文章编号:1671-4598(2025)10-0280-09 DOI:10.16526/j.cnki.11-4762/tp.2025.10.036 中图分类号:TP391 文献标识码:A

基于自适应先验增强的智能 Turbo 译码算法

李若冰，刘丽哲，杨 朔，李 勇，王 斌
（中国电子科技集团公司 第五十四研究所，石家庄 050081）

摘要：针对传统 Turbo 信道译码算法误码率性能不足的问题，采用一种基于模型与数据双驱动的智能 Turbo 信道译码方法；基于传统 Max-Log-MAP 译码算法，将迭代过程深度展开，重构为多层级联的神经网络架构，提出自适应先验信息增强 APETurbo 译码网络模型，在模型中设计 APENet 神经子网络；该子网络采用可学习权重线性调整外部信息，并基于全连接层与非线性激活函数进一步提取非线性特征，利用可训练混合系数结合线性计算结果和非线性提取结果，构建残差连接结构，实现对先验信息的更精准估计；设计基于归一化的概率空间均方误差损失函数进行优化；仿真结果表明，在 AWGN 信道下，所提算法在误码率为 10^{-6} 时比 Max-Log-MAP 算法误码率性能提升约 0.4 dB。

关键词：Turbo 译码；Max-Log-MAP 算法；模型与数据双驱动；神经网络；可学习权重

Intelligent Turbo Decoding Algorithm Based on Adaptive Priori Enhancement

LI Ruobing, LIU Lizhe, YANG Shuo, LI Yong, WANG Bin
(The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China)

Abstract: Aiming at the insufficient bit error rate (BER) performance of traditional Turbo channel decoding algorithms, an intelligent Turbo channel decoding method based on the dual-driven model and data is presented; Based on the traditional Max-Log-MAP decoding algorithm, the iterative process is expanded in depth and reconfigured as a multilayered cascade neural network architecture, and an adaptive apriori information-enhanced APETurbo decoding network model is proposed, and an APENet neural sub-network is designed in the model. The sub-network adopts learnable weights to linearly adjust external information, further extracts nonlinear features based on the fully connected layer with a nonlinear activation function, constructs a residual connection structure by using trainable hybrid coefficients and combining linear computation and nonlinear extraction results, achieves a more accurate estimation of the a priori information, and designs a normalized probability space mean-square error loss function for optimization. Simulation results show that under the AWGN channel, and compared with the Max-Log-MAP algorithm, the proposed algorithm increases the BER by about 0.4 dB with a BER of 10^{-6}.

Keywords: turbo decoding; max-log-MAP algorithm; dual-driven model and data; neural network; learnable weights

0 引言

在第五代移动通信技术（5G，5th generation mobile communication technology）规模化商用及第六代移动通信（6G，6th generation mobile communication）标准预研的背景下，通信系统正面临传输速率、可靠性及复杂场景适应性的多重技术挑战。深空通信、卫星通信、军事通信等高端领域对高效、可靠信息传输的需求日益增长。信道编码作为通信系统的关键技术之一，关系到系统的传输性能和可靠性。相比于其他编码方式，Turbo 码[1]具有出色的误码纠正能力，译码性能逼近香农极限，作为一种经典的信道编码方案在上述多种通信场景中表现出优秀的纠错性能。目前主流的 Turbo 译码算法主要有两类：一类是基于最大后验概率算法（MAP，maximum a posteriori）及其改进算法；另一类是软输出维特比算法（SOVA，soft-output viterbi algorithm）及其改进算法。

MAP 算法的计算复杂度随约束长度和码长呈指数

收稿日期:2025-04-14； **修回日期**:2025-05-21。
基金项目:先进通信网全国重点实验室基金(SXX24104X030,SCX23641X011)。
作者简介:李若冰(2001-),女,硕士研究生。
通讯作者:刘丽哲(1988-),女,硕士,研究员级高级工程师。
引用格式:李若冰,刘丽哲,杨 朔,等.基于自适应先验增强的智能 Turbo 译码算法[J].计算机测量与控制,2025,33(10):280-288.

级增长，难以满足高实时性场景需求。为降低复杂度，Max-Log-MAP 算法采用对数域近似运算，省略了对数修正项[2]，对信息比特的对数似然比（LLRs，log likelihood-ratios）估计不准确，并在后续迭代译码过程中积累误差，导致译码性能下降。文献［3－7］通过引入缩放权重来修正 LLRs 的计算，但现有方法在参数优化效率与场景适应性之间仍存在显著矛盾。此外，由于在实际场景中存在各种噪声干扰与非线性效应，输入译码器的初始信道软信息质量较差，且计算译码器输出的后验概率对数似然比时也会受到噪声的影响，在后续迭代译码过程中先验信息的误差及信道估计偏差会随迭代次数累积，导致后验概率估计不准确，译码性能下降，难以面对复杂多变的通信环境和信道条件。

近年来，深度学习（DL，deep learning）在计算机视觉和自然语言处理等方面得到了广泛的应用并逐渐渗透到各个研究领域。特别的，深度学习与通信技术的结合也引起了广泛关注。以深度神经网络（DNN，deep neural network）为核心的深度学习技术，凭借其强大的非线性建模与特征自主学习能力，在信道译码领域展现出巨大潜力，为突破传统算法的性能瓶颈提供了新的思路和技术路径。

深度学习在信道译码中的研究分为模型驱动和数据驱动两种方案。文献［8－13］证明了基于数据驱动的信道译码算法相比于传统译码方法带来了性能的提升。文献［14］中提出 NEURALBCJR 算法，通过设计和训练循环神经网络（RNN，recurrent neural network），在 AWGN 信道实现卷积码近最优译码，但需传统算法先验知识初始化神经网络。文献［15］中提出一种端到端的深度学习架构，无需依赖传统译码算法，采用非共享权重，可靠性和适应性更优。文献［16］创新性地以端到端方式训练编码器和译码器，实现了中等块长度和复杂信道条件下接近容量极限的性能表现。文献［17］提出一种基于深度学习的自适应信道信噪比 Turbo 自编码器，通过注意力机制动态调整信道编译码应对信道条件变化。虽然上述基于数据驱动的译码方案证明了深度学习在设计高准确度译码器中的优势，但这些方案存在计算复杂度高、依赖大量数据的问题。

因此，针对上述数据驱动方案存在的问题，文献［18］首先提出 TurboNet 架构，通过引入参数化的 Max-Log-MAP 译码单元，使其能够从训练数据中有效学习译码规则，提升译码可靠性。文献［19］对 TurboNet 架构剪枝，通过保留最关键的计算外部 LLRs 所需的权重，降低复杂度的同时提高纠错能力。在最近的一项工作中[20]，作者提出了一种比特间参数共享的译码方案 TinyTurbo，用仅 18 个可训练权重实现泛化能力强的 Turbo 译码器，大幅减少了参数数量并提升了模型泛化性。但现有模型驱动方案可导性较差、特征提取能力不足的问题，限制了译码性能的提升。

