Single-Gaussian Messages and Noise Thresholds for Low-Density Lattice Codes

Brian M. Kurkoski, Kazuhiko Yamaguchi and Kingo Kobayashi

kurkoski@ice.uec.ac.jp

Dept. of Information and Communications Engineering
University of Electro-Communications

Information Security Research Center
National Institute of Information and Communications Technology

Tokyo, Japan

International Symposium on Information Theory
June 30, 2009
Seoul, Korea

Background

- Low-density lattice codes (LDLC) are lattices [Sommer, Feder and Shalvi, IT 2008]
- Decoded using belief-propagation, like low-density parity check codes.
- Decoding complexity is linear in the dimension. Dimension n=10^5 possible.
  - "Classical" lattices can be decoded in dimension n=2~200
- Within 0.6 dB of unconstrained power capacity
- Messages (beliefs) are functions, rather than numbers. Existing implementations
  - Quantize the functions
  - Gaussian mixture approximation

In this talk

- Propose a LDLC decoder: belief-propagation messages approximated by a single Gaussian,
- Single-Gaussian decoding has a noise threshold within 0.1 dB of the quantized-message implementation (that is, 0.7 dB from capacity)
- But, single-Gaussian decoding has much lower complexity than other methods
- Design of regular LDLC lattices alpha=0.7 is better than alpha approaching 1

Low-Density Lattice Codes (LDLCs)

Sommer, Feder and Shalvi gave a lattice construction and decoding algorithm based upon low-density parity-check codes. Extensive convergence analysis in IT Trans, April 2008.

Low-Density Parity-Check Codes Codes

- Code over a finite field (binary)
- Sparse parity check matrix

LDLC

- Lattice: Code over the real numbers

$$\mathbf{x} = G \cdot \mathbf{b}$$

n-dim lattice points n-dim integer vector

- n-by-n inverse generator H=G^{-1} is sparse
- H has row and column weight d
- Non-zero entries in rows and columns:

