基于深度学习的信道译码的改进

暨宇俊

摘要

在文献 [1] 中重新审视了使用深度神经网络对极化码进行一次性解码的想法，并得出以下结论：(i) 极化码比随机码更容易学习；(ii) 神经网络在极化码编码的训练中能够泛化到从未见过的码字。[1] 中解释说明了神经网络可以学习信道译码，但仅限于 $N \leq 64$ 的情况。在短码场景下，[1] 中使用信息比特真实值训练神经网络，能够接近 MAP 译码的性能。本文尝试使用 MAP 译码值训练神经网络，尝试学习信道的概率规律，同时还对构造方法进行改进。实验结果表明，两种改进方式均能够提升译码性能。

关键词：极化码；信道译码；深度学习

1 引言

近年来，深度学习技术在各个领域取得了显著的成功，包括图像识别、自然语言处理等领域。信道编解码技术主要用于提高信息在信道中的传输可靠性，实现信息在信道中的可靠传输。极化码是近年来所提出的一种编码方式，是第一种能够被严格证明达到信道容量的信道编码方法 [2]。极化码的译码是基于迭代的译码方法，主要包含两类译码方式：(i) 串行译码：如 SC，SCL 等译码方法，他们都是基于比特顺序的译码，无法并行化译码，具有时效性低的缺点；(ii) 并行译码：如 BP 等译码方法，其译码性能通常不如串行译码方法，但可并行化译码，具有时效性高的优点。

论文 [1] 中尝试将深度学习技术应用于信道解码，验证深度学习技术在信道编解码领域的应用是否有效，以及它是否能够达到或超过传统解码算法的性能。如果深度学习能够应用于信道编解码领域，接收端则可以对码字进行一次性解码，代替传统的迭代解码，效率更高。进一步地，基于深度学习的译码若能逼近迭代译码算法的性能，那将解决了迭代译码中译码性能和并行化的问题。

2 相关工作

极化码的主流译码算法是 SC（Successive Cancellation，串行抵消）译码 [2] 及其改进算法，SC 译码的改进是沿着降低复杂度、提升性能的两条路线展开的。在降低复杂度方面，代表性的方法是 SSC（Simplified Successive Cancellation，简化串行抵消）译码 [3] 算法，其基本思路是在码树上引入码率 R=0 与 R=1 的子码，对这些子码上的节点信息直接硬判决。在提升译码性能方面，需要改进码树上的路径搜索机制，主要包括广度优先与深度优先两种搜索机制。SCL（Successive Cancellation List，串行抵消列表）译码 [4] 算法的改进思路是在码

树上进行广度优先搜索，每次搜索不进行判决，而是保留一个小规模的幸存路径列表。另一种改进思路是在码树上采用深度优先搜索，也就是 SCS（Successive Cancellation Stack，串行抵消堆栈）译码 [5] 算法。这种算法按照度量大小顺序，将码树上的路径压入堆栈，每次只扩展度量最大的路径。SCS 和 SCL 译码算法都能够接近 ML（Maximum Likelihood，最大似然）译码性能。SCH（Successive Cancellation Hybrid，串行抵消混合）译码 [6] 是将 SCL 与 SCS 组合，达到复杂度与性能的折中。SCP（Successive Cancellation Priority，串行抵消优先）译码 [7] 也可看作一种 SCL/SCS 算法的组合，通过引入优先级队列机制，减少了路径搜索的次数。此外，上述这些改进算法都可以应用于 CRC-Polar 级联码 [8]，构成 CRC 辅助的译码算法。

