GENERALIZATION BOUNDS FOR NEURAL NORMALIZED MIN-SUM DECODERS

Sudarshan Adiga, Ravi Tandon, Bane Vasić, Tamal Bose
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ, 85719
{adiga, tandonr, vasic, tbose}@arizona.edu
Faculty Advisor: Ravi Tandon

ABSTRACT

Machine learning-based decoding algorithms such as neural belief propagation (NBP) have been shown to improve upon prototypical belief propagation (BP) decoders. NBP decoder unfolds the BP iterations into a deep neural network (DNN), and the parameters of the DNN are trained in a data-driven manner. Neural Normalized Min-Sum (NNMS) and Offset min-sum (OMS) decoders with learnable offsets are other adaptations requiring fewer learnable parameters than the NBP decoder. In this paper, we study the generalization capabilities of the neural decoder when the check node messages are scaled by parameters that are learned by optimizing over the training data. Specifically, we show the dependence of the generalization gap (i.e., the difference between empirical and expected BER) on the block length, message length, variable/check node degrees, decoding iterations, and the training dataset size.

INTRODUCTION

[1] Machine learning has emerged as an important tool for channel encoding and channel decoding. Deep neural networks and reinforcement learning have shown better decoding performance at a given channel signal-to-noise ratio [1, 2, 3, 4]. In another line of work, deep neural networks have been combined with prototypical decoding algorithms to enhance decoding performance when the true channel state information is unavailable [5, 6]. Additionally, it is worth noting that deep neural networks and reinforcement learning algorithms have demonstrated comparable decoding performance to contemporary algorithms but with lower complexity [4, 7].

Iterative decoding algorithms, such as belief propagation (BP), are commonly utilized for decoding linear codes. They are typically considered equivalent to maximum a posteriori (MAP) decoding when the Tanner graph does not contain short cycles [8]. However, in the presence of short cycles within the Tanner graph, BP may prove to be sub-optimal [9, 10]. To address this issue, the Neural

[1]This work was supported by NSF grants CAREER 1651492, CCF-2100013, CNS-2317192, CNS-2209951, CNS-1822071, CIF- 1855879, CCSS-2027844, CCSS-2052751, and NSF-ERC 1941583. Bane Vasić was also supported by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration and funded through JPL's Strategic University Research Partnerships (SURP) Program. Bane Vasić has disclosed an outside interest in Codelucida to the University of Arizona. Conflicts of interest resulting from this interest are being managed by The University of Arizona in accordance with its policies.

Belief Propagation (NBP) decoder was introduced as a technique to mitigate the impact of short cycles [11]. The fundamental concept behind NBP involves scaling the variable node messages, with the weights or scaling factors being learned in a data-driven manner. Other adaptations of Neural Belief Propagation have been proposed [12], where the authors suggest scaling the check node messages instead. Moreover, another approach suggested in [13] involves the use of offsets in the check node messages to reduce the number of multiplications. These offsets are also learned in a data-driven manner.

Understanding the relationship between the generalization gap, which is the difference between the training bit error rate and the test bit error rate on unseen samples, and various parameters such as blocklength, message length, code-rate, and channel SNR, holds significant importance in the field of channel decoding. In our recent work [14], it was proved that the generalization gap of neural belief propagation follows a linear scaling pattern with decoding iterations and variable node degree, with a mild dependence on the blocklength. Moreover, it was found that the generalization gap decreases as the training dataset size increases. Importantly, these empirical observations were consistent across various parity check matrices, aligning with the theoretical findings.

The proof techniques employed in [14] incorporated the PAC-learning approach and leveraged Rademacher complexity tools to establish explicit dependencies on various factors, including the decoder parameters, training dataset size, and channel SNR. Nonetheless, there are other alternative approaches exist for bounding the generalization gap. Within the literature, methods such as computing the Vapnik-Chervonenkis (VC) dimension or computing the Rademacher complexity of the function class have been recognized as viable approaches for upper bounding the generalization gap [15]. Another notable technique is PAC-Bayes analysis, which bounds the generalization gap by quantifying the Kullback-Leibler divergence between the prior and posterior distributions of the learned weights. We next state the main contribution of this paper.

Main Contributions: In this paper, we present an extension of the results established in [14] pertaining to Neural Belief Propagation. Our aim is to derive comprehensive bounds for the generalization gap for neural normalized min-sum decoders (NNMS) and neural offset min-sum decoders (NOMS), specifically by considering the scaling of check node messages. Employing the PAC-learning approach, we explore the relationship between the generalization gap and various factors such as the number of decoding iterations, the size of the training dataset, and the characteristics of the parity check matrix, encompassing blocklength, message length, variable node degree, and check node degree.

