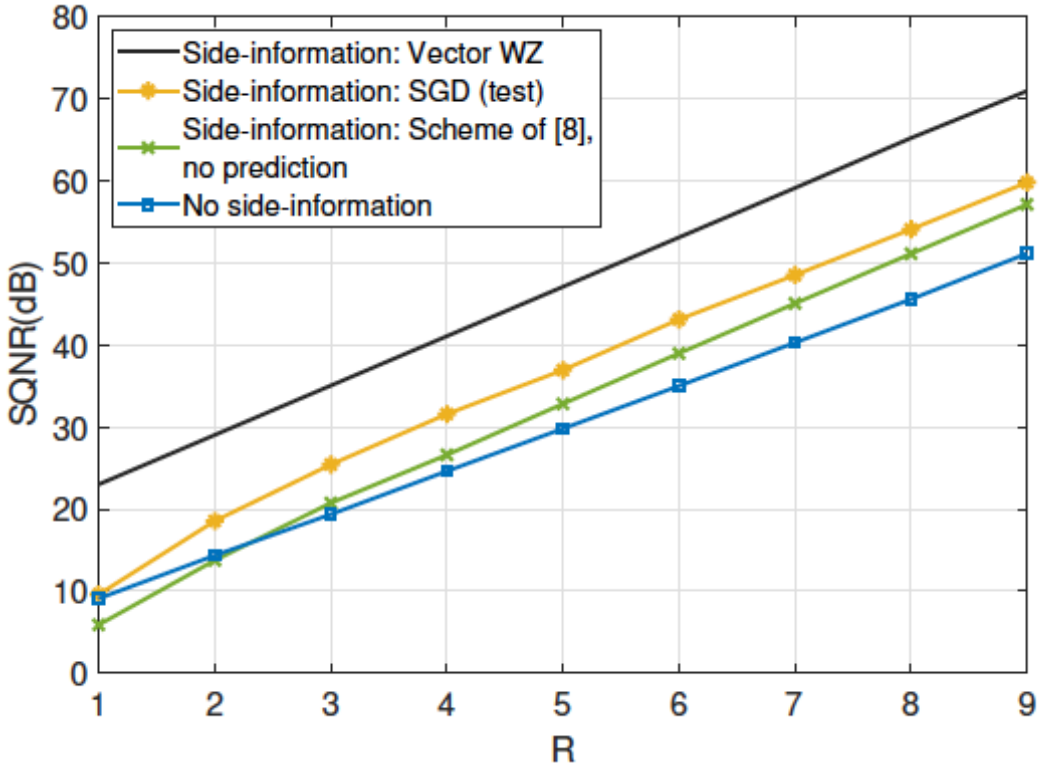
$$X(n) = T_1(n) + N_x(n)$$

$$Y(n) = T_2(n) + N_y(n)$$

$$\rho_{1,1} = 0.05, \rho_{1,2} = 0.5, \rho_{1,3} = 0.25,$$

$$\rho_{2,1} = 0.3, \rho_{2,2} = 0.6$$

$$\sigma_x = \sigma_y = 0.1, \sigma_W = 0.92$$



Conclusions and Next Steps

- Data driven approach for zero-delay lossy source coding with side information
  - Exampled convergence for high-order Gauss Markov Processes
- Updates all blocks simultaneously
- Consistently observed approximately 10dB loss in performance compared to ideal Wyner-Ziv

- Other (general) processes?
- Sensitivity to starting conditions?