On Autoencoder-Based Error Correcting Codes

V. Ninkovic, D. Vukobratovic, C. Häger,
H. Wymeersch, A. Graell i Amat

February 8, 2023

Overview

1. Intro & Motivation
2. System Model and Autoencoder (AE)-Based Codes
3. Rateless Codes
4. Unequal Error Protection (UEP) Codes
5. Appendix: More Interesting Results...

Intro & Motivation

Intro & Motivation

- *O'Shea and Hoydis*[1]: Fundamental "new" way to think about communication systems design - End–to–end reconstruction task
- Key idea - Transmitter, channel and receiver are represented as one deep neural network (NN)
- Joint optimization of transmitter and receiver
- Motivation: Conventional communication systems - Split the signal processing into a chain of multiple independent blocks:
  - Pros: Efficient, versatile, controllable systems
  - Cons: Potential sub–optimal end–to–end performances

[1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.,* vol. 3, no. 4, pp. 563-575, Dec. 2017.

System Model and Autoencoder (AE)-Based Codes

System Model

- We consider a problem of communicating a message m from a set of messages $\mathcal{M} = \{1, 2, \ldots, M\}$ over a noisy channel

- Each message is represented as a sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_k)$ of $k = \log_2(M)$ bits

- Encoder mapping: $f : \mathcal{M} \to \mathbb{R}^n$ encodes m into $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$
  - Codewords obey:
    - Fixed power constraint $\|\boldsymbol{x}\|_2^2 = n$
    - Average power constraint $\frac{1}{M} \sum_{i=1}^{i=M} \|\boldsymbol{x}_i\|_2^2 = n$

- The code rate $R = k/n$ [bits/channel use]

System Model

- The channel $\mathcal{W}$ transforms $\boldsymbol{x} \in \mathbb{R}^n$ into $\boldsymbol{y} \in \mathbb{R}^n$ following the probabilistic channel law $p(\boldsymbol{y}|\boldsymbol{x})$
- Decoder mapping: $g : \mathbb{R}^n \to \mathcal{M}$ produces an estimate $\hat{m}$ of the transmitted message m
- Goal: design a pair (f, g) for a given channel $\mathcal{W}$ to minimize the average error probability:

$$P_{\mathrm{e}} = \frac{1}{M} \sum_{m \in \mathcal{M}} \mathbb{P}\{\hat{m} \neq m | m\}. \tag{1}$$

Solution Using Deep Autoencoders

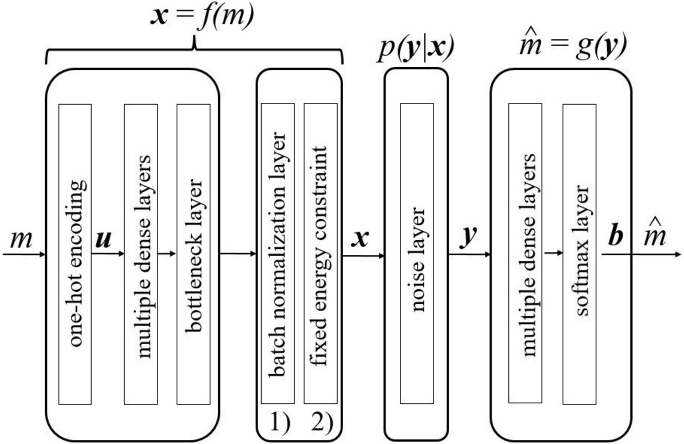
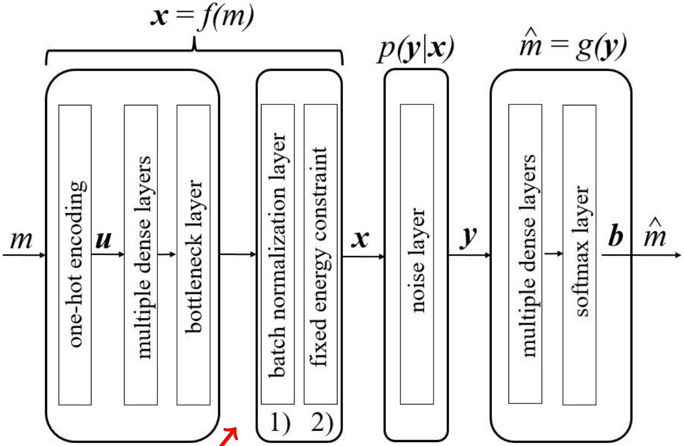


Figure 1: Communication system represented as a deep autoencoder [2]

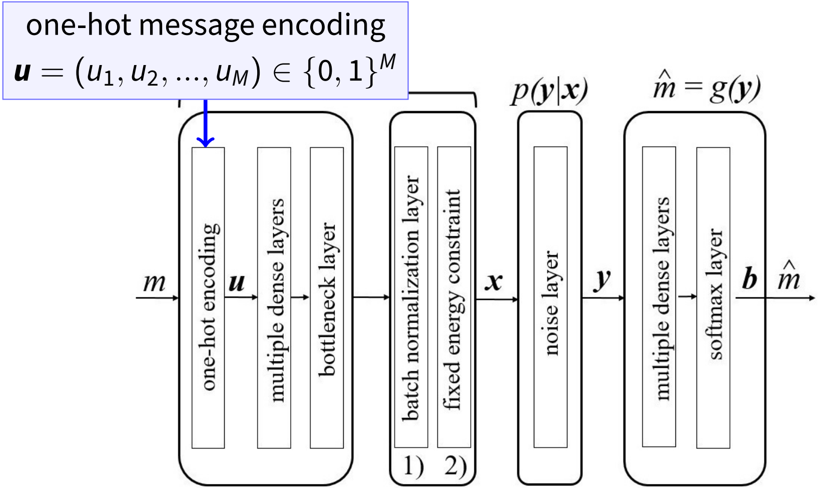
[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.,* vol. 3, no. 4, pp. 563-575, Dec. 2017.