90 年代出现了关于使用神经网络进行译码的不同想法。虽然在 [9] 中，输出节点表示码字的位，但也可以对每个码字 [10] 使用一个输出节点。对于汉明编码，另一种变体是只使用特征作为神经网络的输入，以找到最有可能的错误模式 [11]。随后，卷积码的神经网络译码（Neural Network Decoding，NND）出现在 1996 年，当时 Wang 和 Wicker 证明了 NND 与理想的维特比解码器 [12] 的性能相匹配。但他们也提到了 NND 的一个非常重要的缺点：译码问题比传统的模式识别问题有更多的可能性。这就限制了 NND，它只能译码短码。然而，利用递归神经网络 [13] 进一步改进了卷积码的神经网络译码器。NND 在定长码和卷积码方面都没有取得任何大的突破。由于当时的标准训练技术，这两种码不可能与使用大量神经元和层的神经网络一起工作，这也是人们认为神经网络译码不适合译码较长的码字的原因。因此，人们对神经网络的兴趣减少了，不仅是对机器学习本身，而且将其应用在译码方面的兴趣也减少了。在接下来的几年里，有了一些轻微的改进，例如，通过使用随机神经网络 [14] 或通过减少权重 [15] 的数量。2006 年，一种新的训练技术，即逐层无监督预训练，然后是梯度下降微调 [16]，导致了神经网络的复兴，因为它能训练具有更多层的神经网络。具有许多隐藏层的神经网络被称为深度神经网络。如今，强大的新硬件，如图形处理单元（GPU）可以用来加速学习和推理。在神经网络的复兴中，新的 NND 思想出现了。然而，与以前的工作相比，神经网络学习技术仅用于优化已知的解码方案，我们称之为专家知识的引入。例如，在 [17] 中，给定置信度传播（BP）算法的 Tanner 图权重，并通过神经网络技术进行学习，以改进 BP 算法。机器学习的最新进展似乎还没有适应于学习译码的领域。

本文复现及改进的论文 [1] 是最早提出基于全连接神经网络的极化码译码论文。

3 本文方法

3.1 传统极化码译码

极化码译码的基本算法是 SC 译码算法，最早由 Arıkan 提出 [2]。其核心思想是在整个极化过程中进行软硬信息的计算和传递，最后在信源侧进行逐比特判决，它是一种串行的译码算法。对于码长为 N 的极化码，SC 译码器收到接受信号 y_1^N 后，不断逐比特译码 u_i，利用接收信号和前面 $i-1$ 个译码估计值 $\hat{u}_1^{i-1}$ 来译码 u_i，直到译码完 u_N。

对每个极化子信道 $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i)$，其对数似然比定义为：

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) = ln\frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)} \tag{1}$$

当计算出 $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ 后，对比特 u_i 进行译码判决：

$$\hat{u}_i = \begin{cases} 0, & L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 0 \\ 1, & L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) < 0 \\ u_i & i \in A \end{cases} \tag{2}$$

SC 译码算法在每次译码判决时会直接给出当前比特的估计值，这是一种贪心的方式，仅能保证局部最优，对于整个译码过程来说，相比于全局最优的 MAP(Maximum A Posteriori，最大后验概率) 译码，还是有性能上的差距。