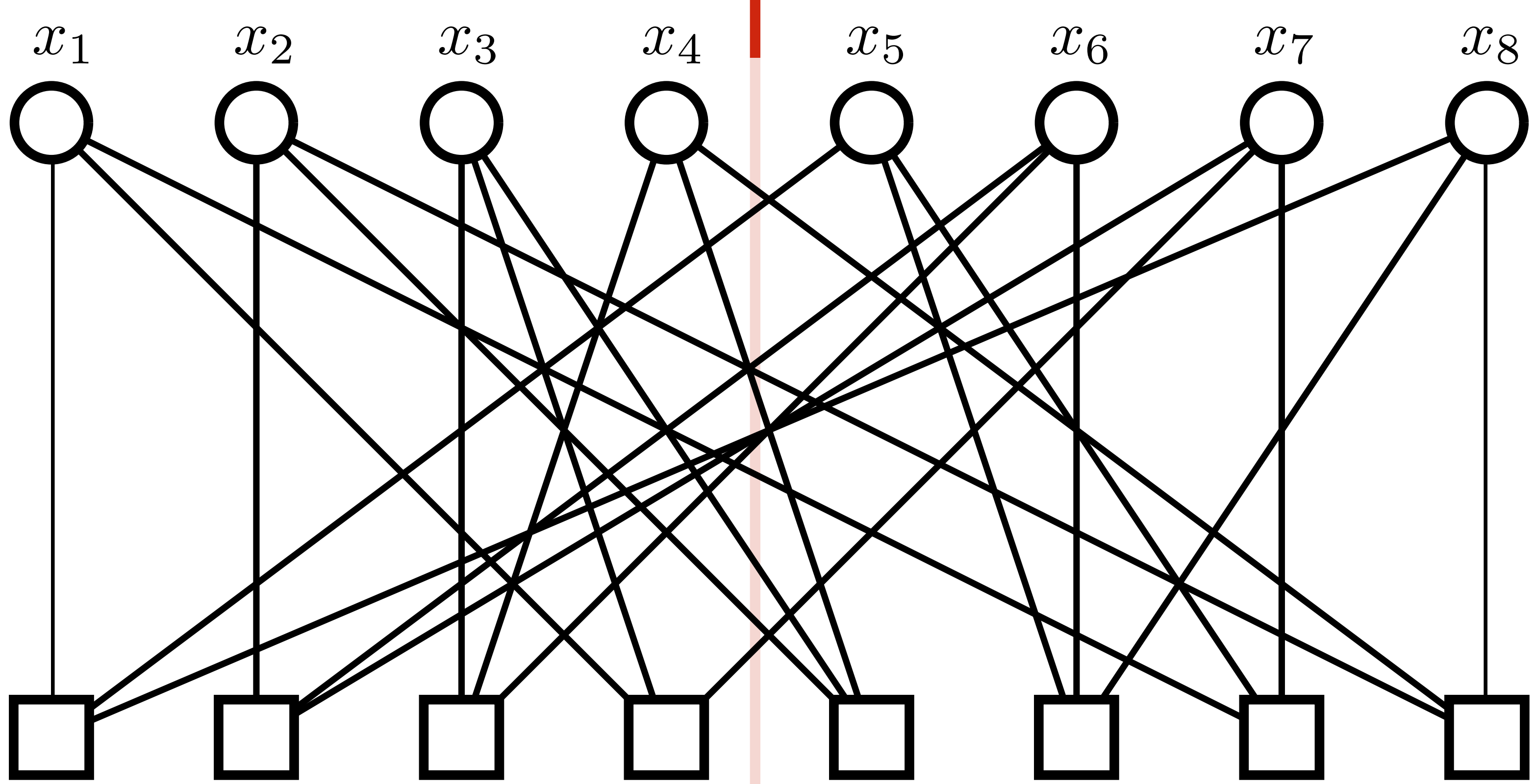
$$\pm\{h_1, h_2, \cdots, h_d\}$$

Low-Density Parity-Check Codes Codes

Variables are from the field

$x_i \in \{0, 1\}$

So, messages are scalars (LLRs)



check nodes < # variable nodes

$x_1 + x_2 + x_3 = 0$ (over field)

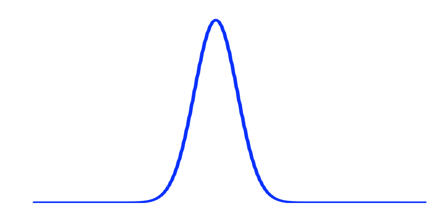
Approaches BIAWGN channel capacity

Low-Density Lattice Codes (LDLC)

Variables are real numbers

$x_i \in \mathbb{R}$

So, messages are functions (!)



check nodes = # variable nodes

$x_1 + x_2 + x_3 = \mathrm{b}$ (over real numbers)

b is an integer

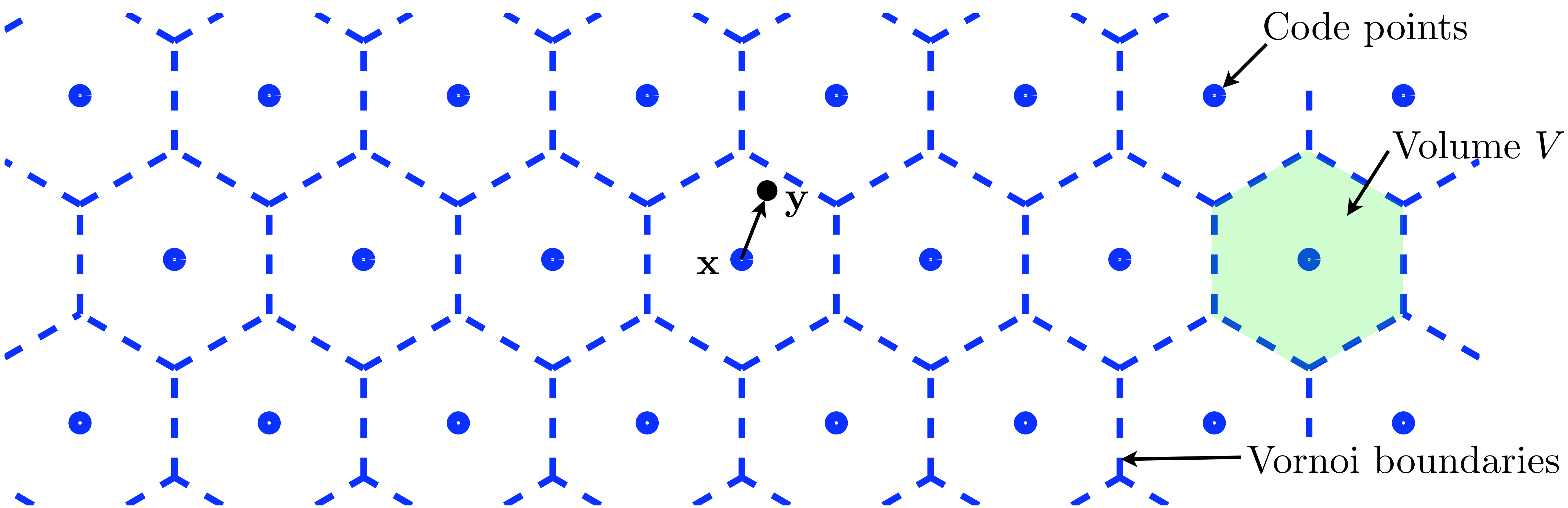
Comes within 0.6 dB of unconstrained power AWGN channel capacity.

Channel Model: Unconstrained Power Transmission

- Use entire lattice for transmission
  - The code is linear!
- Reliable "communication" [Poltyrev, etc.] if and only if

$$\mathrm{SNR} \geq 2\pi e, \qquad \mathrm{SNR} = \frac{\sqrt[n]{V^2}}{\sigma^2} \qquad \text{Noise power } \sigma^2$$

- Transmit power is unconstrained; no shaping region.