Solution Using Deep Autoencoders

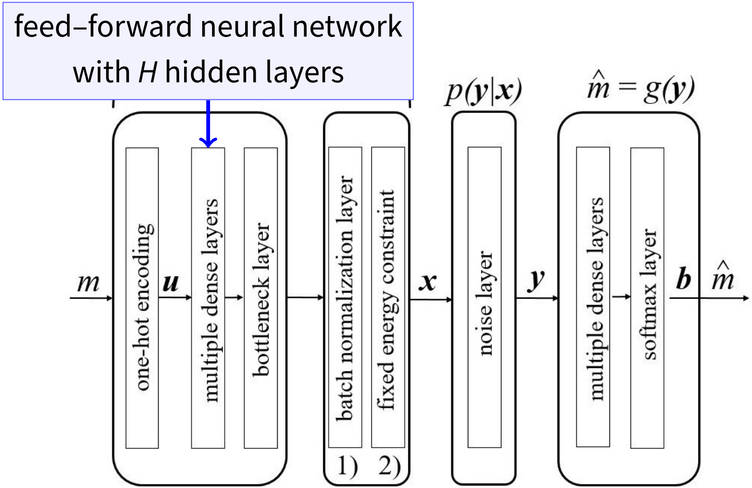


set of encoder layers

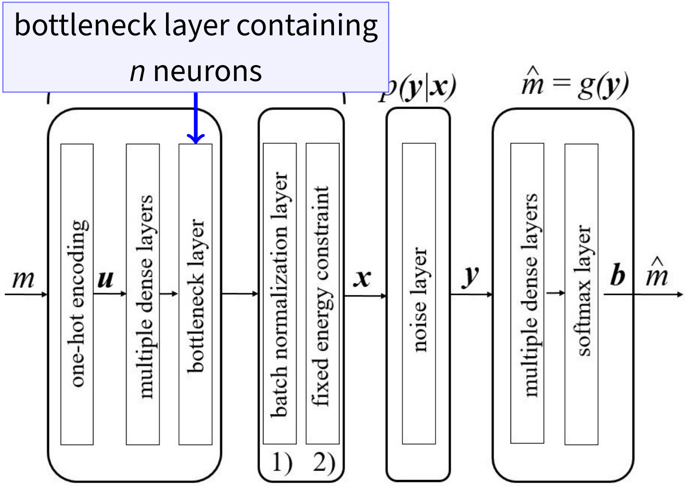
Solution Using Deep Autoencoders



Solution Using Deep Autoencoders

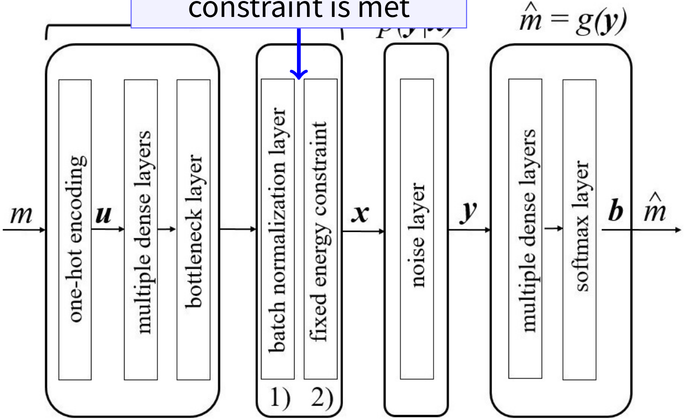


Solution Using Deep Autoencoders

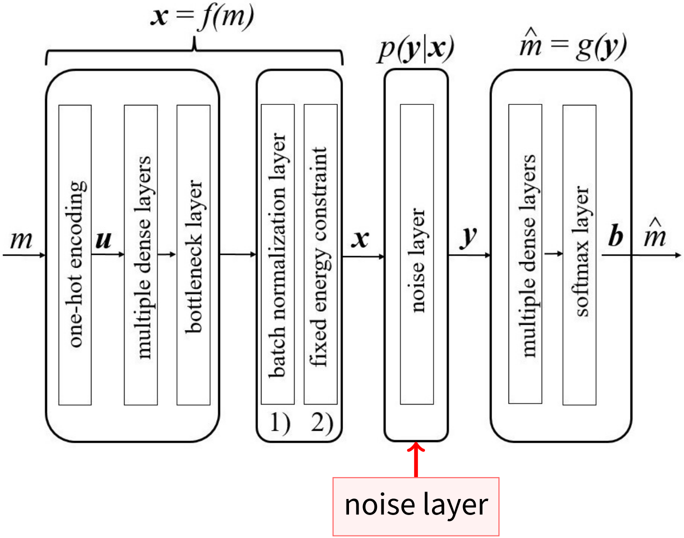


Solution Using Deep Autoencoders

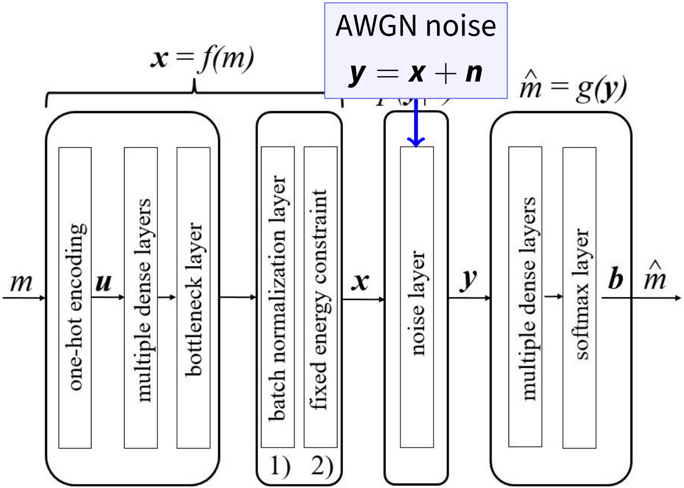
ensures that the energy constraint is met