3.2 深度学习译码

极化码编码和神经网络译码的结构如图 1所示：

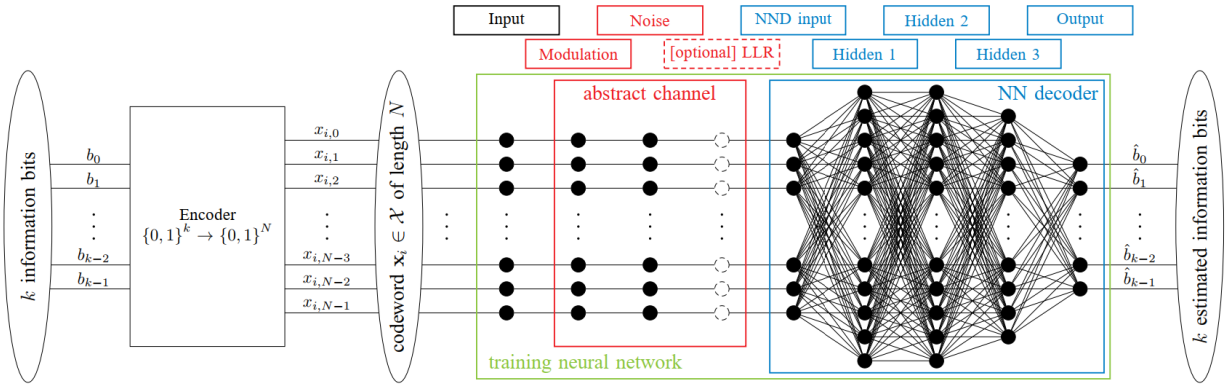

图 1. 极化码编码和神经网络译码结构图

3.2.1 极化码编码

图 1中的 Encoder 结构为极化码的编码结构，将 k bits 向量进行线性变换得到 N bits 向量，极化码的编码分为三步：

(1) 子信道的可靠性估计

常采用巴氏参数构造，通过下式计算子信道 $W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|u_i)$ 的巴氏参数：

$$Z(W_{2N}^{(2i-1)}) \leq 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2 \tag{3}$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2 \tag{4}$$

巴氏参数 $Z(W_N^{(i)})$ 越小越好，在 N 个极化子信道中选择 k 个作为信息位 A，其余位置作为冻结位，用于发送冻结比特，冻结比特一般设为全 0。

(2) 比特混合

将待发的 k bits 信息比特放在可靠性较高的信道发送，信道索引为 A，其余冻结比特统一设为全 0，得到 N bits 向量。

(3) 构造生成矩阵

记生成矩阵为 G_N，其计算公式为：

$$G_N = B_N F^{\otimes n} \tag{5}$$

其中，$N=2^n$，B_N 为比特逆序置换矩阵，$F=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$。

3.2.2 调制与噪声

实验中使用的 BPSK 调制和高斯白噪声信道（Additive White Gaussion Noise，AWGN），BPSK 调制即为映射 $s=1-2x$,AWGN 信道即信道噪声服从 $N(0,\sigma^2)$。图 1中的 LLR 层可选，使用 LLR 层的软译码通常性能会更好。

3.2.3 神经网络译码

采用全连接神经网络，网络的参数描述如下：

- 输入为 N 维向量，输出为 k 维向量，即神经网络译码结果
- 含有三层隐藏层，采用 128-64-32 的网络结构
- 隐藏层的激活函数采用 Relu 激活，输出层采用 Sigmoid 激活
- 优化器采用 Adam 优化器
- 损失函数采用 MSE
- 度量使用误比特率（Bit Error Rate，BER）

该网络与通常的网络不同，该网络不会过拟合，因为我们将噪声层 Noise 放入了网络结构中，但其不存在权重参数，而是直接的映射，因为噪声的随机性，网络每次训练的输入是不同的。

为了检验网络的泛化能力，通常不把全部数据都放进网络训练，将所有的 2^k 个比特向量划分为训练集和验证集，将验证集用于检验网络的泛化能力，由于极化码的一大特点就是其构造与信道紧密相关，为了更好地应用到实际应用中，往往需要对构造方法和信道条件解耦。

为了确定网络训练的噪声条件，论文提出归一化验证误差（Normalized Validation Error, NVE）：

$$NVE\left(\rho_{\mathrm{t}}\right)=\frac{1}{S}\sum_{s=1}^{S}\frac{BER_{NND}\left(\rho_{\mathrm{t}},\rho_{v,s}\right)}{BER_{MAP}\left(\rho_{v,s}\right)} \tag{6}$$

其中，ρ_t 为训练时的信道信噪比，$\rho_{v,s}$ 为第 s 个验证集的信道信噪比，$BER_{NND}\left(\rho_{\mathrm{t}},\rho_{v,s}\right)$ 为在 ρ_t 训练 $\rho_{v,s}$ 的 BER，$BER_{MAP}\left(\rho_{v,s}\right)$ 为在 $\rho_{v,s}$ 下的 MAP 译码

4 复现细节

4.1 与已有开源代码对比

论文 [1] 中仅对文章中部分实验代码开源，能够运行出极化码神经网络译码的结果，因此复现主要基于开源的部分代码的基础上。本次复现实验更多地是找出论文中的不足之处，对其加以改进。