为解决上述传统 Turbo 信道译码算法和基于单一模型驱动或数据驱动的智能 Turbo 信道译码算法存在的问题，进一步提升译码算法的误码率性能，本文的主要贡献如下：

本文在 TinyTurbo 的基础上提出了一种模型驱动与数据双驱动的混合架构 APETurbo，在模型中本文设计了 APENet 模块，引入可学习权重线性调整外部信息，并设计全连接层提取非线性特征，通过残差连接实现对外部信息的更精准估计。此外，本文提出了一个新的损失函数来训练本文提出的译码模型，该函数是一个基于归一化的概率空间均方误差损失函数，以实现对译码精度的优化。

本文展示了在 AWGN 信道上 APETurbo 的误码率性能明显优于传统 Turbo 译码器和现有的基于深度学习的智能 Turbo 译码器，在保证模型可解释性并保持与现有通信标准兼容性的前提下，实现译码算法性能的突破。本文证明了 APETurbo 具有良好的泛化能力，在（40，132）LTE Turbo 码上训练的 APETurbo 在不同码长、码率的 Turbo 码上同样表现良好。

1　系统编译码模型

在发送端，一个二进制信息序列 $u=(u_1,u_2,\cdots,u_k)$ 由一个包含两个相同交织器和两个相同递归系统卷积编码器（RSC，recursive systematic convolutional code）的 Turbo 编码器进行编码，这两个 RSC 编码器分别表示为 E_1 和 E_2，如图 1 所示。该序列被交织成 $\tilde{u}=(\tilde{u}_1,\tilde{u}_2,\cdots,\tilde{u}_k)$，交织后输出索引 $\pi(i)$ 和输入索引 i 之间满足关系 $\pi(i)=(f_1 i+f_2 i^2)\bmod K$，根据 3GPP 规则设置交织长度 K 对应的参数 f_1 和 f_2。RSC 编码器的生成矩阵为 $\left[1,\frac{g_1(D)}{g_0(D)}\right]$，其中 $g_0(D)=1+D^2+D^3$ 为生成多项式，$g_1(D)=1+D+D^3$ 为反馈多项式。反馈通道将一个 K 比特信息的块 $u_k,k=0,1,\cdots,K-1$，传递到编码器的输出，这些比特被称为系统位序列 $x^s=(x_1^s,x_2^s,\cdots,x_K^s)=(u_1,u_2,\cdots,u_k)$。第一个 RSC 编码器 E_1 根据系统位序列 u 生成一个校验位序列 $x^{1p}=(x_1^{1p},x_2^{1p},\cdots,x_K^{1p})$，第二个 RSC 编码器 E_2 根据交织的系统位序列 $\tilde{u}$ 生成一个校验位序列 $x^{2p}=(x_1^{2p},x_2^{2p},\cdots,x_K^{2p})$。最终编码码字为 $x^{en}=(x_1^e,x_2^e,\cdots,x_K^e)$，其中 $x_i^e=(x_i^s,x_i^{1p},x_i^{2p})$。假设编码序列经过二进制相移键控（BPSK，binary phase shift keying）调制并通过 AWGN 信道传输。

在接收端，对接收到的序列解调，并由一个软输出检测器以 LLRs 的形式计算传输比特的可信度信息，得到信道软输出，即相应比特为二进制 0 或 1 的概率。信

道软输出解复用后得到系统信息 y^s 和两个校验信息 y^{1p}、y^{2p}。所得的 LLRs 信息 $y^s=(y_1^s,y_2^s,\cdots,y_K^s)$ 表示 x^s 的可靠性信息，即相应比特为 1 或 0 的概率。$y^{1p}=(y_1^{1p},y_2^{1p},\cdots,y_K^{1p})$ 和 $y^{2p}=(y_1^{2p},y_2^{2p},\cdots,y_K^{2p})$ 分别类似的定义为 x^{1p} 和 x^{2p} 的可靠性信息。将 $y^{de}=(y_1^d,y_2^d,\cdots,y_K^d)$ 输入 Turbo 译码器，其中 $y_i^d=(y_i^s,y_i^{1p},y_i^{2p})$。如图 1 所示，将 y_k^s、y_k^{1p} 以及先验信息一起送入软输入软输出译码器（SISO，soft input soft output）D_1 进行译码，初次迭代时输入 D_1 的先验信息初始化为 0。由 D_1 的译码结果计算外部信息，外部信息和信息序列都进行交织后与 y^{2p} 一起送入 SISOD_2 译码，由 D_2 的译码结果计算外部信息并解交织，获得下一次迭代时所需的先验信息。多次迭代后，对 D_2 的输出解交织和硬判决得到译码序列。

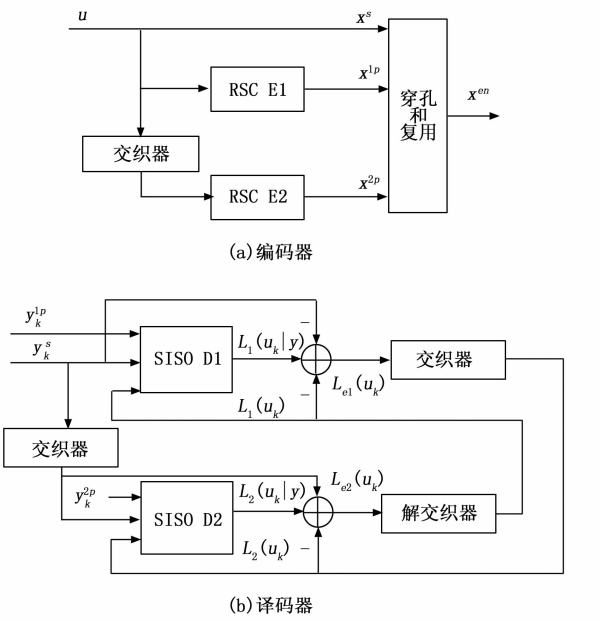

图 1 Turbo 编码器和译码器结构

2 基于 MAP 和 Max-Log-MAP 的信道译码算法

MAP 和 Max-Log-MAP 算法是两种传统经典的 Turbo 信道译码算法，其中 Max-Log-MAP 算法在 MAP 算法基础上更好地实现了译码性能和计算复杂度之间地平衡，现有模型驱动智能 Turbo 译码算法多基于此改进。下面简单回顾 MAP 和 Max-Log-MAP 译码算法的基本原理。