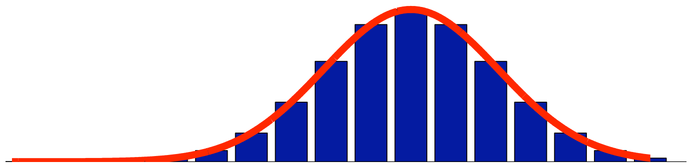


Implementations of LDLC Decoders

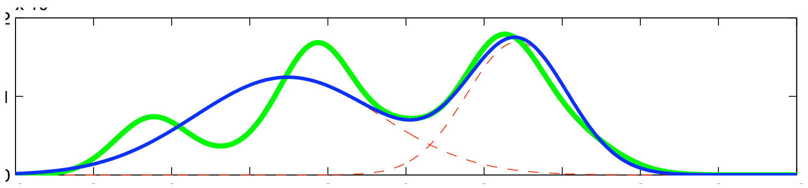
How to represent messages, which are functions?

0. **Exact** True message is a mixture of Gaussians, but number is exponential in iterations.

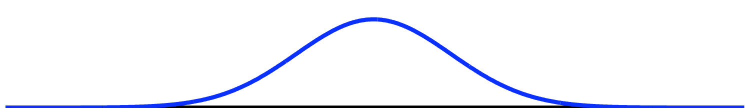
1. **Quantization** of the the messages [Sommer et al.]
   - About 1024 quantization points gets good performance



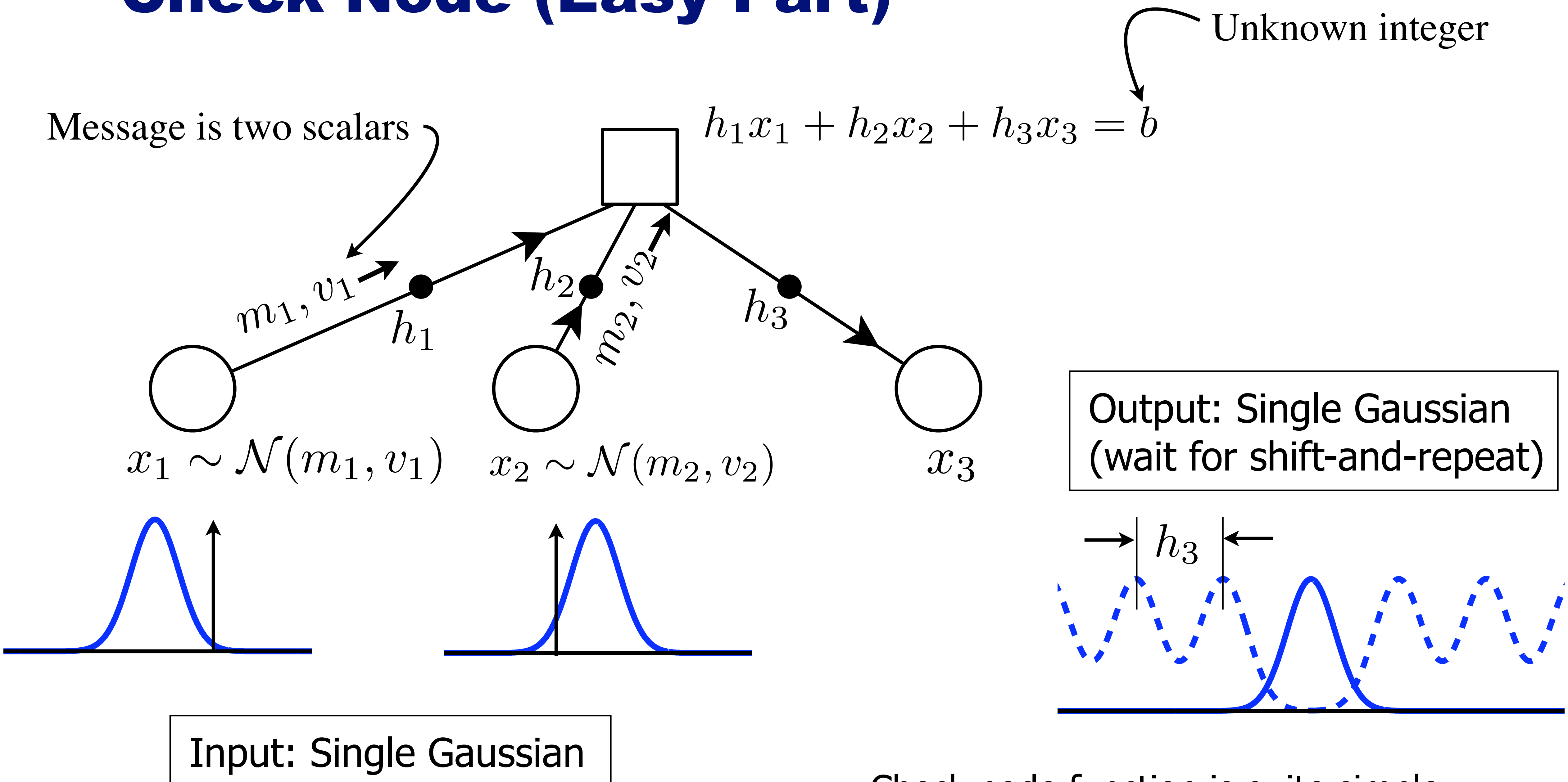
2. **Mixture of Gaussians** — Approximate the message with a mixture of Gaussians [joint work with Justin Dauwels]
   - Uses a "Gaussian Mixture Reduction Algorithm"
   - preserves Gaussian nature of messages, low memory requirements.
   - Further improvements: Yona and Feder, ISIT 2009