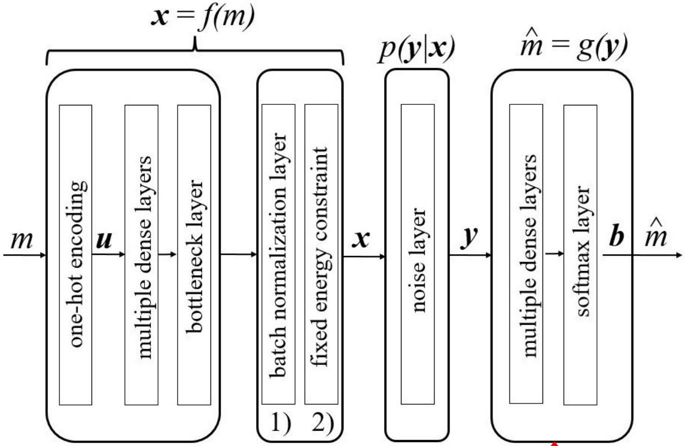
Solution Using Deep Autoencoders



Solution Using Deep Autoencoders

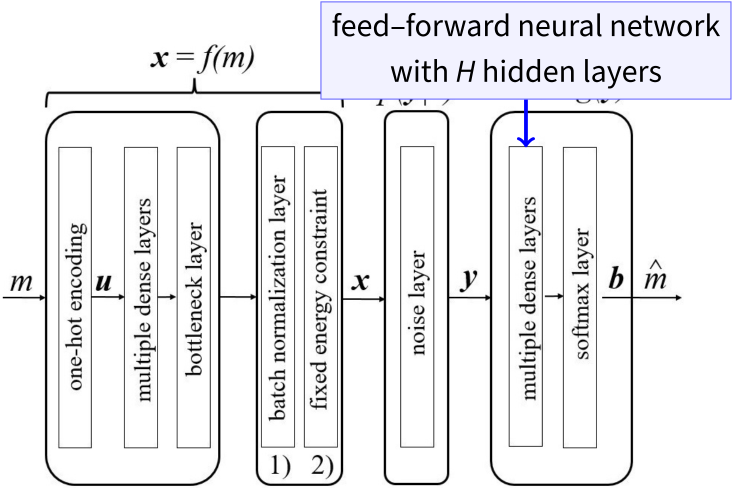


Solution Using Deep Autoencoders



set of decoder layers

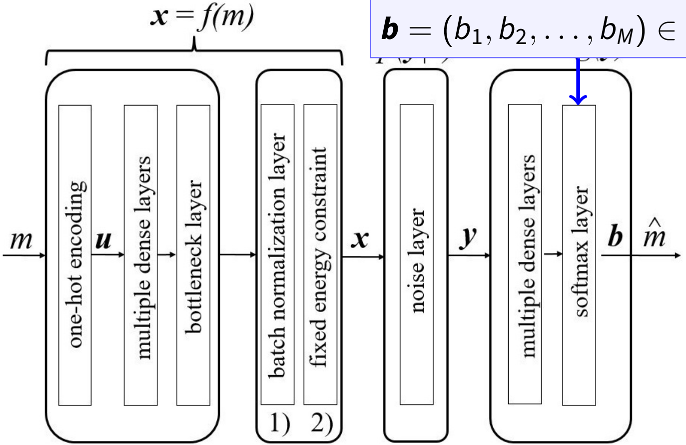
Solution Using Deep Autoencoders



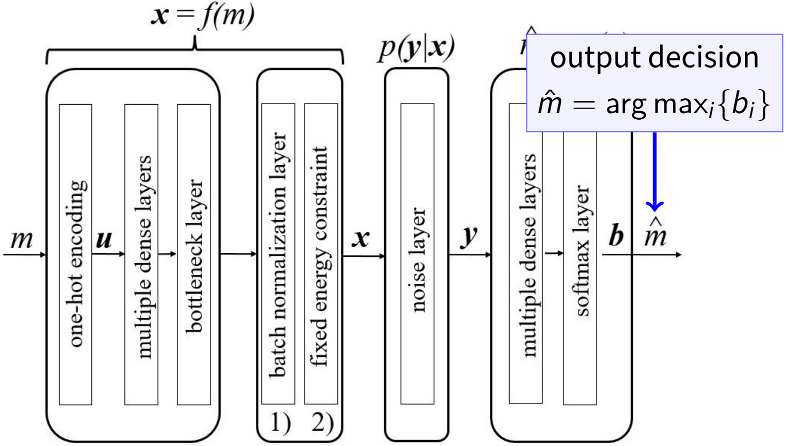
Solution Using Deep Autoencoders

softmax layer

$\boldsymbol{b} = (b_1, b_2, \ldots, b_M) \in (0, 1)^M$



Solution Using Deep Autoencoders



Autoencoder Training

- Autoencoder (AE) is trained in end-to-end manner
  - Stochastic gradient descent (SGD) with Adam optimizer
- **Loss function**: Cross-entropy is used as a surrogate for message error probability:

$$\ell(\boldsymbol{u}, \boldsymbol{b}) = -\sum_{i=1}^{M} u_i \log b_i, \quad (2)$$

