

基于深度学习的高阶调制下的极化码译码器

余俊江

摘要

本文实现了一种基于 BP 神经网络的极化码译码器，并对比研究了不同调制方式以及发送功率下的译码性能。

关键词：深度学习；极化码；深度神经网络译码器

1 引言

极化码^[1]是目前唯一可理论证明达到香农极限，并且具有可实用的线性复杂度编译码能力的信道编码技术。目前，极化译码面临着最优译码算法 MAP(Maximum A Posteriori, 最大后验概率) 译码的复杂度高、缺乏可能的并行化算法以及严重的译码延迟等问题。而深度学习具有强大的计算优势，神经网络经过训练后是静态的，数据只需要一次性通过网络，因此将深度学习应用于极化码的译码过程，可以有效降低译码时延，提高译码效率。

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^[1]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^[2]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^[3]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^[4]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^[5]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^[6]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^[7]，构成 CRC 辅助的译码算法。

近年来，随着诸如 GPU (Graphic Processing Unit, 图形处理单元) 等功能强大的新型硬件可用于加速学习和推理，关于神经网络译码器的研究逐渐增多。Cammerer 等^[8]提出了使用神经网络来代替传统 BP (Belief Propagation, 置信度传播) 译码算法的子译码器，改进了极化码 BP 迭代译码算法，在降低算法复杂度的同时，译码性能也有所提升。Nachmani 等^[9]提出了使用循环神经网络进行线性码译码的译码算法，在减少参数的前提下得出了与全连接神经网络相近的误码率性能。Seo 等^[10]研究了全连接神经网络在不同隐藏层数、每个层的节点数和激活函数条件下的译码性能。Gruber 等^[11]发现极化

码比随机码更容易被学习，将极化码和随机码分别用全连接神经网络进行译码，给出了极化码短码和随机码短码之间的译码差别。

2.1 传统极化码译码

极化码译码算法主要是 SC 译码算法：当码长无限长时，极化码的 SC 译码性能接近香农极限；而对于中短码长极化码，信道极化不完全会导致 SC 译码性能不足。极化码译码器的作用是根据译码器接收序列，通过一系列迭代运算得到信息序列的估计值。SC 译码器通过已知序列 (y_1^N, u_{Ac}) 产生对 u_1^N 的估计值 \hat{u}_1^N 。SC 译码器包含 N 个译码单元，每个译码单元对应一个信息源，SC 译码是按位译码。其中冻结位对于译码端是已知的，对于信息位，译码器需要进行似然比的迭代运算，并且需要借助先前译码估 \hat{u}_1^{i-1} ，对于第 i 位进行译码判决，可以通过式 (1) 计算似然比：

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

译码判决式为：

$$\hat{u}_i = \begin{cases} 0 & L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1 & \text{其他} \end{cases} \quad (2)$$

SC 译码算法在每次译码判决时会直接给出当前位的估计值，这种方式仅能保证局部最优化，对于整个译码过程来说，如果能保证全局最优化，那么就可以改善 SC 译码的缺陷。

2.2 基于深度神经网络的极化码译码

DNN (Deep Neural Networks, 深度神经网络) 的译码步骤如下，在信号发射端，信息序列为 u_1^k ，长度为 k ，然后，将 u_1^k 编码为码率为 k/N 的极化码，其中 N 为码长， k 为信息位长度，编码后的码字为 x_1^N ，通过调制生成适合在信道传输的符号 S ，假设信道为加性高斯白噪声信道。在信号接收端，接收符号 Y 可以表示为： $Y=S+n$ (n 为噪声)。噪声服从高斯分布，传输符号 Y 通过卷积神经网络得到译码符号 \hat{u}_1^k 。损失函数是实际神经网络输出与期望输出之间的差值。在本文中，使用 Sigmoid 作为激活函数，使用均方误差作为损失函数，其定义为：