3. **Single Gaussian (this talk)** Approximate the message with a single Gaussian



Single Gaussian Decoder: Check Node (Easy Part)



Check node function is quite simple:

$$
\begin{aligned}
m_3 &= -\frac{h_1 m_1 + h_2 m_2}{h_3} \\
v_3 &= \frac{h_1^2 v_1 + h_2^2 v_2}{h_3^2}
\end{aligned}
$$

Moment Matching — Gaussian Approximation

Let X be the mixture of two Gaussians X_1 and X_2:

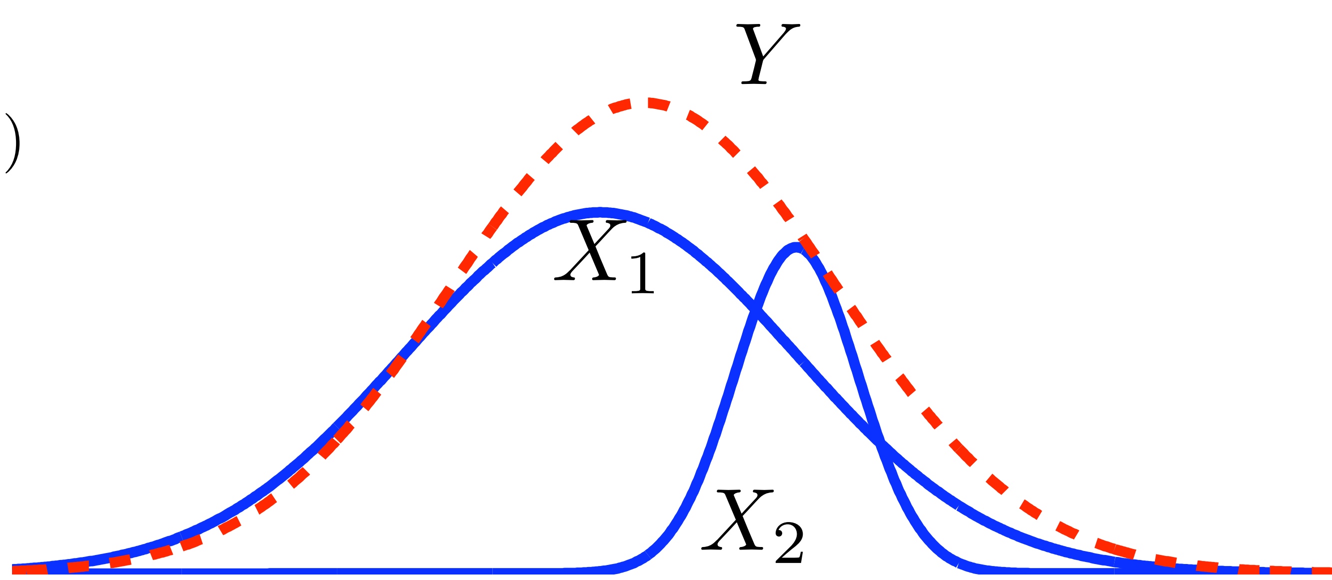
$$\begin{aligned} X &= X_1 + X_2 \\ &= c_1\mathcal{N}(m_1, v_1) + c_2\mathcal{N}(m_2, v_2). \end{aligned}$$

Let Y be the single Gaussian (mean m_Y, variance v_Y) which approximates X. Then, $\mathrm{KL}(X||Y)$ is minimized when:

$$\begin{aligned} E[Y] &= E[X] \quad \text{(first moment)} \\ E[Y^2] &= E[X^2] \quad \text{(second moment)} \end{aligned}$$

Then,

$$\begin{aligned} m_Y = E[Y] &= c_1 m_1 + c_2 m_2 \\ E[Y^2] &= \sum_i c_i \cdot (v_i + m_i^2) \end{aligned}$$