4.2 实验环境搭建

python3.7，Anaconda22.9.0，TensorFlow 框架

4.3 界面分析与使用说明

代码文件包含两个文件夹，分别为 Matlab 代码文件夹和 Python 代码文件夹，Matlab 文件夹（图 2）中给出了传统的极化码译码算法的仿真代码，例如 SC、SCL 译码等，Python 代码文件夹（图 3）中给出了复现代码 ipynb 文件和改进后的 ipynb 文件（文件名含 improved）。在安装了所需的库文件后均可直接运行。

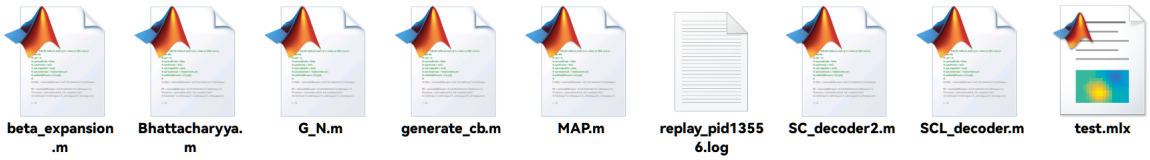

图 2. Matlab 代码文件夹

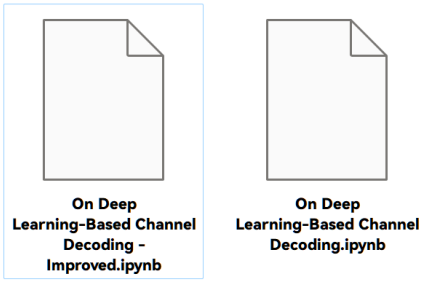

图 3. Python 代码文件夹

4.4 创新点

(1) 改进一：对构造方法的改变

原论文中采用的极化码构造方法为巴氏参数构造法，是依赖于信道条件的构造方法，而文中通过 NVE 选择训练的信噪比 SNR 实际上包含有 [18] 中 the best design-snr 的思想，其目的依然是对构造方法和信道条件的解耦。让两者彻底解耦的构造可采用 [19] 中华为所提出的 β-expansion 构造，其构造方式与信道无关。

在 β-expansion 构造中，每个极化子信道的可靠性由极化权重确定：

$$PW(W_N^{(i)}) = \sum_{i=0}^{n-1} b_i \beta^i \tag{7}$$

其中，$(b_{n-1}, ..., b_1, b_0)$ 为 i 的二进制展开，$\beta = 2^{1/4}$。

(2) 改进二：对训练目标的改进

原论文中在 NND 的训练中目标/标签采用的是信息位的真实值，而事实上任何确定的译码方式都会出错，错误概率最小的译码为 MAP 译码，MAP 译码是对噪声规律最好的解释。因此，应该将 MAP 译码结果作为目标/标签，网络要学习信道噪声的规律。

5 实验结果分析

先展示实验复现的运行结果，再展示第四节中介绍的两种改进后的结果。

5.1 实验复现结果

在码长 $N = 16$，信息长度 $k = 8$ 的条件下进行实验，NND 的网络结构采用 128-64-32 NN，实验结果如图 4所示。由实验结果可知，NND 译码是能够接近 MAP 译码的性能。