$$L_{MSE} = \frac{1}{k} \sum_i (b_i - \hat{b}_i)^2 \quad (3)$$

通过神经网络反向传播算法（梯度下降）优化神经网络参数，达到最小化损失函数的目的，得到最佳的译码性能。

3 本文方法

3.1 本文方法概述

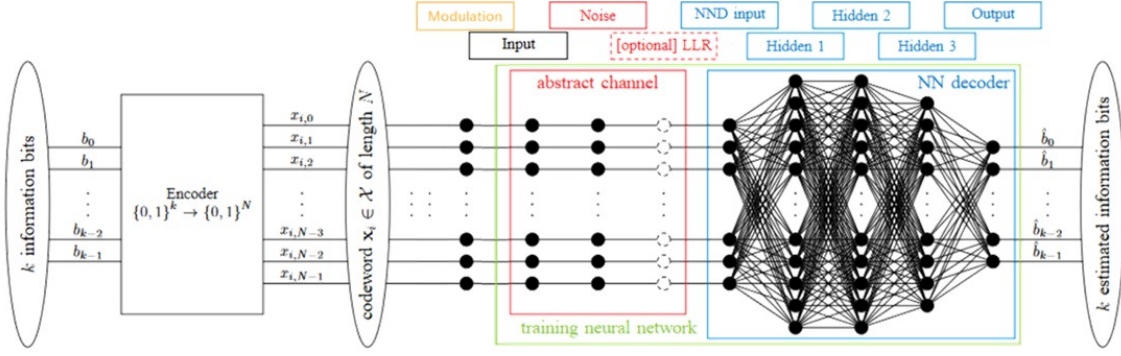


图 1: 方法示意图

本文的方法如图 1 所示，首先确定信息序列的长度 k 以及编码的码率 k/N ，生成所有可能的信息序列并进行编码。然后将生成的码字经过调制获得发送信号。再对发送信号添加高斯噪声模拟传输过程得到输出信号。最后将输出信号作为神经网络的输入，信息序列作为期望输出进行训练。

3.2 极化码构造与编码

本文采用等效巴氏参数构造极化码，其基本思路是将任意信道的巴氏参数等效为 BEC 信道下的巴氏参数，利用 BEC 信道下的迭代公式，得到极化信道的可靠性估计。对于 AWGN 信道，其原始信道的巴氏参数为

$$Z(W) = Z_{BEC}(W_1^{(1)}) = e^{-E_b/N_0} \quad (4)$$

其中 E_b/N_0 为信噪比，极化子信道的等效巴氏参数迭代公式为

$$\begin{cases} Z_{BEC}(W_{2j+1}^{(2i-1)}) = 2Z_{BEC}(W_{2j}^{(i)}) - Z_{BEC}^2(W_{2j}^{(i)}) \\ Z_{BEC}(W_{2j+1}^{(2i)}) = Z_{BEC}^2(W_{2j}^{(i)}) \end{cases} \quad (5)$$

将巴氏参数按照从小到大排序，得到最终的可靠性结果。

极化码的编码依赖于生成矩阵 G_N ，在本文中码长 $N = 2^n$ 的极化码的生成矩阵为

$$\mathbf{G}_N = \mathbf{F}^{\otimes n}, \quad \mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (6)$$

3.3 调制与噪声模块

调制就是把信号转换成适合在信道中传输的形式，本文中可选的调制方式有三种，分别是 BPSK、4ASK 和 4QAM。BPSK 是最简单的调制方式之一，如图 2.(a) 所示，两个星座点分别表示 0 和 1。BPSK 可以很自然地将码字对应到调制符号上。但是，使用 BPSK 一次发送只能传输一个比特的信息，传输效率较低。如果在 BPSK 星座图的基础上再增加两个点，就变成了 4ASK，如图 2.(b) 所示，四个点分别代表 00、01、11 和 10，这样一次发送就传输两个比特的信息。4ASK 的缺点是只利用了横轴的空间，使得发送功率相同时各星座点之间的距离较小，导致传输容易发生错误。如图 2.(c) 所示，4QAM 利用了纵轴的空间，不仅一次发送可以传输两个比特，且相比于 4ASK，传输中发生错误的概率也减小了。