SYSTEM MODEL

In Fig. 1, we consider a linear block code denoted as $\mathcal{C}$ with a block length of n and a message length of k. The code $\mathcal{C}$ is characterized by a regular parity check matrix $\mathbf{H} \in \{0,1\}^{(n-k)\times n}$, and Tanner graph $\mathcal{G} = (\mathcal{V}, \mathcal{P}, \mathcal{E})$. $\mathcal{V} = \{v_1, \cdots, v_n\}$ is the set of variable nodes, $\mathcal{P} = \{p_1, \cdots, p_{n-k}\}$ is the set of parity check nodes, and $\mathcal{E} = \{e_1, \cdots, e_{nd_v}\}$ is the set of edges. Here, d_v represents the column weight in the parity check matrix, i.e., the number of parity checks a variable node participates in. The edge connecting variable node v_i to parity check node p_j in Tanner graph $\mathcal{G}$ is denoted as $\{v_i, p_j\}$. $\mathcal{V}(v_j) = \{p_i | \mathbf{H}[i,j] = 1\}$ denote the set of parity check nodes adjacent to the variable node v_j in the Tanner graph $\mathcal{G}$. Similarly, $\mathcal{P}(p_i) = \{v_j | \mathbf{H}[i,j] = 1\}$ denote the set of variable nodes adjacent to the parity check node p_i in $\mathcal{G}$.

Let $\mathcal{Y} \subseteq \mathbb{R}^n$ be the space of n dimensional channel outputs, $\mathcal{X} \subseteq \{0,1\}^n$ be the space of n

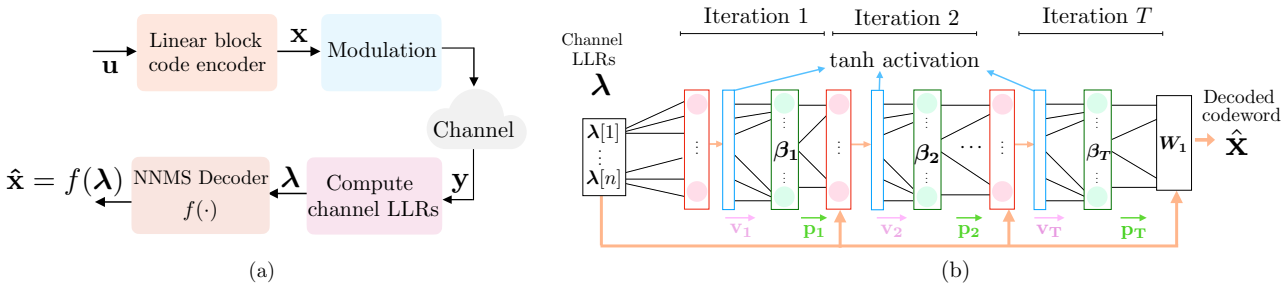

Figure 1: (a) End-to-End block diagram for communication using neural belief propagation (NBP) decoders for linear block codes; (b) Architecture of the NNMS decoder for T decoding iterations where each decoding iteration corresponds: (1) variable node layer, (2) parity check node layer.

dimensional codewords, $\mathcal{U} \subseteq \{0,1\}^k$ be the space of k dimensional messages, and $\mathcal{Z} \subseteq \mathbb{R}^n$ be the space of n dimensional channel noise. The message $\mathbf{u} = [\mathbf{u}[1], \cdots, \mathbf{u}[k]]^\top \in \mathcal{U}$ is encoded to the codeword $\mathbf{x} = [\mathbf{x}[1], \cdots, \mathbf{x}[n]]^\top \in \mathcal{X}$. The channel is assumed to be memoryless, described by $\Pr(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} \Pr(\mathbf{y}[i]|\mathbf{x}[i])$. The receiver receives the channel output $\mathbf{y} = [\mathbf{y}[1], \cdots, \mathbf{y}[n]]^\top \in \mathcal{Y}$; which is the codeword $\mathbf{x}$ modulated, and corrupted with additive noise $\mathbf{z} = [\mathbf{z}[1], \cdots, \mathbf{z}[n]]^\top \in \mathcal{Z}$. The goal of the decoder is to recover the message $\mathbf{u}$ from the channel output $\mathbf{y}$. The input to the decoder is the log-likelihood ratio (LLR) of the posterior probabilities denoted by $\boldsymbol{\lambda} \in \mathbb{R}^{n \times 1}$ and is given as $\boldsymbol{\lambda}[i] = \log \frac{\Pr(\mathbf{x}[i]=0|\mathbf{y}[i])}{\Pr(\mathbf{x}[i]=1|\mathbf{y}[i])}$, for $1 \leq i \leq n$. Denote the output of the NNMS decoder with T decoding iterations as $\hat{\mathbf{x}} = f(\boldsymbol{\lambda})$, where $f(\cdot)$ denotes the decoding function.