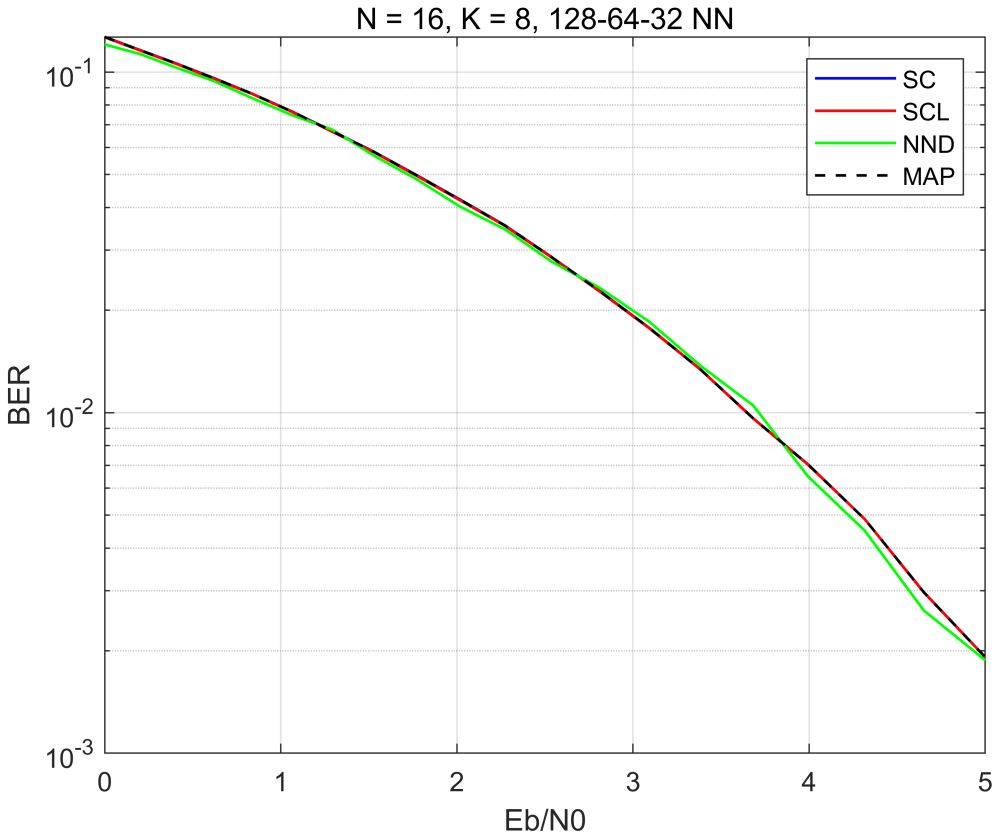

图 4. SC,SCL,NND,MAP 性能对比

5.2 改进一的实验结果

实验参数同 5.1 所示，但训练的信道信噪比选择不再采用原论文中的 NVE 计算，而是直接将构造方法改为使用 β-expansion 构造。实验结果如图 5所示。由实验结果可知，改进后的 NND 译码性能优于原论文的 NND 译码性能，在某些信噪比区间，甚至优于 MAP 译码性能。