2.1 经典 MAP 译码算法

Turbo 码在译码时主要使用两个相同的交织器和 SISO 译码器、对交织器对应的解交织器和辅助硬判决电路，该译码结构的一个显著特征是进行迭代。所用的 SISO 译码器可以输入软信息进行处理，并输出软信息用于后续计算。两个具有相同结构的 SISO 译码器，分别表示为 D_1 和 D_2，均使用经典的 BCJR 算法[21]，以 D_1 译码器为例简单介绍。

设 $S_R=\{0,1,\cdots,2^m-1\}$ 是所有 2^mRSC 编码器状态的集合，m 表示记忆长度，$s\in S_R$ 是编码器在时间 k 的状态，$s'\in S_R$ 是编码器在时间 $k-1$ 的状态。信息序列长度为 K，第 k 时刻：RSC 编码器状态为 S_k，输出 $x=x^{en}=\{x_k\}$，其中 $x_k=x_k^e$。接收端，对接收序列解调，用 $y=\{y^s,y^{1p}\}$ 表示与编码器 E_1 对应编码比特的 LLR 信息，计算 D_1 输出的信息比特的后验概率 LLR：

$$L(u_k\mid y)=\ln\left(\frac{p(u_k=1\mid y)}{p(u_k=0\mid y)}\right)=\ln\left(\frac{p(u_k=1,y)}{p(u_k=0,y)}\right)=\ln\left(\frac{\sum_{(s',s)\to u_k=1}P(S_{k-1}=s',S_k=s,y)}{\sum_{(s',s)\to u_k=0}P(S_{k-1}=s',S_k=s,y)}\right)=\ln\left(\frac{\sum_{(s',s)\to u_k=1}P(S_{k-1}=s',S_k=s,y_{j<k},y_k,y_{j>k})}{\sum_{(s',s)\to u_k=0}P(S_{k-1}=s',S_k=s,y_{j<k},y_k,y_{j>k})}\right)\tag{1}$$

式中，$(s',s)\to u_k=1$ 表示当输入比特 $u_k=1$ 时，所有状态 $S_{k-1}=s'$ 到状态 $S_k=s$ 的分支转移，$(s',s)\to u_k=0$ 表示当输入比特 $u_k=0$ 时，所有状态 $S_{k-1}=s'$ 到状态 $S_k=s$ 的分支转移，各分支转移相互独立，所以转移概率可以由所有分支转移概率求和得到。

假设通过的 AWGN 信道是无记忆的，且 k 时刻后接收到的序列 $y_{j>k}$ 仅与编码器 k 时刻的状态 s 有关，而与 k 时刻之前的状态无关，则：

$$P(S_{k-1}=s',S_k=s,y)=P(s',s,y)=\beta_k(s)\gamma_k(s',s)\alpha_{k-1}(s')\tag{2}$$

其中：$\alpha_{k-1}(s')=P(s',y_{j<k})$ 和 $\beta_k(s)=P(y_{j>k}\mid s)$ 可以通过前向和后向递归计算，$\gamma_k(s',s)=P(y_k,s\mid s')$ 表示状态转移概率，代入 $L(u_k\mid y)$ 公式得：

$$L(u_k\mid y)=\ln\left(\frac{\sum_{(s',s)\to u_k=1}\beta_k(s)\gamma_k(s',s)\alpha_{k-1}(s')}{\sum_{(s',s)\to u_k=0}\beta_k(s)\gamma_k(s',s)\alpha_{k-1}(s')}\right)\tag{3}$$

$\alpha_k(s)$、$\beta_{k-1}(s')$、$\gamma_k(s',s)$ 的计算如下：

$$\alpha_k(s)=\sum_{s'}\gamma_k(s',s)\alpha_{k-1}(s')\tag{4}$$

$$\beta_{k-1}(s')=\sum_{s}\gamma_k(s',s)\beta_k(s)\tag{5}$$

$$\gamma_k(s',s)=C\exp\left(\frac{u_kL(u_k)}{2}\right)\exp\left(\frac{L_c}{2}\sum_1^n x_{kl}y_{kl}\right)\tag{6}$$

编码前后需要对寄存器清零，所以 $\alpha_k(s)$、$\beta_k(s)$ 的初始条件设置为 $\alpha_0(s=0)=1$，$\alpha_0(s\neq 0)=0$，$\beta_k(s=0)=1$，$\beta_k(s\neq 0)=0$。

为简化计算，考虑 $\bar{\alpha}_k(s)=\ln[\alpha_k(s)]$，$\bar{\beta}_k(s)=\ln[\beta_k(s)]$，$\bar{\gamma}_k(s',s)=\ln[\gamma_k(s',s)]$，并定义：

$$max^*(x,y)=\ln(e^x+e^y)=$$

$$max(x,y)+\ln(1+e^{-|x-y|}) \tag{7}$$

此时得到：

$$\bar{\alpha}_k(s) = max_{s'}^{*}[\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)] \tag{8}$$

$$\bar{\beta}_{k-1}(s') = max_{s}^{*}[\bar{\beta}_k(s)+\bar{\gamma}_k(s',s)] \tag{9}$$

$$\bar{\gamma}_k(s',s) = C'+\frac{u_k L(u_k)}{2}+\frac{L_c}{2}\sum_{1}^{n}x_{kl}y_{kl} \tag{10}$$

$$L(u_k \mid y) = max_{(s',s)\to u_k=1}^{*}[\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)+\bar{\beta}_k(s)]- max_{(s',s)\to u_k=0}^{*}[\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)+\bar{\beta}_k(s)] \tag{11}$$

计算外部信息 $L_e(u_k)$：

$$L_e(u_k) = L(u_k \mid y)-y_k^s-L(u_k) \tag{12}$$

后续将 $L_e(u_k)$ 经过交织并作为先验信息输入 SISO 译码器 D_2。迭代译码输出的信息比特 LLR 通过下式计算：

$$L^M(u_k \mid y) = L_e(u_k)+y_k^s+L(u_k) \tag{13}$$

2.2　Max-Log-MAP 译码算法

通过对数运算可以降低计算法复杂度，但仍需计算对数修正项 $\ln(1+e^{-|x-y|})$。由于该项最大值不会超过 ln2，所以采用近似计算 $max^*(x,y)\approx max(x,y)$。此时 $\bar{\alpha}_k(s)$、$\bar{\beta}_{k-1}(s')$ 计算如下：

$$\bar{\alpha}_k(s) = max_{s'}[\bar{\alpha}_{k-1}(s')+\bar{\gamma}_k(s',s)] \tag{14}$$

$$\bar{\beta}_{k-1}(s') = max_{s}[\bar{\beta}_k(s)+\bar{\gamma}_k(s',s)] \tag{15}$$

Max-Log-MAP 算法进一步简化了计算，但信道估计偏差和先验信息误差会在忽略对数修正项的近似处理中会被放大，导致性能显著下降。因此，本文考虑将 Max-Log-MAP 算法深度展开构建神经网络，同时结合模型数据双驱动深度学习思想提升译码性能。