Similar to the NBP decoder, each decoding iteration t (where, $1 \leq t \leq T$) in the NNMS decoder corresponds to two hidden layers each of width $|\mathcal{E}| = nd_v$, namely: (1) variable layer $\mathbf{v_t}$, (2) parity check layer $\mathbf{p_t}$. The output of the node $\mathbf{v_t}[\{l,m\}]$ in the NNMS decoder corresponds to the message passed from variable node v_l to parity check node p_m in the t-th iteration, and is,

$$\mathbf{v_t}[\{l,m\}] = \boldsymbol{\lambda}[l] + \sum_{m' \in \mathcal{V}(l) \setminus m} \mathbf{p_{t-1}}[\{l,m'\}], \tag{1}$$

where, $\mathbf{p_{t-1}}[\{l,m'\}]$ corresponds to the message passed from the parity check node $p_{m'}$ to the variable node v_l in the $(t-1)$-th iteration. The output of the parity check layer in the min-sum decoder is given as follows,

$$\mathbf{p_t}[\{l,m\}] = \prod_{l' \in \mathcal{P}(m) \setminus l} sign(\mathbf{v_t}[\{l',m\}]) \min_{l' \in \mathcal{P}(m) \setminus l} |\mathbf{v_t}[\{l',m\}]|. \tag{2}$$

However, the messages from the hidden layer that performs parity checks can be scaled, and the corresponding decoder is called the neural normalized min-sum decoder. The output of the parity check hidden layer in the t-th decoding iteration for the NNMS decoder is,

$$\mathbf{p_t}[\{l,m\}] = \boldsymbol{\beta_t}[\{l,m\}] \prod_{l' \in \mathcal{P}(m) \setminus l} sign(\mathbf{v_t}[\{l',m\}])\, \tilde{\mathbf{p}}_\mathbf{t}[\{l,m\}]. \tag{3}$$

where, $\tilde{\mathbf{p}}_\mathbf{t}[\{l,m\}] = \min_{l' \in \mathcal{P}(m) \setminus l} |\mathbf{v_t}[\{l',m\}]|$ and $\boldsymbol{\beta_t}$ are learned in a data-driven manner. The output of the parity check hidden layer in the t-th decoding iteration for NOMS decoder is,

$$\mathbf{p_t}[\{l,m\}] = \prod_{l' \in \mathcal{P}(m) \setminus l} sign(\mathbf{v_t}[\{l',m\}])\, reLu\left(\tilde{\mathbf{p}}_\mathbf{t}[\{l,m\}] - \boldsymbol{\beta_t}[\{l,m\}]\right). \tag{4}$$

The estimated codeword after T decoding iterations in the NNMS decoder is given as,

$$\hat{\mathbf{x}}[l] = s(\mathbf{W}_2^{(T)}[l,l]\boldsymbol{\lambda}[l] + \sum_{m' \in \mathcal{V}(l)} \mathbf{W}_1^{(T)}[l, \{l,m'\}]\mathbf{p_T}[\{l,m'\}]) \tag{5}$$

where, $s(\cdot)$ is the sigmoid activation. The weights and the scaling factors are learnt by training the NNMS decoder (denoted by $f(\cdot)$) to minimize the bit error rate (BER) loss that is defined as,

$$l_{\text{BER}}(f(\boldsymbol{\lambda}), \mathbf{x}) = \frac{d_H(f(\boldsymbol{\lambda}), \mathbf{x})}{n} = \frac{\sum_{j=1}^{n} \mathbb{1}\left(f(\boldsymbol{\lambda})[j] \neq \mathbf{x}[j]\right)}{n}. \tag{6}$$

Here, $d_H(\cdot,\cdot)$ denotes the Hamming distance, and $\mathbb{1}(\cdot)$ denotes the indicator function. In practice, we train the NNMS decoder to minimize the BER loss over the dataset $S = \{(\boldsymbol{\lambda}_j, \mathbf{x}_j)\}_{j=1}^{m}$ comprising of pairs of log-likelihood ratio and its corresponding codeword. Then, we define the empirical risk of f as $\hat{\mathcal{R}}_{\text{BER}}(f) = \frac{1}{m}\sum_{j=1}^{m} l_{\text{BER}}(f(\boldsymbol{\lambda}_j), \mathbf{x}_j)$. The true risk of f is defined as $\mathcal{R}_{\text{BER}}(f) = \mathbb{E}_{\boldsymbol{\lambda},\mathbf{x}}[l_{\text{BER}}(f(\boldsymbol{\lambda}), \mathbf{x})]$. The generalization gap is defined as the difference $\mathcal{R}_{\text{BER}}(f) - \hat{\mathcal{R}}_{\text{BER}}(f)$.

MAIN RESULTS

In this section, we present our main findings regarding the generalization gap analysis for NNMS decoders. We consider the training dataset, denoted as $S = \{(\boldsymbol{\lambda}_j, \mathbf{x}_j)\}_{j=1}^{m}$, and let $\mathcal{F}_T$ be the class of NNMS decoders with T decoding iterations. We make the assumption that the scaling factors of the check node messages are bounded. Specifically, for every (l,m) pair, we have $\boldsymbol{\beta_t}[\{l,m\}] \leq w$; or the 2-norm $\|\boldsymbol{\beta_t}[\{l,m\}]\|_2$ is bounded as $\|\boldsymbol{\beta_t}[\{l,m\}]\|_2 \leq B_\beta$. Similarly, for the weight matrices, the maximum absolute value of the (i,j) coordinates for all weight matrices is bounded by a non-negative constant w. In other words, we have $|\mathbf{W}_1[i,j]| \leq w$ and $|\mathbf{W}_2[i,j]| \leq w$. $\mathbf{W}_1$ is strictly an upper triangular matrix with exactly d_v non-zero entries in every row, while $\mathbf{W}_2$ is a diagonal matrix. We express the bounded norms for any $j-$th row vector in matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ as B_{w_1} and B_{w_2}, respectively. Furthermore, we assume a bound on the input log-likelihood ratios, such that $|\boldsymbol{\lambda}[i]| \leq b_\lambda$ for all $i = 1, \ldots, n$. To quantify the complexity of the function class $\mathcal{F}_T$, we adopt the notion of bit-wise Rademacher complexity, as defined in [14]. It is formulated as follows:

$$R_m(\mathcal{F}_T[j]) \triangleq \mathop{\mathbb{E}}_{\sigma}\left[\sup_{f \in \mathcal{F}_T} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \cdot f(\boldsymbol{\lambda}_i)[j]\right] \tag{7}$$

where σ_i's are i.i.d. Rademacher random variables, i.e., $\Pr(\sigma_i = 1) = \Pr(\sigma_i = -1) = \frac{1}{2}$. The bit-wise Rademacher complexity captures the correlation between the j-th channel output of the NNMS decoder and the Rademacher random variable. Leveraging results from standard PAC-learning techniques, we use the upper bound on the generalization gap for any $\delta \in (0,1)$ with probability at least $1-\delta$, as follows:

$$\mathcal{R}_{\text{BER}}(f) - \hat{\mathcal{R}}_{\text{BER}}(f) \leq \frac{1}{n}\sum_{j=1}^{n} R_m(\mathcal{F}_T[j]) + \sqrt{\frac{\log(1/\delta)}{2m}}, \tag{8}$$

In the subsequent theorem, we present the key outcome of our study, where we establish an upper bound on the bit-wise Rademacher complexity term by utilizing PAC-learning techniques. This

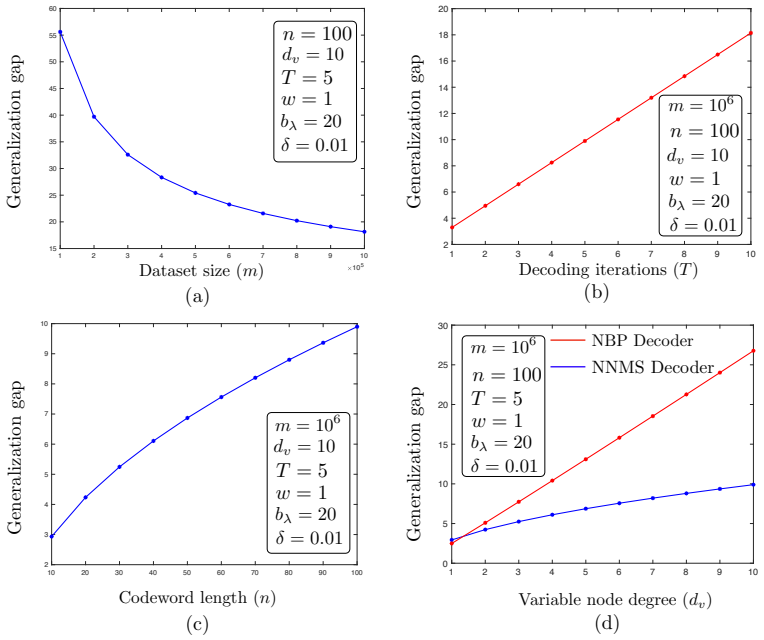

Figure 2: (a) RHS in Theorem 1 vs Dataset size (m), (b) RHS in Theorem 1 vs Decoding iterations (T), (c) RHS in Theorem 1 vs Blocklength (n), (d) RHS in Theorem 1 vs Variable node degree (d_v).

allows us to explicitly capture the dependence on code parameters, the number of decoding iterations, and the size of the training dataset.

Theorem 1. *For any $\delta \in (0,1)$, with probability at least $1-\delta$, the generalization gap for any NNMS decoder $f \in \mathcal{F}_T$ can be upper bounded as follows,*

$$\mathcal{R}_{BER}(f) - \hat{\mathcal{R}}_{BER}(f) \leq \frac{4}{m} + \sqrt{\frac{\log(1/\delta)}{2m}} + 12\sqrt{\frac{nd_v(T+1)^2}{m}\log\left(8\sqrt{m}nwd_vb_\lambda\right)}, \tag{9}$$

where n denotes the blocklength, d_v is the variable node degree, T is the number of decoding iterations (number of layers in NNMS decoder), m is the training dataset size; w and b_λ are upper bounds on the weights in the NNMS decoder and input log-likelihood ratio, respectively.