本文模拟的信道为加性高斯白噪声 (AWGN) 信道，因此在调制后的信号加上一个均值为 0，标准

差为噪声功率的噪声即可。

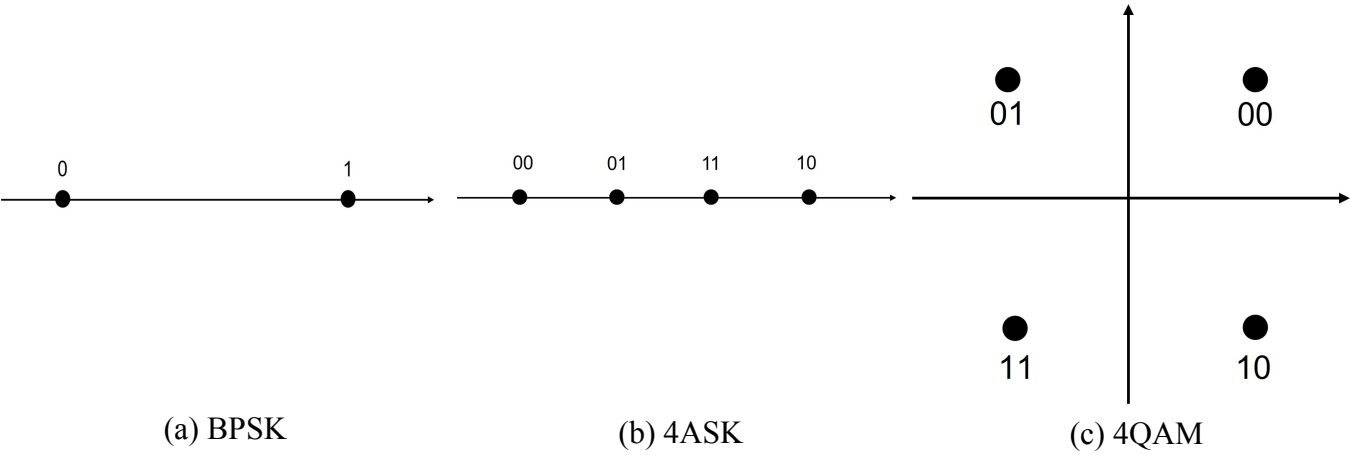


图 2: 三种调制方式的星座图

4 复现细节

4.1 与已有开源代码对比

本文的代码参考了文献 [10] 的源代码，源代码主要目的是探究神经网络译码的可行性，因此选择了 BPSK 作为调制方式，并且信号发送功率也为固定值。为了适应高阶调制的广泛应用，本文的代码可选择 BPSK/4ASK/4QAM 作为调制方式。此外，本文还将发射功率作为可调整的参数之一，可以清晰地观察到发射功率对译码效果的影响。

4.2 实验环境搭建

python3.7, TensorFlow 框架

4.3 界面分析与使用说明

```
k = 6 # 信息位
N = 32 # 码长
E = 1 # 发送功率
modulate = 'BPSK' # 调制方式('4QAM' or 'BPSK' or '4ASK')
train_SNR_Eb = 1 # 信噪比
nb_epoch = 2*16 # 训练轮数
design = [128, 64, 32] # 每个隐含层的节点数
batch_size = 256 # 训练样本的批数
optimizer = 'adam' # 优化器，调整每个节点权重的方法
loss = 'mse' # 损失函数
```

图 3: 操作界面示意

4.4 创新点

本文在原文的基础上，将调制方式以及发送功率作为可调参数，能够训练不同条件下的译码器。

5 实验结果分析

如图 4 - 5所示，不论对于 4ASK 还是 4QAM，提高发射功率可以降低神经网络译码器的误码率。此外，在 $E_b/N_0 < 3$ 时， $E(|X^2|) = 5$ 与 $E(|X^2|) = 10$ 的误比特率有较大的差距。 $E_b/N_0 > 3$ 时，二者

的误比特率没有明显的差异，因此需要根据不同信道条件选择合适的发射功率；如图 6所示，在相同的发射功率下，4QAM 的误比特率要优于 4ASK，因此调制方式的优劣也对神经网络译码器的性能有重要的影响；如图 7所示，高阶调制下的神经网络译码器对比低阶调制存在性能损失。从图 4 - 7中都可以看到，当 $E(|X^2|)$ 增加到 10 时，曲线变得平坦，可能原因是由于码长过小，在 E_b/N_0 很小时已经到达了误比特率的最小极限，增加 E_b/N_0 对误比特率没有太大影响。