- AE is trained using a batches of training data by minimizing cross-entropy loss function averaged across a batch of samples

AE–Based Codes vs Conventional Codes

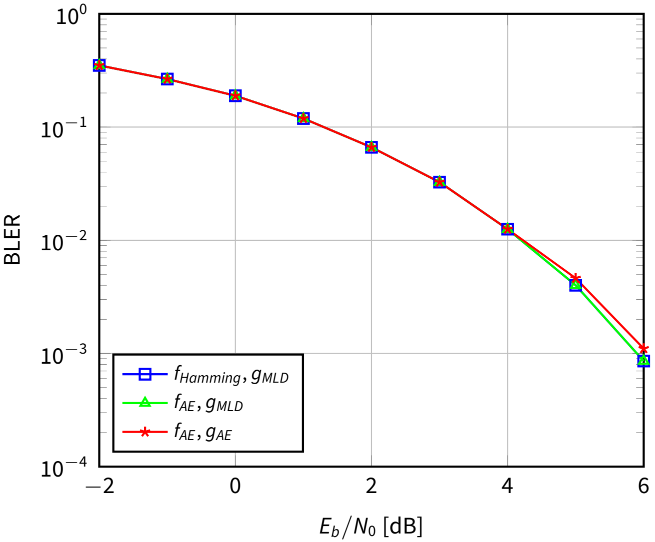


Figure 2: Conventional versus AE–based codes

Rateless Codes

Rateless Codes - Channel Model

- Motivation - Receiver is able to trade off decoding delay against error probability
- The error probability decreases with each received symbol – Delay increases
- Cascade of AWGN channel and erasure channel - **Tail erasures** (dying channel) [3]
- Scenarios where reception of codeword may be interrupted:
  - Deep fade
  - Loss of synchronization, lack of memory
  - Depletion of harvested energy
  - Satelite communications (loss of LEO satellite)

[3] L.R. Varshney, S.K. Mitter, and V.K. Goyal, "An information-theoretic characterization of channels that die," *IEEE Trans. Inf. Theory,* vol. 58, no. 9, pp. 5711-5724, Sept. 2012.

Tail Erasure Channel

- L channel states - p_ℓ denotes the probability that the channel is in the ℓ-th state
- Erasure channel distribution:

$$\boldsymbol{p} = \{p_1, p_2, \ldots, p_L\}, \sum_{i=\ell}^{L} p_\ell = 1 \tag{3}$$

- Receiver receives first r_ℓ symbols of $\boldsymbol{y}$, $n - r_\ell$ symbols are erased
- ℓ-th channel state is defined by (p_ℓ, r_ℓ) pair:

$$\boldsymbol{r} = \{r_1, r_2, \ldots, r_L\} \tag{4}$$

- Receiver in the ℓ-th state receives:

$$\boldsymbol{y}_\ell = \{y_1, y_2, \ldots, y_{r_\ell}\} \tag{5}$$

Autoencoder-Based Rateless Codes

- Novel class of AE codes that allow to trade off decoding delay and reliability is introduced - ***Rateless AE codes***
- ***Randomized dropout strategy*** - Match the AE-based code design to a given erasure channel model by using binary dropout vector $\boldsymbol{d}$

$$\boldsymbol{d} = (d_1, d_2, \ldots, d_n),\ d_i \in \{0, 1\}, \quad (6)$$

- Channel models with multiple states (L > 1) - Sequence of dropout vectors $\boldsymbol{d}_\ell, \ell \in \{1, 2, \ldots, L\}$ is defined, $\boldsymbol{d}_\ell$ corresponds to the ℓ-th class
- Rateless AE training process - Different dropout vectors are applied on **batch-by-batch** basis (randomized dropout strategy):
  - Randomly sample a dropout class $\ell \in \{1, 2, \ldots, L\}$ from the dropout class probability distribution $\boldsymbol{q}$
  - Apply the dropout vector $\boldsymbol{d}_\ell$

Autoencoder-Based Rateless Codes

- Channel with **Tail Erasures** – Topmost r_ℓ positions (neurons) of the ℓ-th class dropout vector $\boldsymbol{d}_\ell$ are set to ones, others to zero

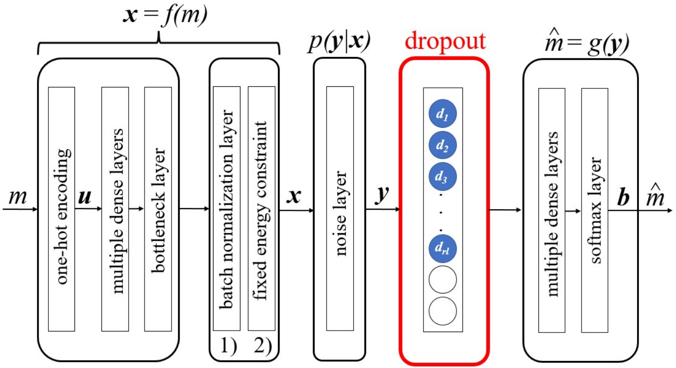


Figure 3: Communication system represented as a deep autoencoder with additional dropout layer -**Tail erasures**