Proof-sketch of Theorem 1: The detailed proof of Theorem 1 can be found in the Appendix and closely follows the proof of the generalization bounds for NBP decoders [14]. Our approach involves upper bounding the bit-wise Rademacher complexity using the Dudley entropy integral, where the bound depends on the covering number of the function class $\mathcal{F}_T$. Informally, the covering number represents the minimum size of a set that can effectively approximate all members of the function class $\mathcal{F}_T$. We upper bound the covering number of $\mathcal{F}_T$ by considering the product of covering numbers of the weights and the scaling factors. This is based on the observation that the NNMS decoder is Lipschitz with respect to its weights and scaling factors. In what follows, we evaluate the covering number of the weights and scaling factors and express it as a function of the number of parameters and the bounds on the weights and scaling factors. By combining these steps, we derive the result in Theorem 1.

Remark 1 (Generalization Gap Bounds Comparison of NNMS and NBP Decoders). *As seen in Fig. 2 generalization gap bound of NNMS decoder exhibits an inverse relationship with the training dataset size, diminishing according to $\mathcal{O}(1/\sqrt{m})$. The generalization gap grows linearly with the number of decoding iterations, scaling as $\mathcal{O}(T)$. Additionally, the generalization gap is influenced by the blocklength n and the variable node degree d_v, and scales as $\mathcal{O}(\sqrt{n})$ and $\mathcal{O}(\sqrt{d_v})$, respectively. Notably, it is worth mentioning that the generalization gap observed in NBP decoders, as described in [14], displays a linear dependency on the variable node degree, characterized by $\mathcal{O}(d_v)$. This is due to the fact that each extrinsic message (i.e., the message from the variable node to the check node) in the NBP decoder is scaled by distinct weights when incident on different parity check nodes. Conversely, in the NNMS decoder, each message from the parity check node is scaled by a single scaling factor, resulting in a reduced number of parameters and causing the generalization gap to depend on d_v as $\mathcal{O}(\sqrt{d_v})$.*

CONCLUSIONS

In this study, we have investigated the generalization capabilities of neural decoders, specifically focusing on the adaptation of Neural Normalized Min-Sum (NNMS). Our analysis involved scaling the check node messages using learned parameters optimized during training. We explored the impact of various factors, including block length, message length, variable/check node degrees, decoding iterations, and training dataset size, on the generalization gap which is a measure of the disparity between empirical and expected bit error rates (BER). By comparing the generalization gap bounds of the NNMS decoder with those of the NBP decoder, we observed that the gap for NNMS decoder has a mild dependence on the column weight in the parity check matrix (or variable node degree). Extending the current framework to study the generalization gap of different types of machine learning-based codes/decoders (beyond belief propagation) is an important future direction.

REFERENCES

[1] F. Carpi, C. Häger, M. Martalò, R. Raheli, and H. D. Pfister, "Reinforcement learning for channel coding: Learned bit-flipping decoding," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 922–929, IEEE, 2019.

[2] H. Kim, Y. Jiang, R. Rana, S. Kannan, S. Oh, and P. Viswanath, "Communication algorithms via deep learning," *arXiv preprint arXiv:1805.09317*, 2018.

[3] X. Wang, H. Zhang, R. Li, L. Huang, S. Dai, Y. Huangfu, and J. Wang, "Learning to flip successive cancellation decoding of polar codes with lstm networks," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–5, IEEE, 2019.

[4] N. Doan, S. A. Hashemi, and W. J. Gross, "Neural successive cancellation decoding of polar codes," in *2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC)*, pp. 1–5, IEEE, 2018.

[5] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-driven symbol detection via model-based machine learning," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 571–575, IEEE, 2021.

[6] N. Shlezinger, N. Farsad, Y. C. Eldar, and A. J. Goldsmith, "Viterbinet: A deep learning based viterbi algorithm for symbol detection," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3319–3331, 2020.

[7] A. Buchberger, C. Häger, H. D. Pfister, L. Schmalen, and A. G. i Amat, "Learned decimation for neural belief propagation decoders," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8273–8277, IEEE, 2021.

[8] H. Tang, J. Xu, S. Lin, and K. A. Abdel-Ghaffar, "Codes on Finite Geometries," *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 572–596, 2005.

[9] J. Liu and R. C. de Lamare, "Low-Latency Reweighted Belief Propagation Decoding for LDPC Codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1660–1663, 2012.

[10] S. Zhang and C. Schlegel, "Causes and Dynamics of LDPC Error Floors on AWGN Channels," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1025–1032, IEEE, 2011.

[11] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to Decode Linear Codes Using Deep Learning," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 341–346, IEEE, 2016.

[12] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep Learning Methods for Improved Decoding of Linear Codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[13] L. Lugosch and W. J. Gross, "Neural Offset Min-Sum Decoding," in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 1361–1365, IEEE, 2017.

[14] S. Adiga, X. Xiao, R. Tandon, B. Vasic, and T. Bose, "Generalization bounds for neural belief propagation decoders," *arXiv preprint arXiv:2305.10540*, 2023.

[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT press, 2018.

[16] V. Garg, S. Jegelka, and T. Jaakkola, "Generalization and Representational Limits of Graph Neural Networks," in *International Conference on Machine Learning*, pp. 3419–3430, PMLR, 2020.

[17] M. Chen, X. Li, and T. Zhao, "On Generalization Bounds of a Family of Recurrent Neural Networks," *arXiv preprint arXiv:1910.12947*, 2019.