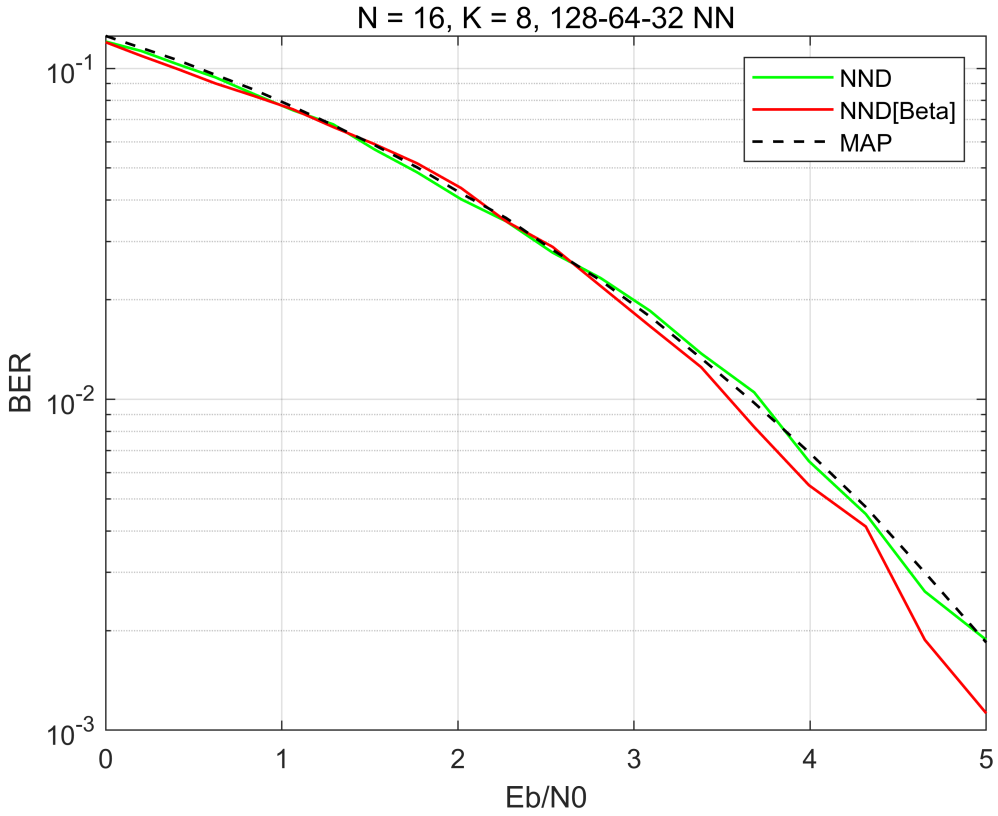

图 5. NND（原论文）,NND（Beta 改进）,MAP 性能对比

5.3 改进二的实验结果

实验参数同 5.1 所示，但训练的目标值由真实值换为 MAP 译码值。实验结果如图 6所示。由实验结果可知，改进后的 NND 译码性能优于原论文的 NND 译码性能，并基本优于 MAP 译码性能。