Numerical Results

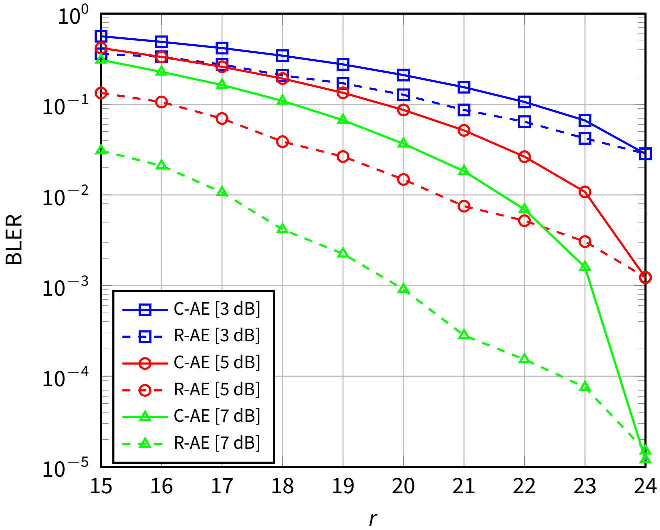


Figure 4: R-AE versus C-AE BLER performances as a function of the number of received symbols ($(n, k) = (24, 12)$).

Numerical Results

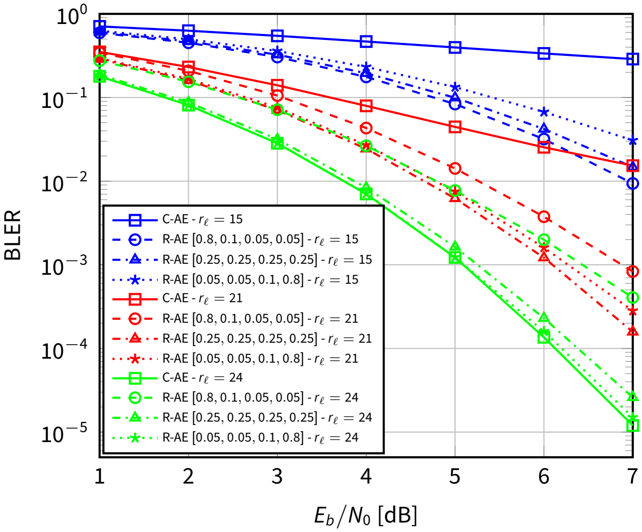


Figure 5: Rateless AE (R-AE) versus Conventional AE (C-AE) decomposed BLER performances for different erasure channel state distributions $\boldsymbol{p}$ ($(n, k) = (24, 12)$).

Numerical Results

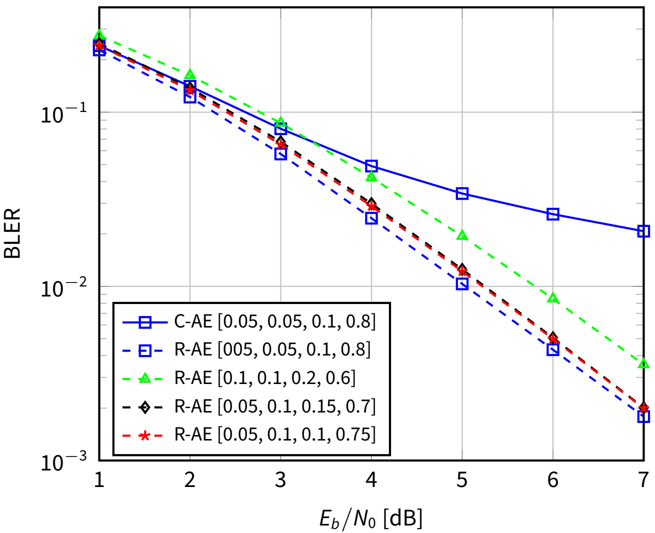


Figure 6: R-AE versus mismatched R-AE codes (Model 1, $(n, k) = (24, 12)$).

Unequal Error Protection (UEP) Codes

Autoencoder-Based UEP Codes

- We present a simple, flexible and efficient method to design AE-based UEP codes
- The key idea is to define appropriate compound **loss function** that generalizes the cross-entropy loss function to the UEP case
- Message-wise UEP - Trivial manipulation

Message-Wise AE-Based UEP Codes

- Message set $\mathcal{M}$ partitioned into $C \leq M$ disjoint subsets - Each with different error protection requirements
- Message class $\mathcal{M}_i$ contains $|\mathcal{M}_i| = M_i$ messages, $M = \sum_{i=1}^{C} M_i$
- For a given encoder-decoder pair (f, g) and channel $\mathcal{W}$, we define the per-class probability of error:

$$P_{\mathrm{e}}^{(i)} = \frac{1}{M_i} \sum_{m \in \mathcal{M}_i} \mathbb{P}\{\hat{m} \neq m | m\}. \tag{7}$$

- Collecting per-class error probabilities, we obtain error probability vector $\boldsymbol{P}_{\mathrm{e}} = (P_{\mathrm{e}}^{(1)}, P_{\mathrm{e}}^{(2)}, \ldots, P_{\mathrm{e}}^{(C)})$

Message-Wise AE-Based UEP Codes

- Let $\ell_{\mathcal{M}_j}(\boldsymbol{u}, \boldsymbol{b})$ be the loss function associated to the j-th message class:

$$\ell_{\mathcal{M}_j}(\boldsymbol{u}, \boldsymbol{b}) = -\sum_{i \in \mathcal{M}_j} u_i \log b_i. \tag{8}$$

Redefined loss function for message-wise UEP

$$\ell(\boldsymbol{u}, \boldsymbol{b}) = \sum_{j=1}^{C} \lambda_j \ell_{\mathcal{M}_j}(\boldsymbol{u}, \boldsymbol{b}) \tag{9}$$