3　基于自适应先验增强的智能 Turbo 译码算法设计

本文提出一种基于模型与数据双驱动的智能 Turbo 信道译码算法，通过深度展开重构传统译码架构，构建 Turbo 智能译码网络模型 APETurbo。该模型设计“模型驱动优化＋数据驱动优化”的混合架构，融合传统 Turbo 译码算法的迭代计算与深度学习的非线性特征提取能力。

本文所提出的 APETurbo 信道译码模型采用当前主流的智能信道译码算法流程，包括线下训练和线上测试两部分。线下训练并测试 APETurbo 译码模型的流程如图 2 所示。首先构建训练数据集，将搭建好的深度神经网络模型放在训练数据集上训练，验证模型性能并保存模型参数；线上测试时，APETurbo 译码网络模型加载线下训练阶段保存好的模型参数，替代经典 Turbo 译码器，接收端将接收到的信道软信息输入预训练深度译码网络模型，对模型性能进行测试。

深度学习通过多层神经网络进行数据表示学习，能够自动从大量数据中提取特征，实现复杂模式的识别和建模。相比于现有基于线性组合计算外部信息的模型驱动 Turbo 译码器，本文所提出的 APETurbo 译码器设计了一个非线性特征增强神经子网络模块 APENet 来捕捉复杂的信道特性，对迭代译码过程中交互的外部信息进行非线性特征提取，提升模型对复杂信道响应的建模能力，从而进一步提升了模型的译码准确度。

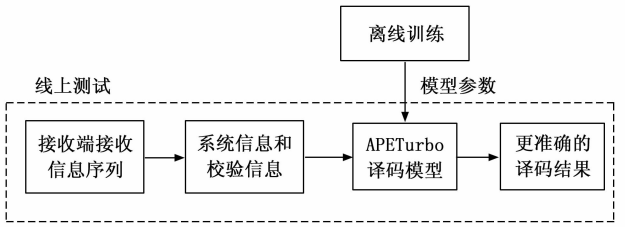

图 2　训练并测试 APETurbo 译码模型流程图

4　APETurbo 网络模型架构

本文基于模型与数据双驱动思想提出 APETurbo 译码网络模型，将 Max-Log-MAP 算法的迭代译码过程深度展开重构多层级联神经网络架构，每个迭代层级用一个智能译码单元表示，每个智能译码单元对应一次完整的外部信息交互过程。如图 3 所示，APETurbo 包括多个结构相同的智能译码单元以及一个辅助硬判决器，其中，$L_1^m(u_k)$ 表示第 m 次迭代计算中输入 SISO 译码器 D_1 的先验概率对数似然比信息，$L^M(u_k \mid y)$ 表示经过 M 次迭代计算得到每个信息比特的后验概率对数似然比信息。用 Sigmoid 函数对每个信息比特的后验概率对数似然比归一化处理，输出位于 [0,1] 区间的 o_k。对 o_k 硬判决得到最终的译码结果 $\overline{u}_k$。

Sigmoid 激活函数如下式：

$$S(x) = \frac{1}{1+e^{-x}} \tag{16}$$

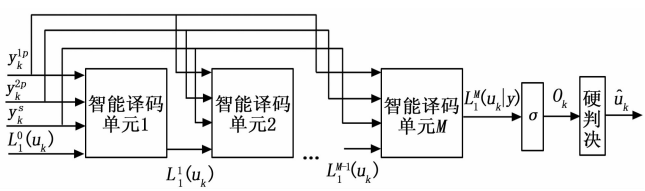

图 3　APETurbo 译码器架构图

智能译码单元结构如图 4 所示，每个智能译码单元包括两个相同的交织器、两个相同的 SISO 译码器、一个与交织器对应的解交织器以及两个结构相同而权重参数不同的 APENet 子网络。其中，两个 SISO 译码器及其对应的 APENet 神经子网络的模块分别对应正常层和交织层的外部信息计算，APENet 神经子网络模型可以提取外部信息的非线性特征，并进一步修正外部信息。

在每次迭代中首先由 SISO 译码器调用 Max-Log-MAP 算法计算后验概率对数似然比信息并输入 APENet 神经子网络。APENet 子网络通过可学习参数调

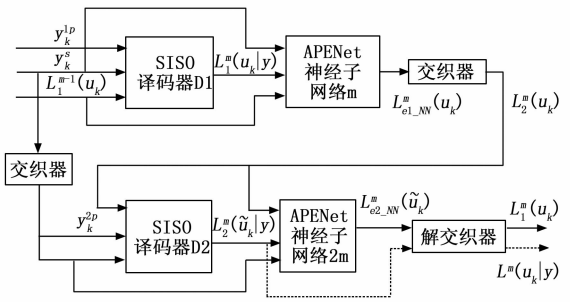

图 4 智能译码单元结构图

整线性计算得到外部信息，构建神经网络对外部信息进行进一步的非线性特征提取，并将线性组合计算的外部信息与经过神经网络增强的外部信息非线性特征提取结果通过残差连接结合，得到修正后的外部信息，最后进行交织/解交织操作得到译码器间交互的先验信息，实现两个译码器间的信息交互，多次迭代后对 APENet 神经子网络 2 的输出解交织和硬判决后得到最终的译码序列。其中，$L_1^m(u_k \mid y)$、$L_2^m(\tilde{u}_k \mid y)$ 分别表示第 m 个智能译码单元中 SISO 译码器 D_1、D_2 译码获得的信息序列、交织序列每个比特的后验概率 LLRs 信息，$L^m(u_k \mid y)$ 表示第 m 次迭代后输出的每个信息比特的后验概率 LLRs 信息，$L_{e\_NN}^m(u_k)$、$L^m(u_k)$ 表示第 m 个译码单元中经 APENet 子网络修正后的外部信息和先验信息。

本文所构建的 APENet 神经子网络模型结构如图 5 所示。该网络的核心功能是进一步精确估计两个 SISO 译码器间交互的外部信息。

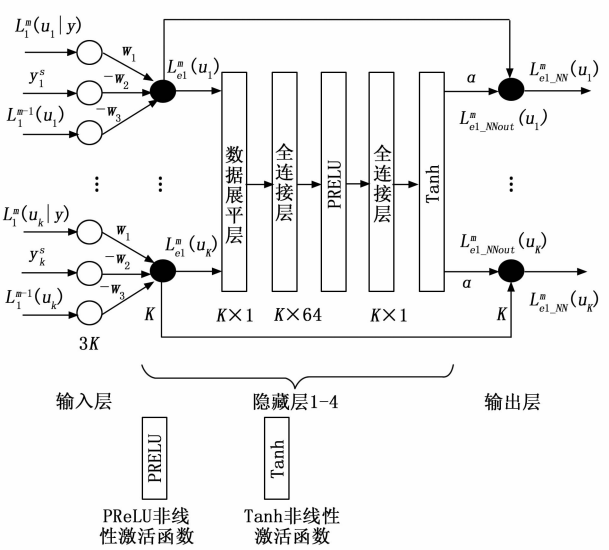

图 5 APENet 神经子网络模型原理

APENet 神经子网络模型由线性部分计算模块与非线性特征增强模块两部分构成。该模型的输入包括：SISO 译码器输出的每个比特的后验概率对数似然比、输入该 SISO 译码器的先验信息和系统信息，输出为修正后的外部信息。网络模型总层数为 6 层：第一层为输入层，中间 4 层为隐藏层，最后一层为输出层。