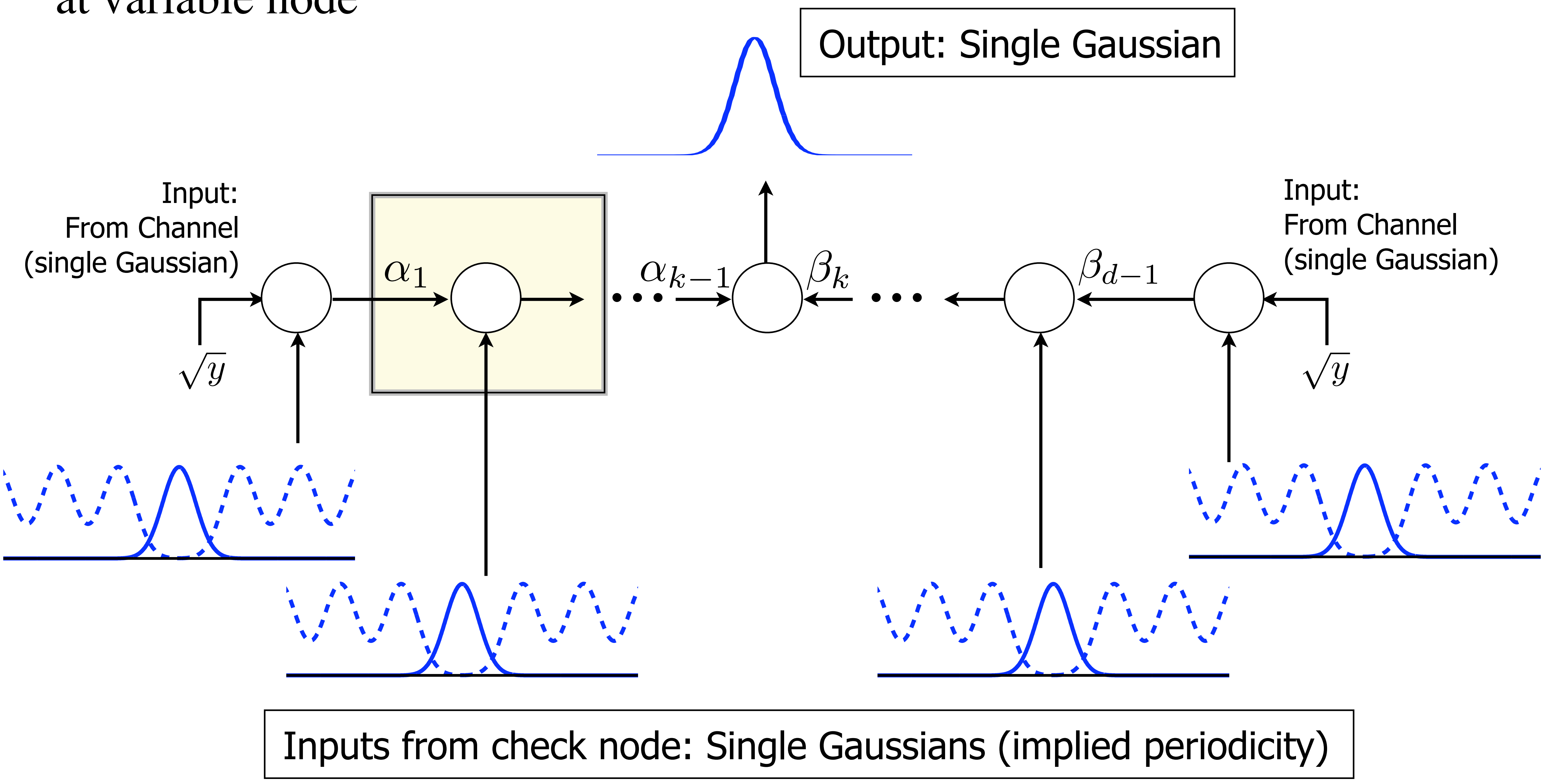


Of course:

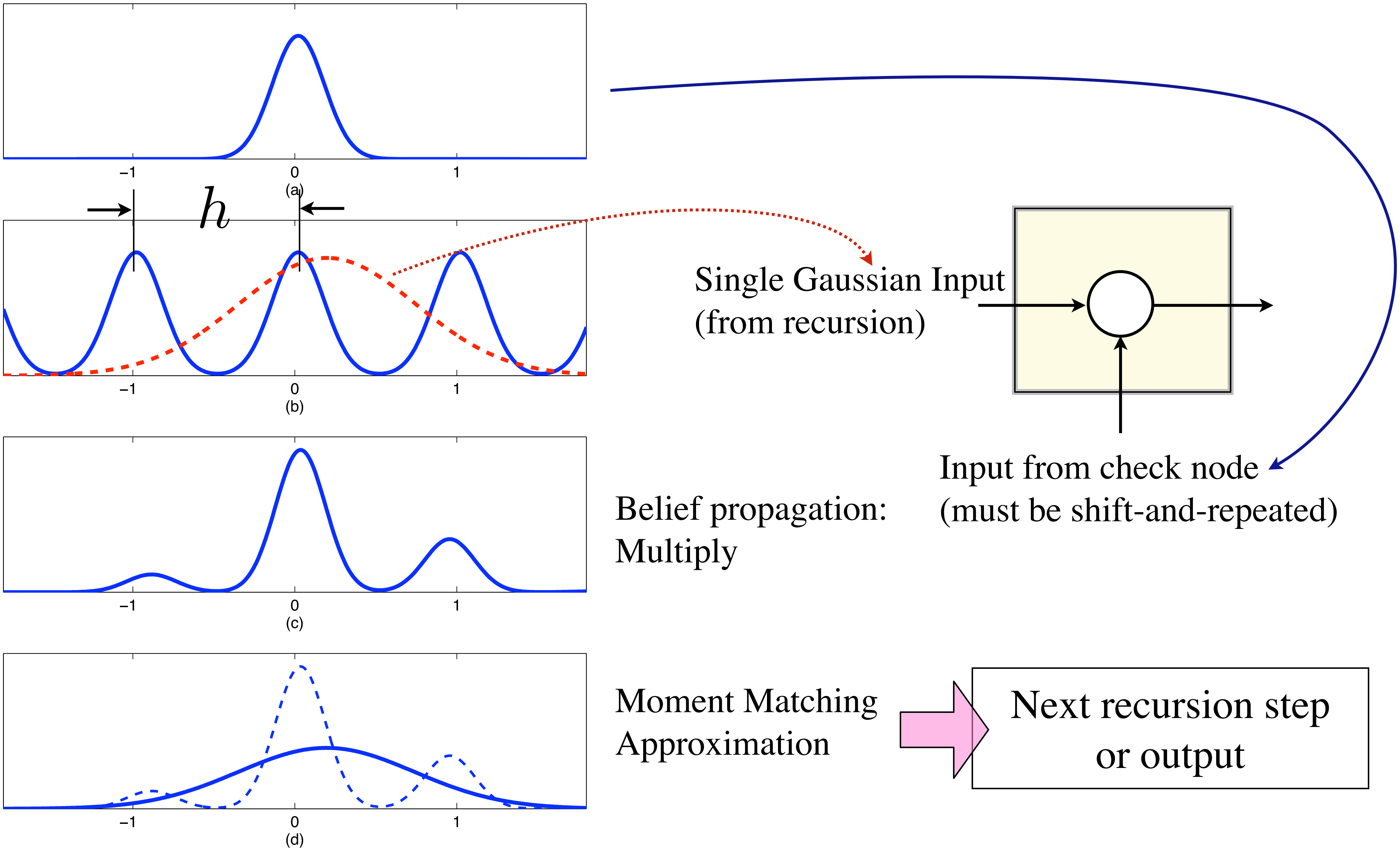
$$v_Y = E[Y^2] - E[Y]^2$$

Single Gaussian Decoder: Variable Node (Hard Part)

Forward-backward algorithm at variable node



Variable Node One-Step Function



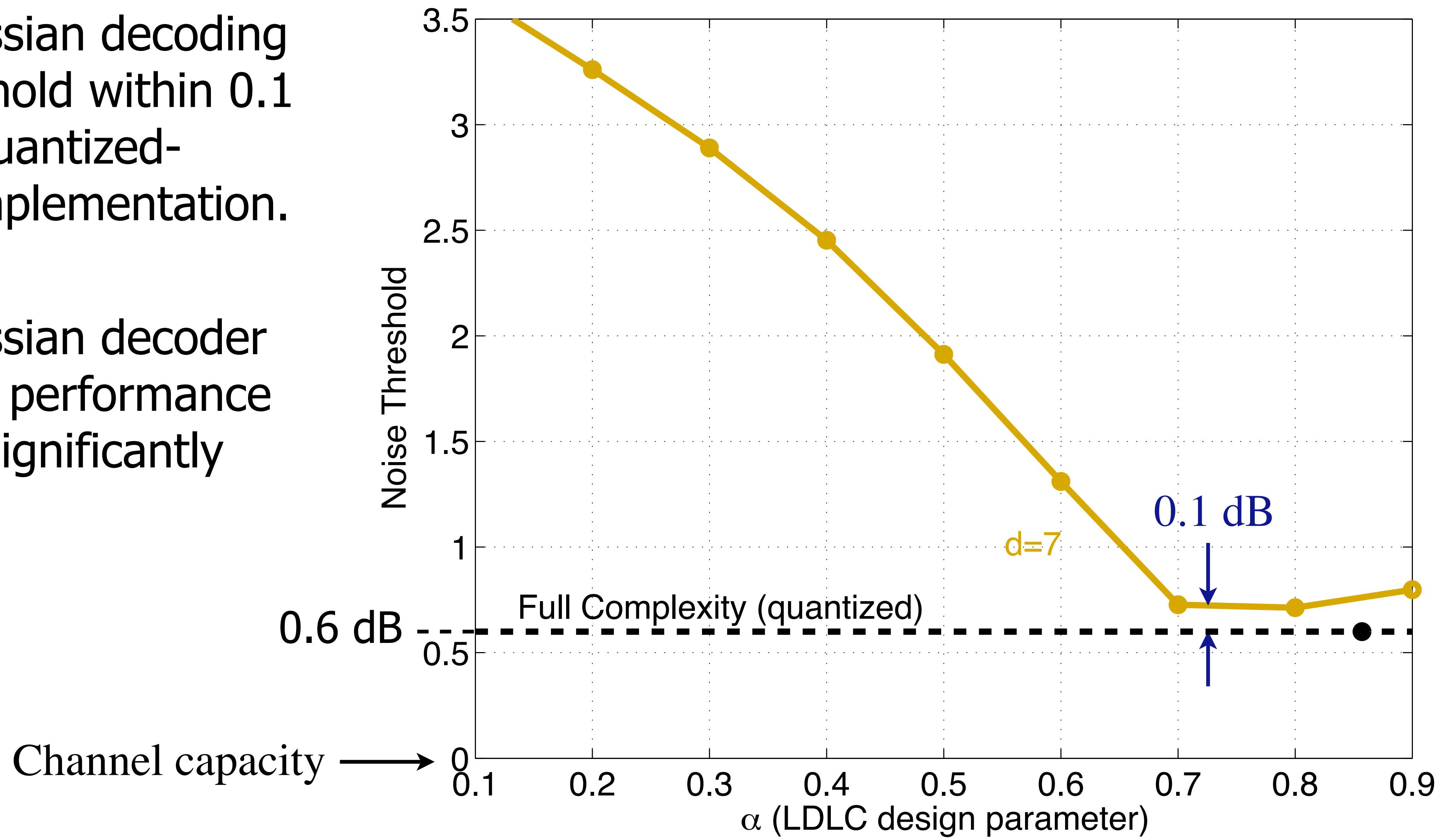
Complexity Measure: Can you write the decoding rule on one slide?