- $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_C)$ is a weight vector associated to the message classes, $\sum_{j=1}^{C} \lambda_j = 1$, and $\lambda_j \geq 0$

Bit-Wise AE-Based UEP Codes

- Recall that each message is represented as a binary sequence $\boldsymbol{s} = (s_1, s_2, \ldots, s_k)$
- We assume $\boldsymbol{s}$ consists of C sub-messages representing disjoint sequences of bits $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_C)$:
  - The length of $\boldsymbol{s}_i$ is equal k_i bits and $k = \sum_{i=1}^{C} k_i$
- We denote by $\mathcal{S}_i$ the set of all possible binary sub-messages $\boldsymbol{s}_i$ where $|\mathcal{S}_i| = 2^{k_i}$
- For a sub-message $\boldsymbol{s}_i \in \mathcal{S}_i$, we denote by $\mathcal{M}_{\boldsymbol{s}_i}$ all messages from $\mathcal{M}$ which are consistent with $\boldsymbol{s}_i$
- For a given encoder-decoder pair (f, g) and channel $\mathcal{W}$, we define the set of per-class error probabilities:

$$P_{\mathrm{e}}^{(i)} = \frac{1}{|\mathcal{S}_i|} \sum_{\boldsymbol{s}_i \in \mathcal{S}_i} \mathbb{P}\{\hat{m} \notin \mathcal{M}_{\boldsymbol{s}_i} | m \in \mathcal{M}_{\boldsymbol{s}_i}\}, \tag{10}$$

- Collecting per-class error probabilities, we obtain error probability vector $\boldsymbol{P}_{\mathrm{e}} = (P_{\mathrm{e}}^{(1)}, P_{\mathrm{e}}^{(2)}, \ldots, P_{\mathrm{e}}^{(C)})$

Bit-Wise AE-Based UEP Codes

- **Main idea**: Similar but more involved manipulation of loss function
- We need to extend the definition of one-hot vector $\boldsymbol{u}$ so that it indicates a subset of messages in $\mathcal{M}$ consistent with a given $\boldsymbol{s}_j \in \mathcal{S}_j$
- For every $\boldsymbol{s}_j \in \mathcal{S}_j$, we define $\boldsymbol{u}_{\boldsymbol{s}_j} = (u_1, u_2, \ldots, u_M)$, such that its m-th position is equal 1 if the message m is consistent with $\boldsymbol{s}_j$:
  - Note that $\boldsymbol{u}_{\boldsymbol{s}_j}$ is now a binary vector with 2^{k-k_j} ones
- Let $\ell(\boldsymbol{u}_{\boldsymbol{s}_j}, \boldsymbol{b})$ be the loss function associated to the j-th submessage:

$$\ell(\boldsymbol{u}_{\boldsymbol{s}_j}, \boldsymbol{b}) = -\sum_{i=1}^{M} u_i \log b_i. \tag{11}$$

Bit-Wise AE-Based UEP Codes

- Given the binary sequence representation $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_C)$ of a message $m \in \mathcal{M}$, we define a set of C vectors:

$$\mathcal{U} = \{\boldsymbol{u}_{\boldsymbol{s}_1}, \boldsymbol{u}_{\boldsymbol{s}_2}, \ldots, \boldsymbol{u}_{\boldsymbol{s}_C}\} \tag{12}$$

Loss function for the bit-wise UEP case

$$\ell(\mathcal{U}, \boldsymbol{b}) = \sum_{j=1}^{C} \lambda_j \ell(\boldsymbol{u}_{\boldsymbol{s}_j}, \boldsymbol{b}) \tag{13}$$

- $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_C)$ is a weight vector associated to the message classes, $\sum_{j=1}^{C} \lambda_j = 1$, and $\lambda_j \geq 0$

Numerical Results

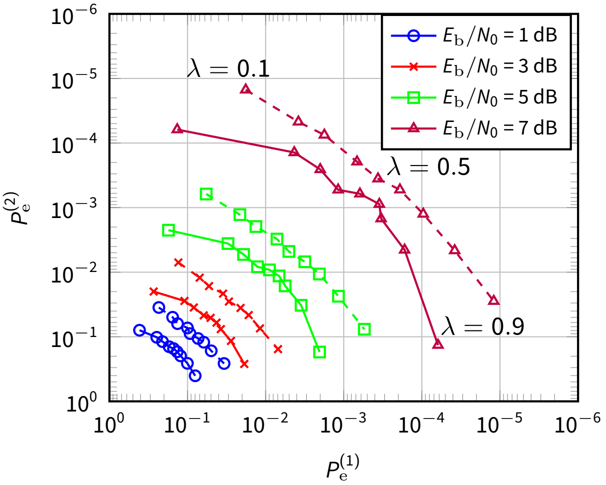


Figure 7: $(P_e^{(1)}, P_e^{(2)})$ performance of AE-based message-wise and bit-wise UEP codes with $C = 2$ [4]

[4] V. Ninkovic, D. Vukobratovic, C. Haeger, H. Wymeersch, and A. Graell i Amat, "Autoencoder-Based Unequal Error Protection Codes," *IEEE Commun. Lett.*, vol. 25, no. 11, pp.3575-3579, Nov. 2021.