设 K 表示信息序列的长度，以 1 个样本输入为例，各层具体结构如下：

输入层由 $3K$ 个神经元组成，表示 K 个信息比特对应的后验概率对数似然比、系统信息及先验信息。

第一个隐藏层为线性部分计算层：由 K 个神经元组成，定义 3 个可学习权重 w_1、w_2、w_3，分别作用于 APENet 神经子网络输入的后验概率对数似然比项、系统信息项及先验信息项，输出为经可学习参数调整线性计算得到的外部信息，具体表达式如下：

$$L_e(u_k) = w_1 L(u_k \mid y) - w_2 y_k^s - w_3 L(u_k) \quad (17)$$

其中：$L_e(u_k)$ 为外部信息，$L(u_k \mid y)$ 为每个比特对应的后验概率对数似然比信息，y_k^s 为系统信息、$L(u_k)$ 为先验信息，输出数据形状为 $1 \times K$。

第二个隐藏层为数据展平层，如图 5 所示，第二个隐藏层输出的数据形状为 $K \times 1$。

第三和第四个隐藏层为非线性特征提取层：其中，第三个隐藏层采用一个输入为 1 维、输出为 64 维特征的全连接层，权重矩阵形状为 64×1，输出形状为 $K \times 64$，全连接操作后使用 PReLU 非线性激活函数；第四个隐藏层采用一个输入维度 64 维、输出 1 维的全连接层，权重矩阵形状为 1×64，输出的数据形状为 $K \times 1$，后接 Tanh 非线性激活函数，输出为经神经网络处理得到的外部信息非线性特征提取项。

输出层经过数据重塑恢复原始数据维度 $1 \times K$，构建残差连接结构，将经过可学习权重调整的线性计算结果与经过神经网络增强的外部信息非线性特征提取项混合。残差结构将线性部分计算层的输出作为基准路径，将经过非线性特征增强模块输出的非线性特征提取项作为残差路径，通过跳跃连接实现梯度稳定与特征增强。

在输出层的残差连接结构中，非线性特征提取项前设计了一个可学习混合系数 $\alpha \in [0,1]$，用于平衡线性计算的外部信息值与神经网络增强的外部信息非线性特征提取值，动态调整两者的融合比例，实现两路径的自适应混合，既保留了线性组合计算外部信息的可靠性，又通过神经网络学习信道中的非线性特征，对外部信息进行进一步修正，提升了译码模型的误码率性能。同时，也使模型保持稳定，避免了梯度消失问题。表达式如下：

$$L_{e\_NN}(u_k) = L_e(u_k) + \alpha \cdot L_{e\_NNout}(u_k) \quad (18)$$

其中：$L_e(u_k)$ 表示经过线性组合计算得到的外部信息，$L_{e\_NNout}(u_k)$ 表示用神经网络对 $L_e(u_k)$ 特征增强得到的非线性特征提取项，$L_{e\_NN}(u_k)$ 表示将线性计算部分与非线性特征提取部分按一定比例结合得到的修正后的

外部信息。如式（18）所示，APENet 神经子网络模型在保留传统译码算法可解释性的同时，提高了模型对信道非线性特征的提取能力。

5　训练策略

本文提出了一个端到端的均方误差（MSE，mean square error）损失函数来训练 APETurbo 网络模型的参数，优化译码精度，其数学构造如下：

$$L_{\text{total}}(u_k)=\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K}\{\sigma[L_{\text{pred}}^{(k)}(u^{(i)}\mid y^{(i)})]-\sigma\{L_{\text{true}}^{(k)}(u^{(i)}\mid y^{(i)})]\}^2 \tag{19}$$

其中：$\sigma(\cdot)$ 表示 Sigmoid 函数，将对数似然比映射至概率空间；假设训练批次大小为 B，即有 B 个长度为 K 的消息序列 $u^{(1)},u^{(2)},\cdots,u^{(B)},u^{(i)}\in\{0,1\}^K,i=\{1,2,\cdots,B\}$，$y^{(1)},y^{(2)},\cdots,y^{(B)}$ 为接收到的信息比特的 LLR 信息，$L_{\text{pred}}^{(k)}[u^{(i)}\mid y^{(i)}]$ 为每个比特的后验概率 LLR 预测值，即 APETurbo 译码器最后一次迭代输出的每个信息比特的后验概率 LLR，将一个批次信息序列的后验概率 LLR 预测值作为模型预测值；$L_{\text{true}}^{(k)}[u^{(i)}\mid y^{(i)}]$ 为 Log-MAP 算法最后一次迭代输出的每个信息比特的后验概率 LLR，将该批次数据经 Log-MAP 算法译码得到的后验概率 LLR 作为模型真实值。

该损失函数通过 Sigmoid 函数将 LLR 值映射到概率空间，缓解了 LLR 幅值无界导致的梯度爆炸问题。采用 MSE 损失对 LLR 值进行回归监督，计算每个批次所有数据的均方误差损失，求和并取批次均方误差的平均值，最小化预测值与真实值在概率空间的均方误差。相比 BCE 的二元分类监督，MSE 损失更契合 Turbo 译码中软信息传递的特性，使网络能更精确地学习后验概率分布。

6　计算复杂度分析

本节选取了 Max-Log-MAP、TinyTurbo、APETurbo 三种算法进行计算复杂度的比较，不同译码算法单次迭代正常层的计算复杂度如表 1 所示，设码长为 K。

表 1　译码器计算复杂度分析

	复杂度
Max-Log-MAP	$O(K)$
TinyTurbo	$O(K)$
APETurbo	$O(K)$

设码长 $K=40$，编码器状态数 $S=8$，神经网络维度 $N_m=64$。Max-Log-MAP 译码算法的总复杂度为 $(22S+2)K$FLOPs，其中分支度量计算复杂度为 14 SK-FLOPs，前/后向递推计算复杂度为 4 SKFLOPs。TinyTurbo 译码算法通过优化外部信息生成流程，引入 3 KFLOPs 的额外计算，总复杂度为 $(22S+5)K$FLOPs。本文所提出的 APETurbo 译码算法融合轻量级神经网络辅助外部信息修正，总复杂度为 $(22S+2N_m+7)K$FLOPs，相较 TinyTurbo 译码算法新增 $(2N_m+2)K$FLOPs，通过构建轻量级非线性模块，保持了线性计算复杂度 $O(K)$。