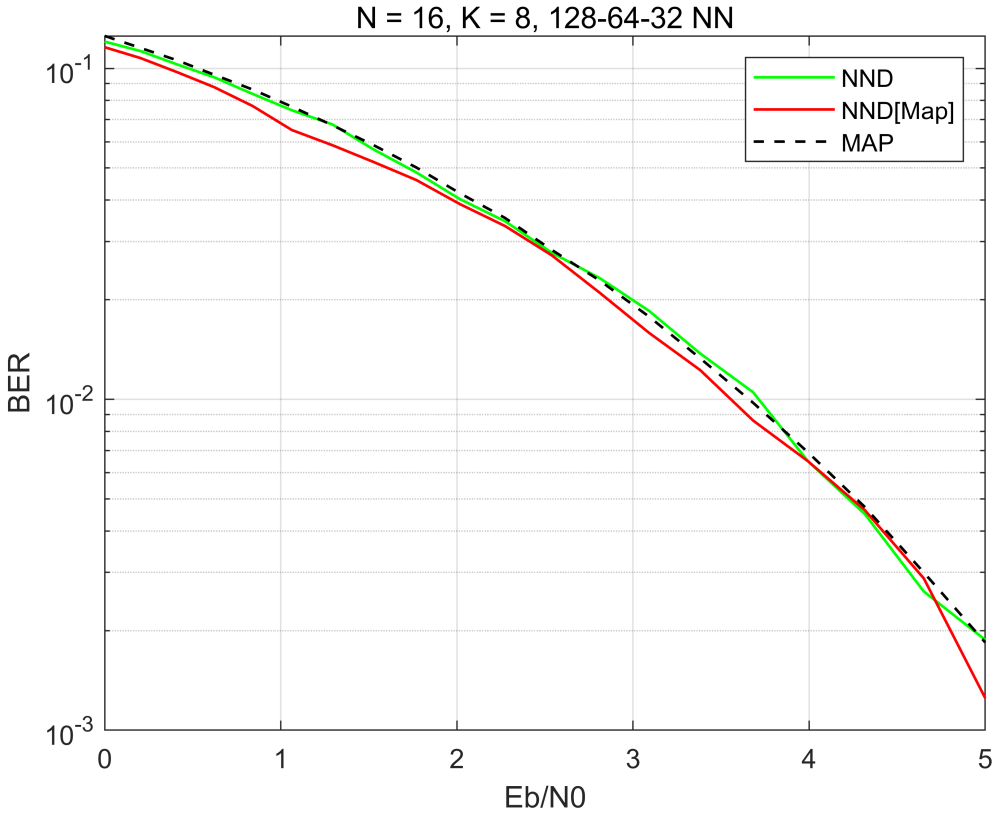

图 6. NND（原论文）,NND（Map 改进）,MAP 性能对比

5.4 实验结果汇总

实验参数同 5.1 所示，多种不同译码算法的实验结果如图 7所示。可以给出结论：NND 及其各种改进都能够达到 MAP 甚至优于 MAP。

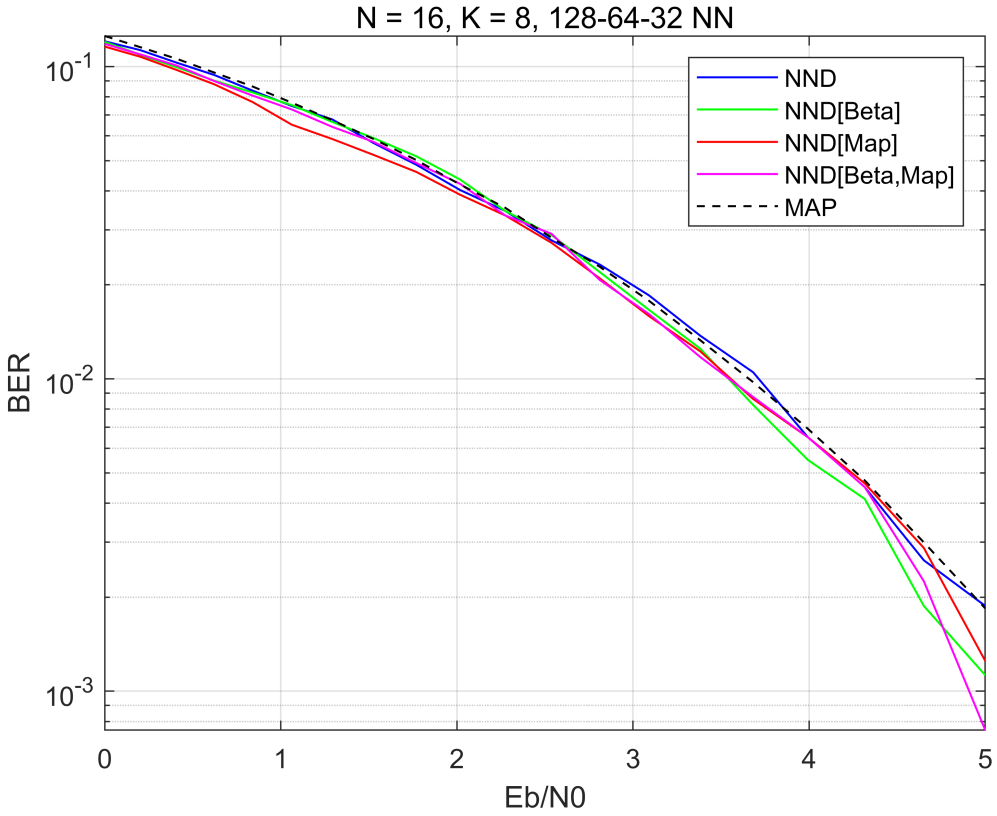

图 7. NND,NND(Beta),NND(Map),NND(Beta,Map),MAP 性能对比

6 总结与展望

本次实验复现了论文 [1] 中的仿真结果，并提出了两种对神经网络训练的改进，一是对构造方法的改进，二是对训练目标选择的改进。对论文的复现仿真可知 NND 译码是能够接近 MAP 译码的性能。两种不同的改进均能有效提高译码性能，甚至可优于 MAP 译码的性能。