[18] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

PROOF OF THEOREM 1

We employ a PAC-Learning approach to establish an upper bound on the bit-wise Rademacher complexity term $R_m(\mathcal{F}_T[j])$ in terms of training dataset size m and the spectral norm of the weight matrices of the NNMS decoder. This approach is inspired by the reasoning used in generalization bound results for graph neural networks, recurrent neural networks, and NBP decoders presented in [16, 17, 14]. We adopt Lemma A.5. in [18] to bound the bit-wise Rademacher complexity as,

$$R_m(\mathcal{F}_T[j]) \leq \inf_{\alpha>0}\left(\frac{4\alpha}{\sqrt{m}} + \frac{12}{m}\int_{\alpha}^{\sqrt{m}} \sqrt{\log \mathcal{N}(\mathcal{F}_T[j], \epsilon, \|\cdot\|_2)}d\epsilon\right). \tag{10}$$

where, $\mathcal{N}(\mathcal{F}_T[j], \epsilon, \|\cdot\|_2)$ is the covering number of the function class $\mathcal{F}_T$. To upper bound the covering number, we show in Lemma 1 that the NNMS decoder is Lipschitz in its weights and scaling factors. In other words, we have that,

$$\begin{aligned}\|f(\boldsymbol{\lambda})[j] - f'(\boldsymbol{\lambda})[j]\|_2 \leq \rho_{w_2}\|\mathbf{W}_2[j,:] - \mathbf{W}_2'[j,:]\|_2 + \rho_{w_1}\|\mathbf{W}_1[j,:] - \mathbf{W}_1'[j,:]\|_2 \\ + \sum_{i=1}^{T}\rho_{\beta_i}\|\boldsymbol{\beta_i} - \boldsymbol{\beta_i'}\|_2.\end{aligned} \tag{11}$$

where, $\rho_{w_2}, \rho_{w_1}, \rho_{\beta_1}, \cdots, \rho_{\beta_T}$ are Lipschitz parameters are given as,

$$\begin{aligned}\rho_{w_2} &= b_\lambda; \qquad \rho_{w_1} = B_\beta\left(\sqrt{n}b_\lambda\left(\frac{B_\beta^{T-1}-1}{B_\beta - 1}\right) + B_\beta^{T-1}\sqrt{nd_v}b_\lambda\right); \\ \rho_{\beta_i} &= (nd_vB_\beta)^{T-i}\left(\sqrt{n}b_\lambda\left(\frac{B_\beta^{i-1}-1}{B_\beta-1}\right) + B_\beta^{i-1}\sqrt{nd_v}b_\lambda\right).\end{aligned} \tag{12}$$

Therefore, the covering number $\mathcal{N}(\mathcal{F}_T[j], \epsilon, \|\cdot\|_2)$ can be upper-bounded by the product of covering number of the weight matrices and the scaling factors as,

$$\begin{aligned}\mathcal{N}(\mathcal{F}_T[j], \epsilon, \|\cdot\|_2) \leq &\mathcal{N}(\mathbf{W}_1[j,:], \frac{\epsilon}{(T+2)\rho_{w_1}}, \|\cdot\|_2) \times \mathcal{N}(\mathbf{W}_2[j,:], \frac{\epsilon}{(T+2)\rho_{w_2}}, \|\cdot\|_2) \\ &\times \prod_{i=1}^{T}\mathcal{N}(\boldsymbol{\beta_i}, \frac{\epsilon}{(T+2)\rho_{\beta_i}}, \|\cdot\|_2)\end{aligned} \tag{13}$$

The covering number of the weights and the scaling factors of the NNMS decoder f for $1 \leq i \leq T$ can be bounded as follows,

$$\begin{aligned}\mathcal{N}(\mathbf{W}_1[j,:], \frac{\epsilon}{(T+2)\rho_{w_1}}, \|\cdot\|_2) &\leq \left(1 + \frac{2(T+2)B_{w_1}\rho_{w_1}}{\epsilon}\right)^{d_v} \\ \mathcal{N}(\mathbf{W}_2[j,:], \frac{\epsilon}{(T+2)\rho_{w_2}}, \|\cdot\|_2) &\leq \left(1 + \frac{2(T+2)B_{w_2}\rho_{w_2}}{\epsilon}\right) \\ \mathcal{N}(\boldsymbol{\beta_i}, \frac{\epsilon}{(T+2)\rho_{\beta_i}}, \|\cdot\|_2) &\leq \left(1 + \frac{2(T+2)B_{\beta}\rho_{\beta_i}}{\epsilon}\right)^{nd_v}\end{aligned} \tag{14}$$

Substituting (14) in (13), we can loosely upper bound the product of the covering numbers as,

$$\mathcal{N}(\mathcal{F}_T[j], \epsilon, \|\cdot\|_2) \leq \left(1 + \frac{2(T+2)nd_v w b_\lambda}{\epsilon}\right)^{(T+1)^2 nd_v} \tag{15}$$

Computing the integral of the covering number, and substituting in (10), we can obtain the upper bound on the generalization gap as,

$$\mathcal{R}_{\text{BER}}(f) - \hat{\mathcal{R}}_{\text{BER}}(f) \leq \frac{4}{m} + \sqrt{\frac{\log(1/\delta)}{2m}} + 12\sqrt{\frac{nd_v(T+1)^2}{m}\log\left(8\sqrt{m}nwd_v b_\lambda\right)}, \tag{16}$$

This concludes the proof of Theorem 1.