7　实验结果与分析

7.1　数据集构建

使用 LTE 标准下的（40，132）Turbo 码进行仿真实验以评估译码性能。在信噪比为 -0.5 dB 的 AWGN 信道下随机生成 60 000 组训练数据，构建训练数据集。在信噪比为 -0.5、0、0.5、1、1.5、2、2.5、3、3.5 dB 的 AWGN 信道上分别随机生成 5 000 000 组数据，构建测试数据集。

7.2　仿真参数设置

采用包含 3 个智能译码单元的 APETurbo 译码模型进行训练和测试，对应 3 次迭代过程。具体仿真参数设置如表 2 中所示。

表 2　APETurbo 仿真参数设置

算法	APETurbo
译码迭代次数	3
线性权重学习率	8×10^{-4}
神经网络学习率	8×10^{-5}
混合系数学习率	8×10^{-4}
优化器	Adam 优化器
损失函数	归一化 MSE 损失函数

训练时，本文实施分层参数优化，对基础线性权重、神经网络参数和残差混合系数设置差异化学习率，通过参数类型感知的梯度更新策略平衡收敛速度与过拟合风险，并配合梯度裁剪策略有效抑制梯度爆炸。

7.3　误码率性能分析

测试集采用 LTE 标准下的码长为 40，码率为 1/3 的（40，132）LTE Turbo 码进行仿真测试。测试集的样本数为 5 000 000，在信噪比为 $-0.5\sim3.5$ dB 的 AWGN 信道上进行测试。

将 APETurbo 译码器（记为“APETurbo”）的译码性能与标准 Max-Log-MAP 译码器（记为“MaxLog-MAP”）、MAP 译码器（记为“MAP”），以及基于模型驱动的 TinyTurbo 译码器（记为“TinyTurbo”）的译码性能进行比较，实验结果如图 6 所示。

在采用 3 次迭代译码的情况下，本文提出的 APE-Turbo 译码器在（40，132）LTE Turbo 码上的译码准确度明显优于 Max-Log-MAP 译码器，其误码率性能接近并超越 3 次迭代的 MAP 译码器，随着信噪比的提升误码率性能优势愈加明显。且 APETurbo 译码器的误码

率性能明显优于 TinyTurbo 译码器。特别的，在误码率为 10^{-6} 时，APETurbo 译码器的误码率性能比传统的 Max-Log-MAP 译码器提升约 0.4 dB，比 TinyTurbo 译码器提升约 0.2 dB。本文构建的 APETurbo 译码器复杂度提升较小，维持在线性计算复杂度范围，但误码率性能逼近甚至超越了 MAP 算法，验证了该架构在平衡译码性能与计算效率方面的优越性。

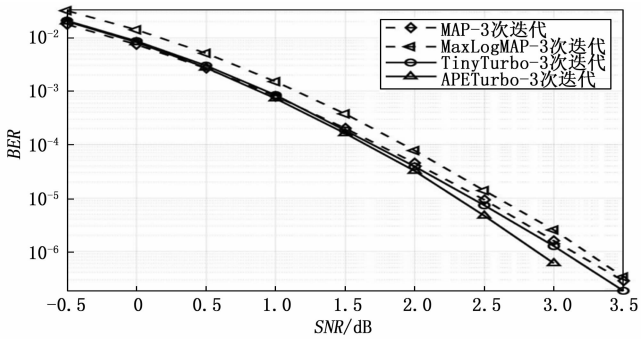

图 6 （40，132）LTE Turbo 码的误码率性能曲线

7.4 推广到不同块长度、码率和网络

较长码块下训练译码器模型需要大量的 GPU 内存，由于本文训练所采用的 GPU 内存不足，若使用大批量长码块的训练数据来训练，将导致训练不稳定[22-24]。因此，本文希望所设计的译码模型可以在短码上训练，并能够推广到长码块上使用。下面将验证 APETurbo 译码器在不同的块长度、码率和网格的条件下，同样具有误码率性能的优势

如图 7 所示，本文在（40，132）LTE Turbo 码上训练的 APETurbo 码在（200，612）LTE Turbo 码上测试时相较于传统译码算法表现出良好的性能，且误码率性能在测试的整个信噪比范围内均优于 TinyTurbo 译码器。特别的，在信噪比为 1.5 dB 时，误码率可以达到 10^{-8}，相比于 TinyTurbo 译码器，误码率性能提升约一个数量级，表明 APETurbo 可以推广到不同的码长的 Turbo 码译码。

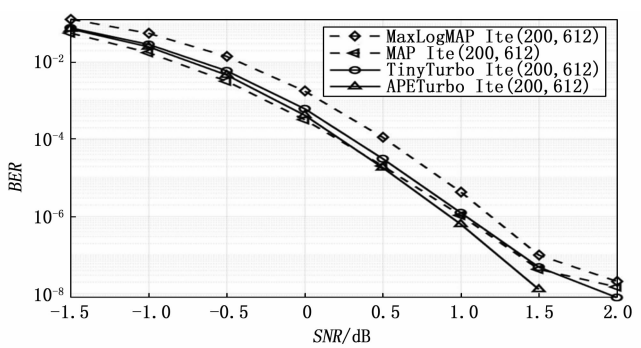

图 7 （200，612）LTE Turbo 码的误码率性能曲线

在码长为 200，码率为 $\frac{1}{2}$ 的（200，412）LTE Turbo 码上进行测试，测试结果如图 8 所示，APETurbo 译码器的误码率性能在测试的整个信噪比范围内均优于 Max-Log-MAP 译码器和 TinyTurbo 译码器，接近并超越了 MAP 译码器的误码率性能，证明了 APETurbo 译码器能推广到不同码率的 Turbo 码上。

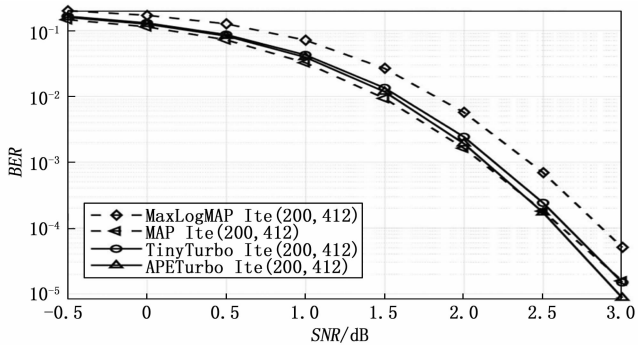

图 8 （200，412）LTE Turbo 码的误码率性能曲线

本文用在 LTE Turbo 上训练的 APETurbo 译码器来译码码长 40、码率 $\frac{1}{3}$ 的 Turbo-757 码，生成矩阵为 $\left[1,\frac{1+D^2}{1+D+D^2}\right]$，结果如图 9 所示。APETurbo 译码器的误码率性能显著优于 Max-Log-MAP 译码器，接近并逐渐超越 MAP 译码器，证明了 APETurbo 译码器能较好的推广到采用不同生成矩阵的 Turbo 码上。