NND 存在着一个致命的缺点：当信息位长度 k 较大时，k 维码字空间中含有 2^k 个码字，如果将全部码字拿来划分训练集和验证集，即使不考虑相同码字训练轮次 epochs，都会具有致命的复杂度问题。在 $N = 16$ 时，epochs 得达到 2^{16} 才能接近 MAP 译码性能。若只对部分码字划分，那对于没被划分到的码字的译码结果必然很差，网络的泛化能力较为一般。因此，对于解决非短码的情形有两个未来可研究的方向：一是借助极化码本身的递归结构，由短码 NND 译码递归到更长的 NND 译码；二是采用更有效适合的网络结构和神经元。

参考文献

[1] Tobias Gruber, Sebastian Cammerer, Jakob Hoydis, and Stephan Ten Brink. On deep learning-based channel decoding. In *2017 51st annual conference on information sciences and systems (CISS)*, pages 1–6. IEEE, 2017.

[2] Erdal Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on information Theory*, 55(7):3051–3073, 2009.

[3] Amin Alamdar-Yazdi and Frank R Kschischang. A simplified successive-cancellation decoder for polar codes. *IEEE communications letters*, 15(12):1378–1380, 2011.

[4] Ido Tal and Alexander Vardy. List decoding of polar codes. *IEEE transactions on information theory*, 61(5):2213–2226, 2015.

[5] Kai Niu and Kai Chen. Stack decoding of polar codes. *Electronics letters*, 48(12):695–697, 2012.

[6] Kai Chen, Kai Niu, and Jiaru Lin. Improved successive cancellation decoding of polar codes. *IEEE Transactions on Communications*, 61(8):3100–3107, 2013.

[7] Di Guan, Kai Niu, Chao Dong, and Ping Zhang. Successive cancellation priority decoding of polar codes. *IEEE Access*, 7:9575–9585, 2019.

[8] Bin Li, Hui Shen, and David Tse. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check. *IEEE communications letters*, 16(12):2044–2047, 2012.

[9] William R Caid and Robert W Means. Neural network error correcting decoders for block and convolutional codes. In *[Proceedings] GLOBECOM'90: IEEE Global Telecommunications Conference and Exhibition*, pages 1028–1031. IEEE, 1990.

[10] A Di Stefano, O Mirabella, G Di Cataldo, and G Palumbo. On the use of neural networks for hamming coding. In *1991., IEEE International Sympoisum on Circuits and Systems*, pages 1601–1604. IEEE, 1991.

[11] LG Tallini and P Cull. Neural nets for decoding error-correcting codes. In *IEEE Technical applications conference and workshops. Northcon/95. Conference record*, page 89. IEEE, 1995.

[12] Xiao-An Wang and Stephen B Wicker. An artificial neural net viterbi decoder. *IEEE Transactions on communications*, 44(2):165–171, 1996.

[13] Ari Hamalainen and Jukka Henriksson. A recurrent neural decoder for convolutional codes. In *1999 IEEE International Conference on Communications (Cat. No. 99CH36311)*, volume 2, pages 1305–1309. IEEE, 1999.

[14] Hossam Abdelbaki, Erol Gelenbe, and Said E El-Khamy. Random neural network decoder for error correcting codes. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 5, pages 3241–3245. IEEE, 1999.

[15] Ja-Ling Wu, Yuen-Hsien Tseng, and Yuh-Ming Huang. Neural network decoders for linear block codes. *International Journal of Computational Engineering Science*, 3(03):235–255, 2002.

[16] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[17] Eliya Nachmani, Yair Be'ery, and David Burshtein. Learning to decode linear codes using deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 341–346. IEEE, 2016.

[18] Harish Vangala, Emanuele Viterbo, and Yi Hong. A comparative study of polar code constructions for the awgn channel. *arXiv preprint arXiv:1501.02473*, 2015.

[19] Gaoning He, Jean-Claude Belfiore, Ingmar Land, Ganghua Yang, Xiaocheng Liu, Ying Chen, Rong Li, Jun Wang, Yiqun Ge, Ran Zhang, et al. Beta-expansion: A theoretical framework for fast and recursive construction of polar codes. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.