LIPSCHITZNESS OF NNMS DECODER

Lemma 1. *For n length codeword, the bit-wise output of the NNMS decoder $f \in \mathcal{F}_T$ is Lipschitz in its weight matrices $\mathbf{W}_1, \mathbf{W}_2$ and scaling factors $\boldsymbol{\beta_1}, \cdots, \boldsymbol{\beta_T}$ such that,*

$$\begin{aligned}\|f(\boldsymbol{\lambda})[j] - f'(\boldsymbol{\lambda})[j]\|_2 \leq \rho_{w_2} \|\mathbf{W}_2[j,:] - \mathbf{W}_2'[j,:]\|_2 + \rho_{w_1} \|\mathbf{W}_1[j,:] - \mathbf{W}_1'[j,:]\|_2 \\ + \sum_{i=1}^{T} \rho_{\beta_i} \|\boldsymbol{\beta_i} - \boldsymbol{\beta_i'}\|_2.\end{aligned}$$

The coefficients ρ_{w_2}, ρ_{w_1}, ρ_{β_1}, $\cdots$, ρ_{β_T} are as follows:

$$\begin{aligned}\rho_{w_2} &= b_\lambda; \qquad \rho_{w_1} = B_\beta \left(\sqrt{n} b_\lambda \left(\frac{B_\beta^{T-1} - 1}{B_\beta - 1}\right) + B_\beta^{T-1}\sqrt{nd_v} b_\lambda\right); \\ \rho_{\beta_i} &= (nd_v B_\beta)^{T-i} \left(\sqrt{n} b_\lambda \left(\frac{B_\beta^{i-1} - 1}{B_\beta - 1}\right) + B_\beta^{i-1}\sqrt{nd_v} b_\lambda\right)\end{aligned} \tag{17}$$

Proof. For outputs $f(\boldsymbol{\lambda})$ and $f'(\boldsymbol{\lambda})$, respectively we consider the following parameter sets: (a) $\mathbf{W}_1$, $\mathbf{W}_2$, $\boldsymbol{\beta_1}, \cdots, \boldsymbol{\beta_T}$, and (b) $\mathbf{W}_1'$, $\mathbf{W}_2'$, $\boldsymbol{\beta_1'}, \cdots, \boldsymbol{\beta_T'}$. For the j-th output in the NNMS decoder, we have that,

$$\begin{aligned}&\|f(\boldsymbol{\lambda})[j] - f'(\boldsymbol{\lambda})[j]\|_2 = \|s\left(\mathbf{W}_2[j,:]\boldsymbol{\lambda}[j] + \mathbf{W}_1[j,:]\mathbf{p_T}\right) - s\left(\mathbf{W}_2'[j,:]\boldsymbol{\lambda}[j] + \mathbf{W}_1'[j,:]\mathbf{p_T'}\right)\|_2 \\ &\leq \|\left(\mathbf{W}_2[j,:] - \mathbf{W}_2'[j,:]\right)\boldsymbol{\lambda}[j] + \mathbf{W}_1[j,:]\mathbf{p_T} - \mathbf{W}_1'[j,:]\mathbf{p_T} + \mathbf{W}_1'[j,:]\mathbf{p_T} - \mathbf{W}_1'[j,:]\mathbf{p_T'}\|_2 \\ &\leq \|\left(\mathbf{W}_2[j,:] - \mathbf{W}_2'[j,:]\right)\boldsymbol{\lambda}[j]\|_2 + \|\left(\mathbf{W}_1[j,:] - \mathbf{W}_1'[j,:]\right)\mathbf{p_T}\|_2 + \|\mathbf{W}_1'[j,:]\left(\mathbf{p_T} - \mathbf{p_T'}\right)\|_2 \\ &\leq \|\mathbf{W}_2[j,:] - \mathbf{W}_2'[j,:]\|_2 b_\lambda + \|\mathbf{p_T}\|_2 \|\mathbf{W}_1[j,:] - \mathbf{W}_1'[j,:]\|_2 + B_{w_1}\|\mathbf{p_T} - \mathbf{p_T'}\|_2.\end{aligned} \tag{18}$$