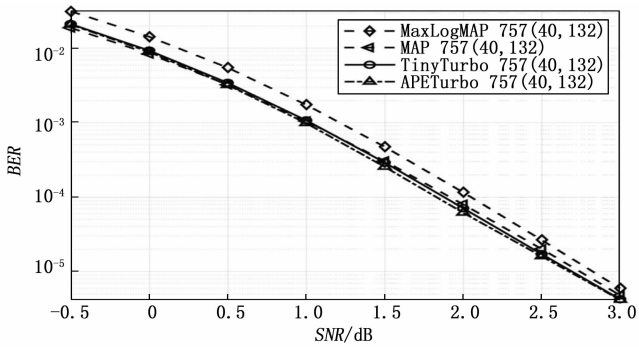

图 9 Turbo-757 的误码率性能曲线

7.5 非高斯白噪声信道下的误码率性能

本文评估了在 AWGN 信道上训练的 APETurbo 译码器，在非 AWGN 信道下译码的误码率性能，如图 10 所示。

本文选择突发噪声信道进行测试，定义 $y=x+z+w$，其中 $z\sim N(0,\sigma_1^2)$ 是 AWGN 噪声，$w\sim N(0,\sigma_b^2)$ 是具有高噪声功率和低发生概率的突发噪声 ρ，本文考虑 $\sigma_b=3$ 和 $\rho=0.01$。如图 10 所示，APETurbo 译码器在突发噪声信道下表现出较好的泛化能力。APETurbo 译码器的译码准确度明显优于传统译码器和 TinyTurbo 译码器，随着信噪比的增加误码率性能的优势愈加明显。

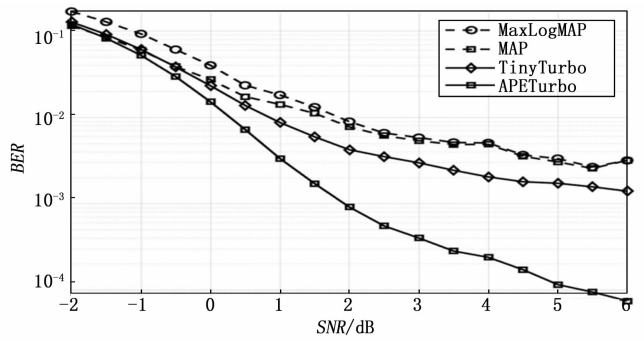

图 10　APETurbo 译码器在突发噪声信道下的鲁棒性

7.6　消融研究

本文所提出的 APETurbo 译码器的优越性源于以下创新：1）通过添加神经网络分支学习信道的非线性特征，并设计可学习参数，动态调整线性组合计算的外部信息与经神经网络非线性特征提取增强后的外部信息的占比，获取最终交互的外部信息；2）本文对 APETurbo 译码器输出的后验 LLRs 值和 Log-MAP 译码器输出的后验 LLRs 值进行归一化，将 LLRs 值映射到概率空间，并以 Log-MAP 算法的后验 LLRs 值作为训练目标，采用端到端的 MSE 损失进行训练。

本节通过以下消融实验说明这些组成部分对 APETurbo 译码器误码率性能提升的贡献，实验在 AWGN 信道下测试，采用（40，132）LTE Turbo 码，译码模型均采用 3 次迭代：

1）模型中加入含残差的神经子网络 APENet，但采用与 TinyTurbo 相同的训练过程。如图 11 所示，在测试的整个信噪比范围内均优于 Max-Log-MAP 译码器和 TinyTurbo 译码器，接近并超越了 MAP 译码器的误码率性能。特别是在信噪比为 3 dB 时，APETurbo 的误码率性能可以达到 10^{-6} 数量级。

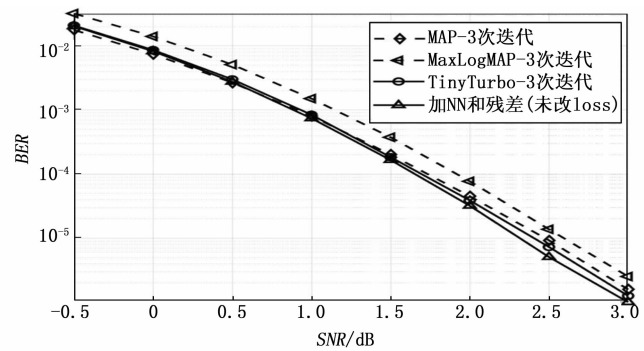

图 11　采用 APENet 的模型误码率性能

2）不添加神经网络分支和残差结构，但改用基于 Sigmoid 归一化的端到端 MSE 损失函数进行训练。如图 12 所示，用基于 Sigmoid 归一化的端到端 MSE 损失函数进行训练的模型译码准确度明显优于传统的 Max-Log-MAP 译码器，接近并超越 MAP 译码器的译码准确度。随着信噪比的提升误码率性能优势愈加明显，且在整个测试的信噪比范围内误码率明显优于 TinyTurbo 译码器。特别的，在信噪比为 3 dB 时，采用基于归一化的 MSE 损失函数进行训练的模型误码率性能可以达到 10^{-7} 数量级。

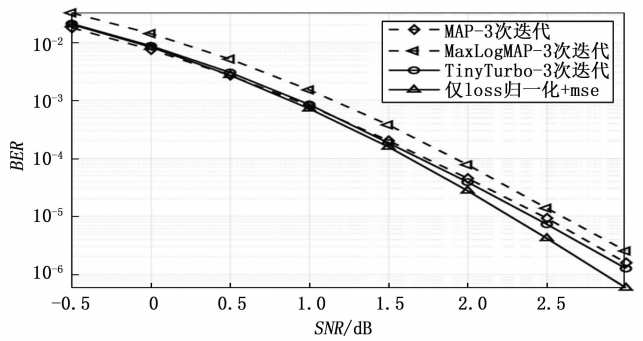

图 12　采用不同训练策略的 APETurbo 误码率性能对比图

3）如图 13 所示，模型中加入含残差的神经子网络 APENet，并改用基于归一化的 MSE 损失函数进行训练。模型译码准确度明显优于传统的 Max-Log-MAP 译码器，接近并超越 MAP 译码器。在整个测试的信噪比范围内误码率明显优于 TinyTurbo 译码器。在信噪比为 3 dB 时，相比于 1）中仅加入含残差的神经子网络的译码模型，采用基于归一化的 MSE 损失函数进行训练时，模型误码率性能提升约半个数量级。