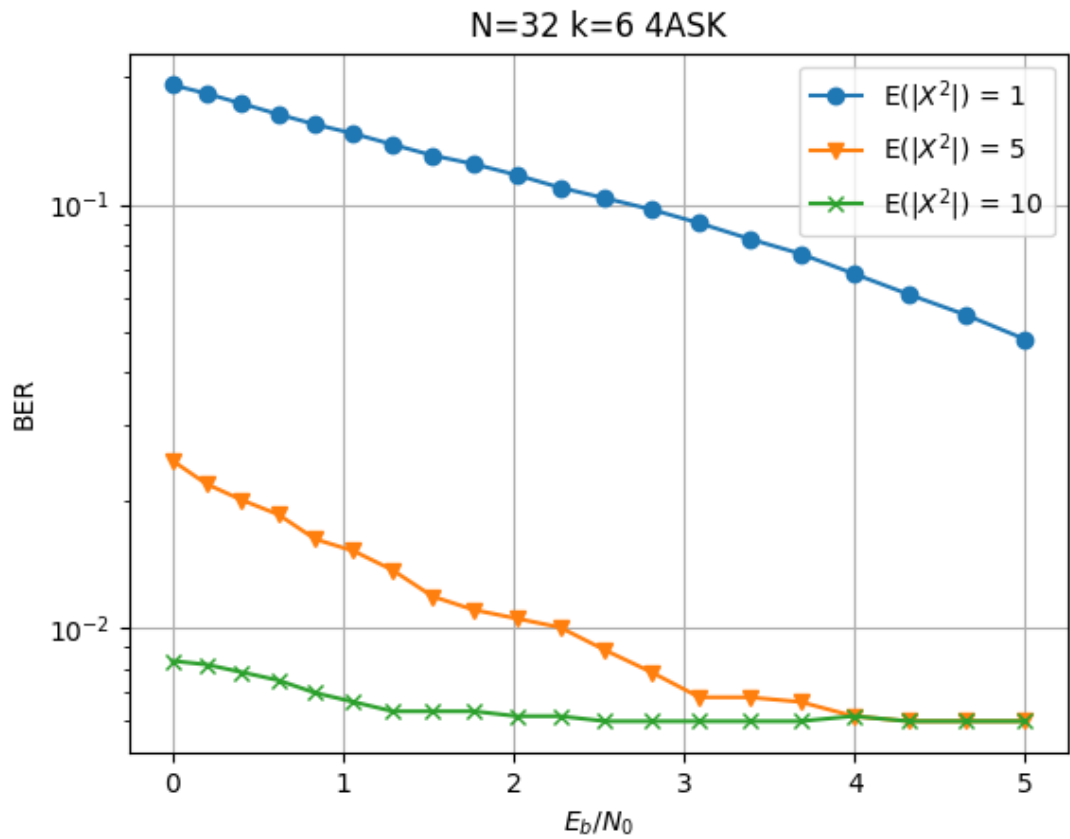


图 4: 4ASK 调制下不同发射功率的误比特率

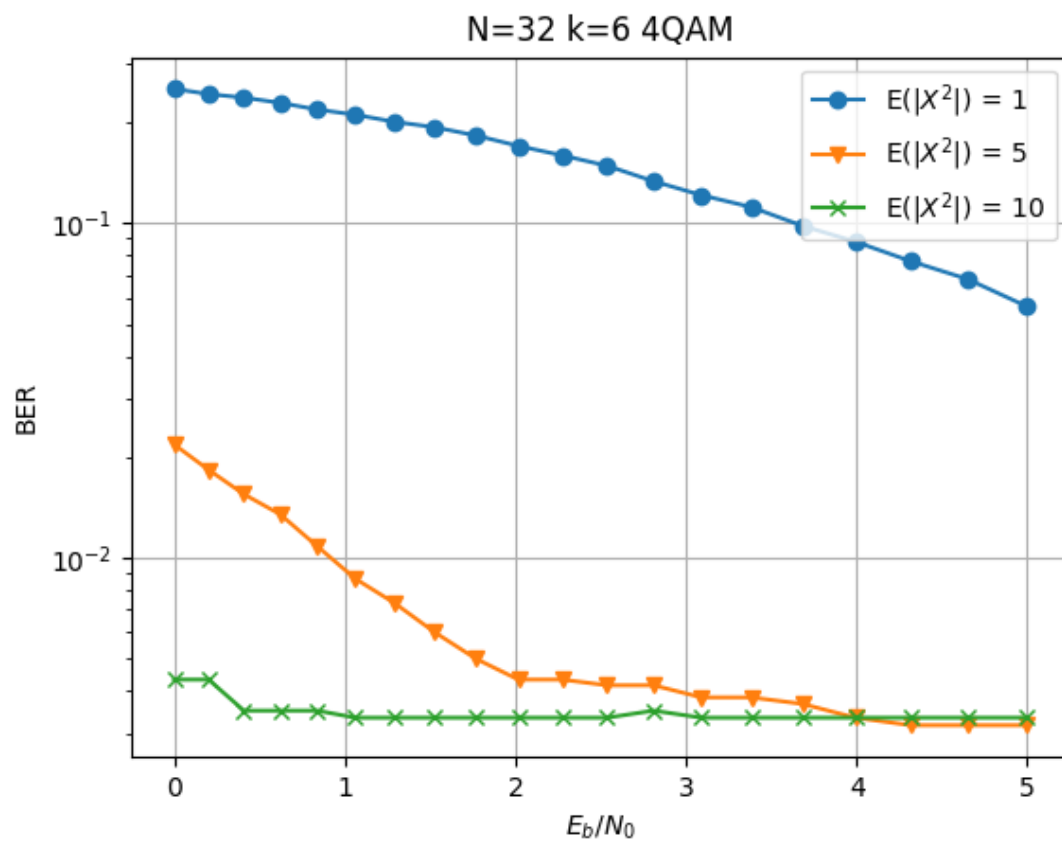


图 5: 4QAM 调制下不同发射功率的误比特率

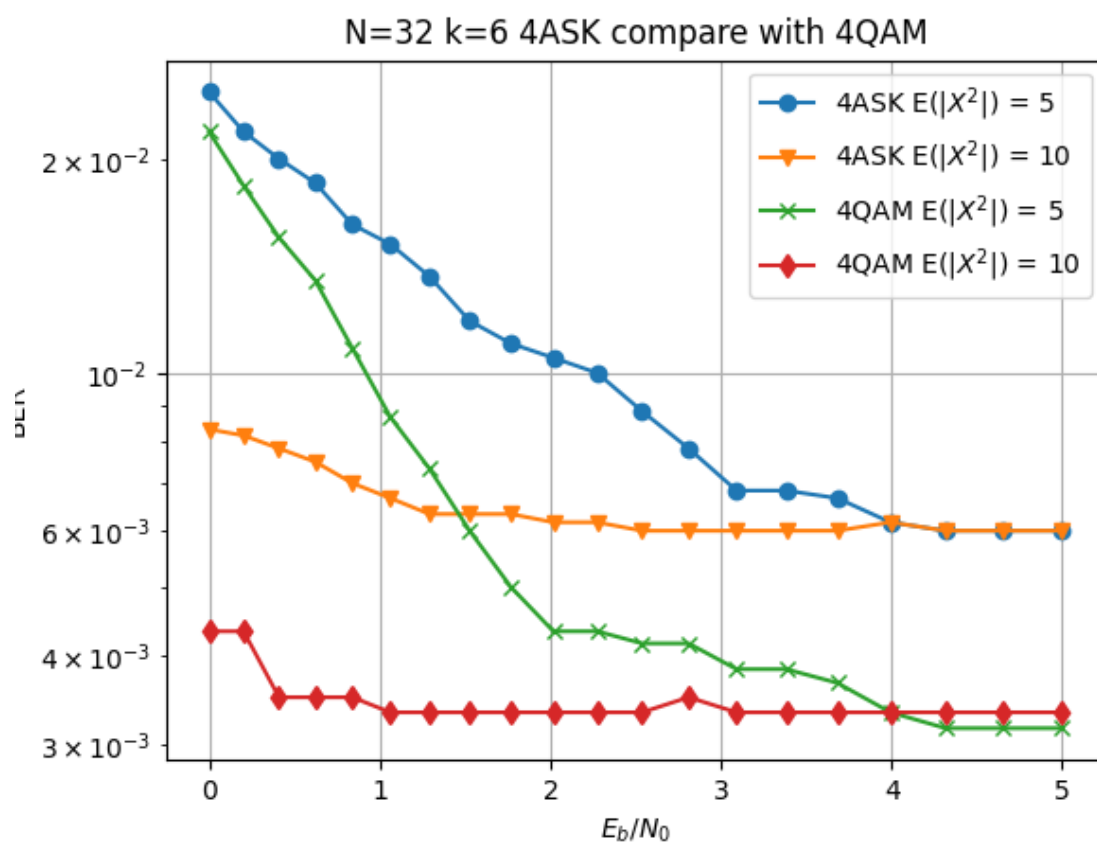


图 6: 4QAM 与 4ASK 误比特率对比

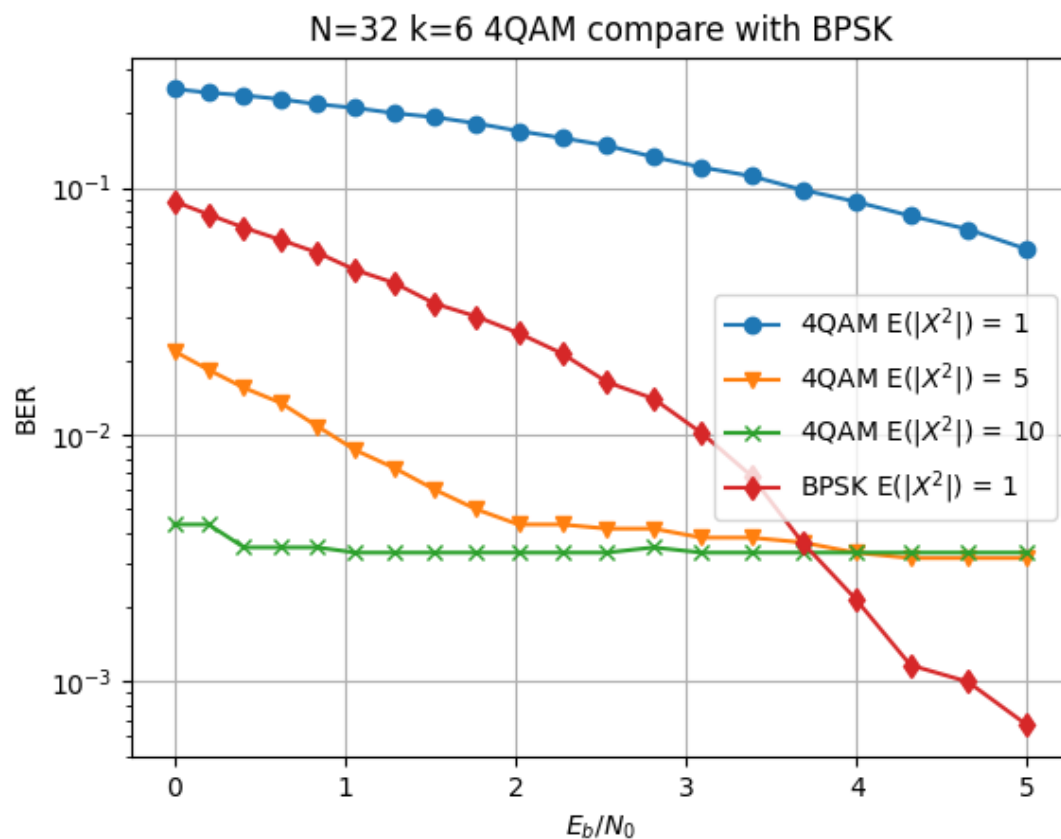


图 7: 4QAM 与 BPSK 误比特率对比

6 总结与展望

本文实现了应用于高阶调制的神经网络译码器，但是在性能上于低阶调制的神经网络译码器还有差距，并且编码的码率较低。今后可以在提高译码器性能，提高编码码率，使用更高阶的调制方式等方面进行改进。

参考文献

- [1] ARIKAN E. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels[J]. *IEEE Transactions on Information Theory*, 2009, 55(7): 3051-3073.
- [2] ALAMDAR-YAZDI A, KSCHISCHANG F R. A simplified successive-cancellation decoder for polar codes[J]. *IEEE communications letters*, 2011, 15(12): 1378-1380.