<table>
<tr><th></th><th>Variable Node</th><th>Check Node</th></tr>
<tr><td>Quantized message</td><td>128-point DFT</td><td>Sum of 1024 numbers</td></tr>
<tr><td>Multiple Gaussians</td><td colspan="2">Gaussian Mixture Reduction:
• Compute pair-wise distance between M^2 Gaussians
• Repeatedly combine Gaussians</td></tr>
<tr><td>Single Gaussian</td><td>$$\begin{aligned} m &= v' \sum_{b \in \mathcal{B}} \Big(\frac{b}{v_{\mathsf{c}} h} + \frac{m_{\mathsf{c}}}{v_{\mathsf{c}}} + \frac{m_{\mathsf{a}}}{v_{\mathsf{a}}}\Big) \exp\Big(-\frac{1}{2}\frac{(b/h + m_{\mathsf{c}} - m_{\mathsf{a}})^2}{v_{\mathsf{c}} + v_{\mathsf{a}}}\Big) \\ v &= v' - m^2 \\ &\quad + v'^2 \sum_{b \in \mathcal{B}} \Big(\frac{b}{v_{\mathsf{c}} h} + \frac{m_{\mathsf{c}}}{v_{\mathsf{c}}} + \frac{m_{\mathsf{a}}}{v_{\mathsf{a}}}\Big)^2 \exp\Big(-\frac{1}{2}\frac{(b/h + m_{\mathsf{c}} - m_{\mathsf{a}})^2}{v_{\mathsf{c}} + v_{\mathsf{a}}}\Big) \end{aligned}$$</td><td>$$\begin{aligned} m &= m_{\mathsf{a}} + m_{\mathsf{c}} \\ v &= v_{\mathsf{a}} + v_{\mathsf{c}} \end{aligned}$$</td></tr>
<tr><td>Binary LDPC (log domain)</td><td>$$x_{\mathsf{a}} + x_{\mathsf{c}}$$</td><td>$$\mathrm{sgn}(x_{\mathsf{a}}, x_{\mathsf{b}}) \cdot 2 \tanh^{-1}\Big(\log\Big|\tanh\frac{x_{\mathsf{a}}}{2}\Big| + \log\Big|\tanh\frac{x_{\mathsf{b}}}{2}\Big|\Big)$$</td></tr>
</table>

- Single-Gaussian decoder is considerably simpler that previous methods
- Single-Gaussian decoder is more complicated than binary LDPC, but at least the decoding rule fits on the page.

Noise Thresholds — Monte Carlo Density Evolution

- Single-Gaussian decoding noise threshold within 0.1 dB of the quantized-message implementation.
- Single-Gaussian decoder has a slight performance loss but is significantly simpler.