To further upper bound $\|\mathbf{p_T}\|_2$, we have that, $\|\mathbf{p_T}\|_2 = \|\boldsymbol{\beta_T}\tilde{\boldsymbol{p}}_{\boldsymbol{T}}\|_2 \leq \|\boldsymbol{\beta_T}\|_2 \|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2$. Furthermore, we know from Lemma 2 that $\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2 \leq \|\mathbf{v_T}\|_2$. The norm $\|\mathbf{v_T}\|_2$ can be upper bounded as follows,

$$\begin{aligned}\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2 \leq \|\mathbf{v_T}\|_2 &= \|\boldsymbol{\lambda} + \boldsymbol{\beta_{T-1}}\tilde{\mathbf{p}}_{\mathbf{T-1}}\|_2 \\ &\leq \|\boldsymbol{\lambda}\|_2 + \|\boldsymbol{\beta_{T-1}}\tilde{\mathbf{p}}_{\mathbf{T-1}}\|_2 \\ &\leq \sqrt{n} b_\lambda + B_\beta \|\tilde{\mathbf{p}}_{\mathbf{T-1}}\|_2.\end{aligned} \tag{19}$$

We can further upper bound $\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2$ in terms of $\|\tilde{\mathbf{p}}_{\mathbf{T-2}}\|_2$ as,

$$\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2 \leq \sqrt{n}b_\lambda + B_\beta \left(\sqrt{n}b_\lambda + B_\beta \|\tilde{\mathbf{p}}_{\mathbf{T-2}}\|_2\right) \tag{20}$$

By recursively upper bounding across T decoding iterations, we obtain the upper bound on $\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2$ in terms of $\|\tilde{\mathbf{p}}_{\mathbf{1}}\|_2$ as,

$$\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2 \leq \sqrt{n}b_\lambda \left(1 + B_\beta + \cdots + {B_\beta}^{T-2}\right) + B_\beta^{T-1} \|\tilde{\mathbf{p}}_{\mathbf{1}}\|_2 \tag{21}$$

We can upper bound $\|\tilde{\mathbf{p}}_{\mathbf{1}}\|_2$ in terms of $\|\mathbf{v}_{\mathbf{1}}\|_2$ as $\|\tilde{\mathbf{p}}_{\mathbf{1}}\|_2 \leq \|\mathbf{v}_{\mathbf{1}}\|_2$, and $\|\mathbf{v}_{\mathbf{1}}\|_2$ can be upper bounded as $\|\mathbf{v}_{\mathbf{1}}\|_2 \leq \sqrt{nd_v}b_\lambda$. Substituting in (21) as,

$$\|\tilde{\mathbf{p}}_{\mathbf{T}}\|_2 \leq \sqrt{n}b_\lambda \left(1 + B_\beta + \cdots + (B_\beta)^{T-2}\right) + B_\beta^{T-1}\sqrt{nd_v}b_\lambda \tag{22}$$

We now upper bound the term $\|\mathbf{p}_{\mathbf{T}} - \mathbf{p}'_{\mathbf{T}}\|_2$ as follows,

$$\begin{aligned}
\|\mathbf{p}_{\mathbf{T}} - \mathbf{p}'_{\mathbf{T}}\|_2 &= \|\boldsymbol{\beta}_{\boldsymbol{T}}\tilde{\mathbf{p}}_{\mathbf{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\tilde{\mathbf{p}}'_{\mathbf{T}}\| \\
&\leq \|\boldsymbol{\beta}_{\boldsymbol{T}}\tilde{\mathbf{p}}_{\mathbf{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\tilde{\mathbf{p}}_{\mathbf{T}} + \boldsymbol{\beta}'_{\boldsymbol{T}}\tilde{\mathbf{p}}_{\mathbf{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\tilde{\mathbf{p}}'_{\mathbf{T}}\| \\
&\leq \|\boldsymbol{\beta}_{\boldsymbol{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\|\|\tilde{\mathbf{p}}_{\mathbf{T}}\| + \|\boldsymbol{\beta}'_{\boldsymbol{T}}\|\|\tilde{\mathbf{p}}_{\mathbf{T}} - \tilde{\mathbf{p}}'_{\mathbf{T}}\| \\
&\leq \|\boldsymbol{\beta}_{\boldsymbol{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\|\|\tilde{\mathbf{p}}_{\mathbf{T}}\| + \left(\sqrt{nd_v}\right)^2 B_\beta \|\mathbf{p}_{\mathbf{T-1}} - \mathbf{p}'_{\mathbf{T-1}}\|
\end{aligned} \tag{23}$$

Iterating the steps across T decoding iterations we have that,

$$\begin{aligned}
\|\mathbf{p}_{\mathbf{T}} - \mathbf{p}'_{\mathbf{T}}\|_2 \leq &\|\boldsymbol{\beta}_{\boldsymbol{T}} - \boldsymbol{\beta}'_{\boldsymbol{T}}\|\|\tilde{\mathbf{p}}_{\mathbf{T}}\| + nd_vB_\beta\|\boldsymbol{\beta}_{\boldsymbol{T-1}} - \boldsymbol{\beta}'_{\boldsymbol{T-1}}\|\|\tilde{\mathbf{p}}_{\mathbf{T-1}}\| \\
&+ \cdots + (nd_vB_\beta)^{T-1}\sqrt{nd_v}b_\lambda\|\boldsymbol{\beta}_{\mathbf{1}} - \boldsymbol{\beta}'_{\mathbf{1}}\|
\end{aligned} \tag{24}$$

□