- [3] TAL I, VARDY A. List decoding of polar codes[J]. *IEEE Transactions on Information Theory*, 2015, 61(5): 2213-2226.
- [4] NIU K, CHEN K. Stack decoding of polar codes[J]. *Electronics letters*, 2012, 48(12): 695-697.
- [5] CHEN K, NIU K, LIN J. Improved successive cancellation decoding of polar codes[J]. *IEEE Transactions on Communications*, 2013, 61(8): 3100-3107.
- [6] GUAN D, NIU K, DONG C, et al. Successive cancellation priority decoding of polar codes[J]. *IEEE Access*, 2019, 7: 9575-9585.

- [7] LI B, SHEN H, TSE D. An adaptive successive cancellation list decoder for polar codes with cyclic redundancy check[J]. *IEEE communications letters*, 2012, 16(12): 2044-2047.
- [8] CAMMERER S, GRUBER T, HOYDIS J, et al. Scaling deep learning-based decoding of polar codes via partitioning[C] // *GLOBECOM 2017-2017 IEEE global communications conference*. 2017: 1-6.
- [9] NACHMANI E, MARCIANO E, BURSHTAIN D, et al. RNN decoding of linear block codes[J]. *arXiv preprint arXiv:1702.07560*, 2017.
- [10] SEO J, LEE J, KIM K. Decoding of polar code by using deep feed-forward neural networks[C] // *2018 international conference on computing, networking and communications (ICNC)*. 2018: 238-242.
- [11] GRUBER T, CAMMERER S, HOYDIS J, et al. On deep learning-based channel decoding[C] // *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. 2017: 1-6.