Noise Thresholds — Regular Code Design

LDLC sparse matrix non-zero entries:

$$h_1 \geq h_2 \geq \cdots \geq h_d$$

Characterized by parameter α:

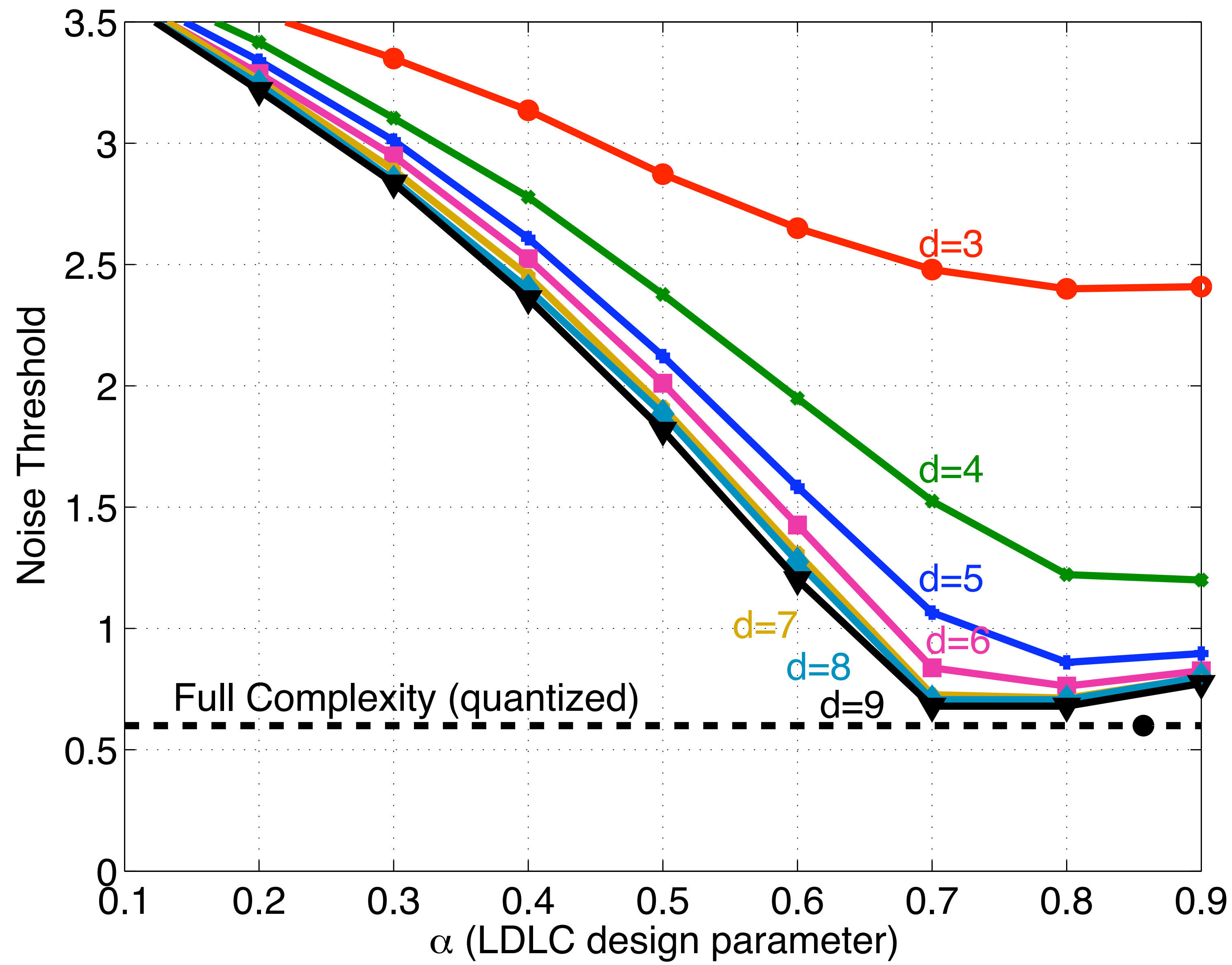
$$\alpha = \frac{h_2^2 + h_3^2 + \cdots + h_d^2}{h_1^2} \geq 0$$

Theorem [Sommer, et al.]: Convergence is exponentially fast for $\alpha \leq 1$.

It was noted,

$$\alpha \quad \rightarrow \quad 1$$

gave good performance.



- Even alpha =0.7 is slightly better that other higher values.
- Increasing d beyond 7 gives little benefit (same as Sommer et al.)

Conclusion

- **Proposed** a single-Gaussian decoder for LDLC lattices
- **Complexity** is much lower than previous methods — we could write the decoding rule on one slide
- **Performance:** Minimal loss, noise threshold 0.1 dB away from more complicated decoder.
- **Code Design** Regular codes found that alpha->1 wasn’t always best, slight gain by picking alpha = 0.7

What about finite or smaller dimension?

"Low" dimension lattices (dimension 100-1000)

- Single Gaussian numerical results poor.
- Multiple Gaussian decoder works OK at low dimension.

High dimension (10000 to infinity)
Single Gaussian OK

Low Dimension (100-1000)
More Gaussians
(Performance loss with single Gaussian)

- Suggestion: modify BP decoder to improve convergence at low dimensions