Numerical Results - Message-Wise UEP

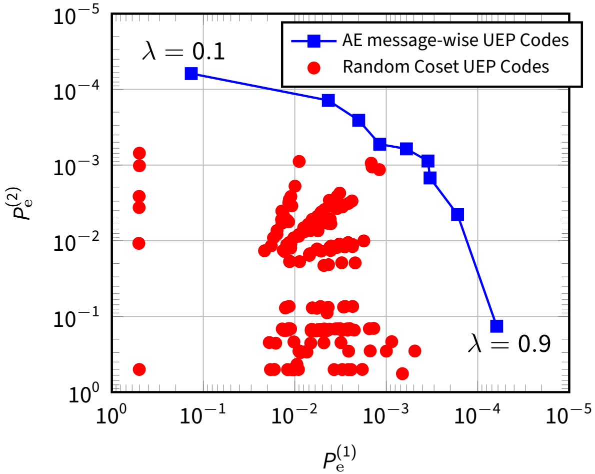


Figure 8: $(P_e^{(1)}, P_e^{(2)})$ performance of AE message-wise UEP codes vs random coset UEP codes [5]

[5] Y. Y. Shkel, V. Y. Tan, and S. C. Draper, "Unequal message protection: Asymptotic and non-asymptotic tradeoffs," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5396-5416, Oct. 2015.

Numerical Results - Bit-Wise UEP

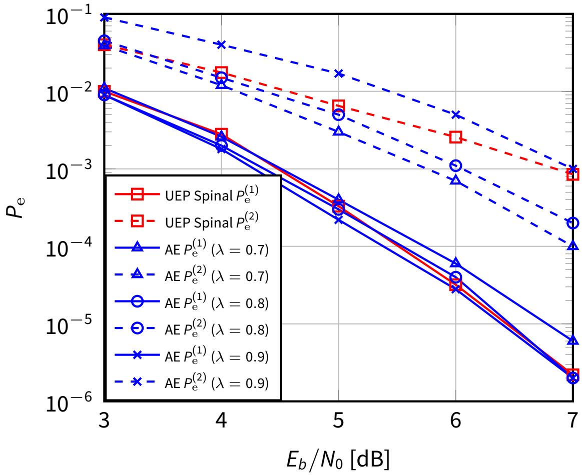


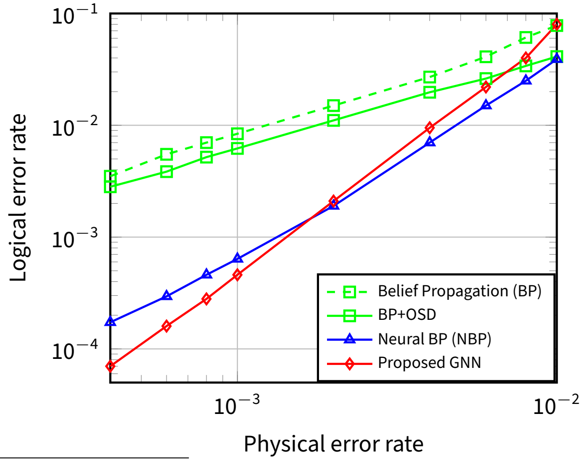
Figure 9: $(P_e^{(1)}, P_e^{(2)})$ vs E_b/N_0 performance of AE-based and spinal bit-wise UEP codes[6]

[6]X. Yu, Y. Li, W. Yang, and Y. Sun, “Design and analysis of unequal error protection rateless spinal codes,” *IEEE Trans. Commun.*, vol. 64, no. 11, pp. 4461-4473, Nov. 2016.

Appendix: More Interesting Results...

Ongoing Work - GNN–based QEC

- Graph neural network (GNN)–based decoding of quantum LDPC codes - Quantum hypergraph–product (hgp) code with code parameter [129, 28] [7]



[7] Y.H. Liu and D. Poulin, "Neural belief–propagation decoders for quantum error–correcting codes," *Phy. Rev. Lett.*, vol. 122, p. 200501, May 2019.

Thank you for your attention!

mail: vukan.ninkovic@gmail.com

Progressive Bit-Wise UEP

- Importance of sub-messages decreases from $\mathcal{S}_1$ to $\mathcal{S}_C$
- Due to inter-dependance, $\boldsymbol{s}_i$ is decoded if all $\boldsymbol{s}_j, j < i$, are also decoded
- For progressive bit-wise UEP, error probability is redefined as:

$$P_{\mathrm{e}}^{(i)} = \frac{1}{|(\mathcal{S}_1, \ldots, \mathcal{S}_i)|} \cdot \sum_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i \in (\mathcal{S}_1, \ldots, \mathcal{S}_i)} \mathbb{P}\{\hat{m} \notin \mathcal{M}_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i} | m \in \mathcal{M}_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i}\}, \tag{14}$$

where $\mathcal{M}_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i}$ is the set of all messages $m \in \mathcal{M}$ whose binary representation $\boldsymbol{s}$ is consistent with $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_i$

Progressive Bit-Wise AE-Based UEP Codes

- For every $\boldsymbol{s}_i \in \mathcal{S}_i$, we define $\boldsymbol{u}_{\boldsymbol{s}_1,\dots,\boldsymbol{s}_i} = (u_1, u_2, \dots, u_M)$, such that its m-th position is equal 1 if the message $m \in \mathcal{M}_{\boldsymbol{s}_1,\dots,\boldsymbol{s}_i}$
- Given a binary sequence representation $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_C)$ of a message $m \in \mathcal{M}$, we define a set of C vectors:

$$\mathcal{U} = \{\boldsymbol{u}_{\boldsymbol{s}_1}, \boldsymbol{u}_{\boldsymbol{s}_1,\boldsymbol{s}_2}, \dots, \boldsymbol{u}_{\boldsymbol{s}_1,\boldsymbol{s}_2,\dots,\boldsymbol{s}_C}\} \tag{15}$$