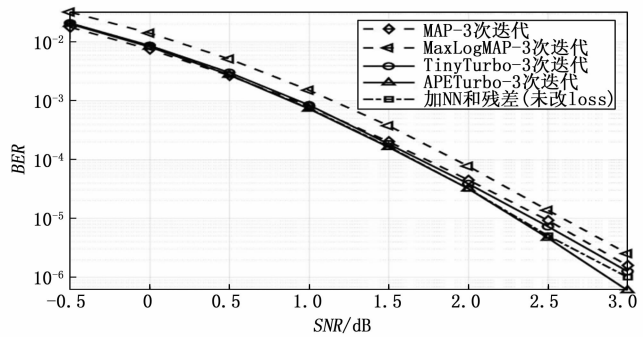

图 13　APETurbo 的误码率性能

8　结束语

本文提出了一种基于模型与数据双驱动的智能 Turbo 信道译码算法，构建了 APETurbo 智能信道译码网络模型，将 Max-Log-MAP 算法的迭代译码过程深度展开，重构为多层级联神经网络架构。在 APETurbo 译码网络模型中设计 APENet 神经子网络，利用可学习权重线性调整外部信息计算，并基于全连接层与非线性激活函数进一步提取非线性特征，通过残差连接结构结合线性计算结果和非线性特征提取结果，实现对先验信息的更精准估计，进一步增强译码模型的非线性特征的提取能力以及复杂信道的建模能力。此外，本文还提出了一个基于 Sigmoid 归一化的概率空间均方误差损失来改

善训练过程，优化译码精度。所提出的 APETurbo 译码模型误码率性能显著优于传统 Turbo 信道译码算法。

参考文献：

[1] BERROU C，GLAVIEUX A，THITIMAJSHIMA P. Near Shannon limit error-correcting coding and decoding：Turbo-codes [C] //Proceedings of ICC'93-IEEE International Conference on Communications. IEEE，1993，2：1064－1070.

[2] ERFANIAN J，PASUPATHY S，GULAK G. Reduced complexity symbol detectors with parallel structure for ISI channels [J]. IEEE Transactions on Communications，1994，42 (234)：1661－1671.

[3] VOGT J，FINGER A. Improving the max-log-MAP turbo decoder [J]. Electronics letters，2000，36 (23)：1937－1939.

[4] CHAIKALIS C，NORAS J M，RIERA-PALOU F. Improving the reconfigurable SOVA/log-MAP turbo decoder for 3GPP [C] //IEEE/IEE International Symposium on Communication Systems，Networks and DSP，2002.

[5] YUE D W，NGUYEN H H. Unified scaling factor approach for turbo decoding algorithms [J]. IET Communications，2010，4 (8)：905－914.

[6] CLAUSSEN H，KARIMI H R，MULGREW B. Improved max-log map turbo decoding using maximum mutual information combining [C] //14th IEEE Proceedings on Personal，Indoor and Mobile Radio Communications，2003. PIMRC 2003. IEEE，2003，1：424－428.

[7] SUN L，WANG H. Improved max-log-map turbo decoding by extrinsic information scaling and combining [C] //Communications，Signal Processing，and Systems：Proceedings of the 2018 CSPS Volume II：Signal Processing 7th. Springer Singapore，2020：336－344.

[8] NACHMANI E，BE'ERY Y，BURSHTEIN D. Learning to decode linear codes using deep learning [C] //2016 54th Annual Allerton Conference on Communication，Control，and Computing (Allerton). IEEE，2016：341－346.

[9] NACHMANI E，MARCIANO E，BURSHTEIN D，et al. RNN decoding of linear block codes [J]. Arxiv Preprint Arxiv：1702. 07560，2017.

[10] 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. IEEE，2017：1－6.

[11] LIANG F，SHEN C，WU F. An iterative BP-CNN architecture for channel decoding [J]. IEEE Journal of Selected Topics in Signal Processing，2018，12 (1)：144－159.

[12] VASIC B，XIAO X，LIN S. Learning to decode LDPC codes with finite-alphabet message passing [C] //2018 Information Theory and Applications Workshop (ITA). IEEE，2018：1－9.

[13] TENG C F，WU C H D，HO A K S，et al. Low-complexity recurrent neural network-based polar decoder with weight quantization mechanism [C] //ICASSP 2019-2019 IEEE International Conference on Acoustics，Speech and Signal Processing (ICASSP). IEEE，2019：1413－1417.

[14] KIM H，JIANG Y，RANA R，et al. Communication algorithms via deep learning [J]. Arxiv Preprint Arxiv：1805. 09317，2018.

[15] JIANG Y，KANNAN S，KIM H，et al. Deepturbo：deep turbo decoder [C] //2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)，IEEE，2019：1－5.

[16] JIANG Y，KIM H，ASNANI H，et al. Turbo autoencoder：deep learning based channel codes for point-to-point communication channels [J]. Advances in Neural Information Processing Systems，2019：32.

[17] 胡启蕾，许佳龙，李 伦，等. 信噪比自适应 Turbo 自编码器信道编译码技术 [J]. 无线电通信技术，2022，48 (4)：9.

[18] HE Y，ZHANG J，WEN C K，et al. TurboNet：a model-driven DNN decoder based on max-log-MAP algorithm for turbo code [C] //2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS). IEEE，2019：1－5.

[19] HE Y，ZHANG J，JIN S，et al. Model-driven DNN decoder for turbo codes：design，simulation，and experimental results [J]. IEEE Transactions on Communications，2020，68 (10)：6127－6140.

[20] HEBBAR S A，MISHRA R K，ANKIREDDY S K，et al. TinyTurbo：Efficient turbo decoders on edge [C] //2022 IEEE International Symposium on Information Theory (ISIT). IEEE，2022：2797－2802.

[21] BAHL L，COCKE J，JELINEK F，et al. Optimal decoding of linear codes for minimizing symbol error rate (corresp.) [J]. IEEE Transactions on Information Theory，1974，20 (2)：284－287.

[22] EBADA M，CAMMERER S，ELKELESH A，et al. Deep learning-based polar code design [C] //2019 57th Annual Allerton Conference on Communication，Control，and Computing (Allerton). IEEE，2019：177－183.

[23] JIANG Y，KIM H，ASNANI H，et al. Learn codes：Inventing low-latency codes via recurrent neural networks [J]. IEEE Journal on Selected Areas in Information Theory，2020，1 (1)：207－216.

[24] MAKKUVA A V，LIU X，JAMALI M V，et al. Ko codes：inventing nonlinear encoding and decoding for reliable wireless communication via deep-learning [C] //International Conference on Machine Learning. PMLR，2021：7368－7378.