- We reuse loss function from Eq. 13

Numerical Results - Rateless Codes

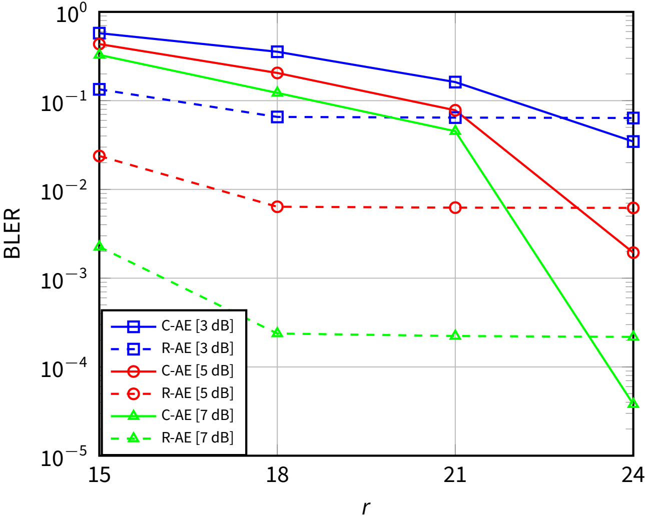


Figure 10: R-AE versus C-AE BLER performances as a function of the number of received symbols - Fixed power constraint $((n, k) = (24, 12))$

Numerical Results - Rateless Codes

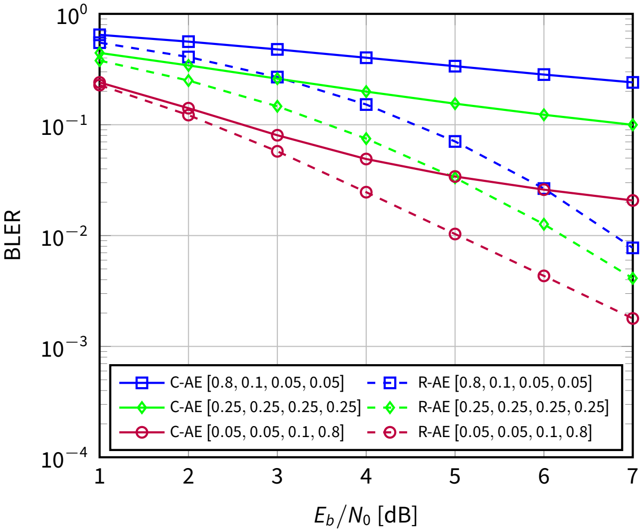


Figure 11: R-AE versus C-AE averaged BLER performances ($(n, k) = (24, 12)$).

Numerical Results - UEP Codes

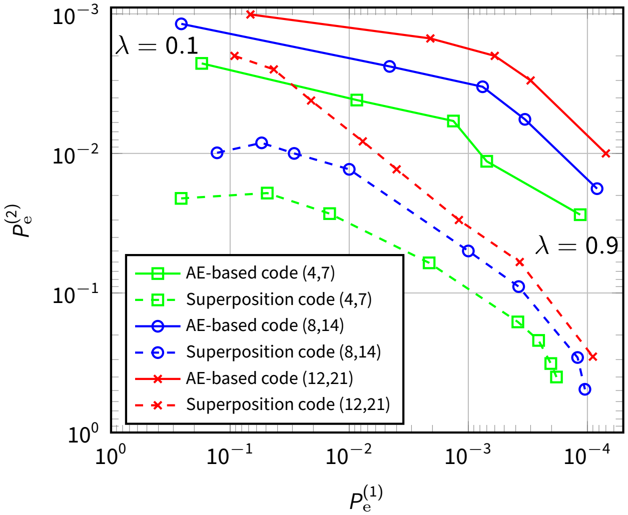


Figure 12: Comparison of $(P_e^{(1)}, P_e^{(2)})$ performance of AE-based and superposition of random Gaussian codes where $k_1 = \frac{1}{4}k$ and $k_2 = \frac{3}{4}k$ at $E_b/N_0 = 5$ dB.

Numerical Results - UEP Codes

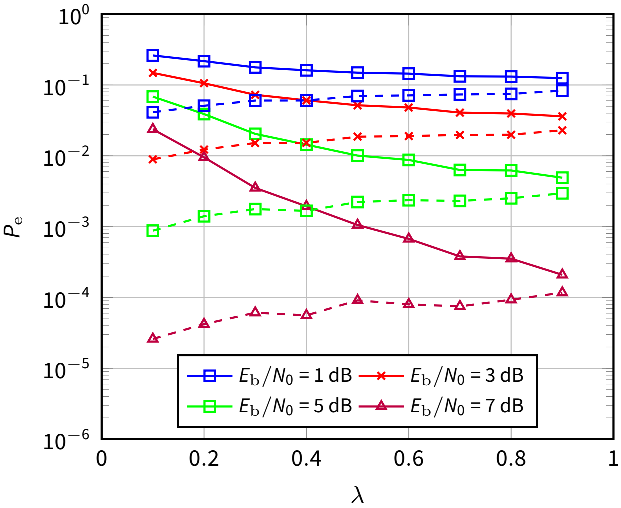


Figure 13: ($P_e^{(1)}$, $P_e^{(2)}$) performance ($P_e^{(1)}$ solid curves, $P_e^{(2)}$ dashed curves) of AE-based progressive bit-wise UEP codes ($C = 2$, $k_1 = 2$, $k_2 = 2$, $n = 7$)