TCLNet：一种利用语言模型作为无损压缩器进行CSI反馈的混合Transformer–CNN框架

Zijiu Yang、Qianqian Yang、Shunpu Tang、Tingting Yang 和 Zhiguo Shi

arXiv:2601.06588v1 [cs.IT] 10 Jan 2026

摘要——在频分双工（FDD）大规模多输入多输出（MIMO）系统中，下行链路信道状态信息（CSI）对于实现高频谱效率和高能量效率至关重要。然而，随着天线数量增加，CSI反馈开销逐渐成为主要瓶颈。尽管现有基于深度学习的CSI压缩方法展现出巨大潜力，但在捕获CSI的局部特征和全局特征方面仍存在局限，从而限制了可实现的压缩效率。为解决这些问题，我们提出TCLNet，这是一种统一的CSI压缩框架，结合了用于有损压缩的混合Transformer–CNN架构，以及用于无损压缩的混合语言模型（LM）与因子化模型（FM）设计。有损模块联合利用局部特征和全局上下文，而无损模块则在上下文感知编码和并行编码之间自适应切换，以优化率失真复杂度（RDC）权衡。在真实世界和仿真数据集上进行的大量实验表明，所提出的TCLNet在重建精度和传输效率方面优于现有方法，在多种场景下性能最高可提升5分贝。此外，我们还表明，通过精心设计的提示，大语言模型（LLM）可用作零样本CSI无损压缩器。

索引词—CSI反馈、混合Transformer-卷积神经网络架构、语言模型

I. 引言

A. 背景

大规模多输入多输出（MIMO）是5G的关键技术，预计在未来的6G系统中将发挥更加重要的作用。通过在基站部署数百甚至数千根天线，大规模MIMO能够显著提升频谱效率和能量效率，同时提高大量用户和设备的频谱效率 [2]。为充分实现这些优势，基站（BS）需要准确的下行链路信道状态信息（CSI），以支持预编码、资源分配和干扰管理 [3]。如图1所示，在时分双工（TDD）系统中，基站利用信道互易性直接估计下行信道状态信息根据上行链路测量结果。相比之下，频分双工（FDD）系统为上行链路和下行链路使用不同的频段。在这种情况下，仅存在弱互易性，因此很难根据上行链路 CSI 推断下行信道状态信息 [4]。因此，用户设备（UE）必须估计下行信道状态信息，然后将其反馈给基站。

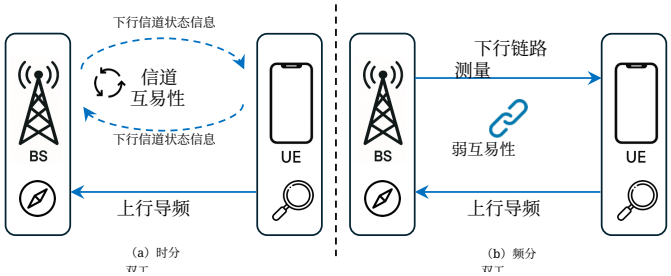


图1. 大规模MIMO系统中CSI反馈示意图。在频分双工系统中，由于上行链路与下行信道状态信息之间缺乏互易性，直接估计下行信道状态信息颇具挑战。

然而，随着大规模MIMO系统中的天线数量不断增加，CSI维度也相应增大，从而导致难以承受的反馈负担。为解决这一问题，研究人员开发了高效的CSI压缩与反馈方法，以降低开销，涵盖从压缩感知等经典方法[5], [6] 到较新的基于学习的技术 [7]–[13]。

B. 相关工作与研究动机

在多输入多输出（MIMO）研究的早期阶段，传统的信道状态信息（CSI）压缩方法（如压缩感知（CS））在 [14]–[17]中得到了广泛探索。然而，由于理想化假设与实际信道条件之间存在偏差，这些方法在实际应用中的性能受到限制。此外，大多数压缩感知算法都依赖迭代优化，导致计算复杂度较高，处理时延也不可忽视。为克服这些局限，近期研究开始探索基于学习的解决方案。具体而言，这些方法可分为两大类 [18]:

1) 模型驱动方法： 模型驱动方法遵循传统的压缩感知理论框架，但将其中一些步骤替换为神经网络，从而在提升性能的同时保持可解释性。一种常见策略是将迭代优化算法展开为神经网络架构。例如，ISTA-Net [19] 将用于压缩感知中稀疏恢复的经典求解器迭代收缩阈值算法（ISTA）展开为深度网络，其中变换矩阵和阈值函数直接从数据中学习，以改进信道状态信息重建。在此基础上，TiLISTA-Joint [20]等更先进的算法

本研究的部分成果曾于 2025 [1]年7月在加拿大温哥华举行的第42届国际机器学习会议（ICML）无线通信机器学习（ML4Wireless）研讨会上报告。Z. Yang、Q. Yang、S. Tang 和 Z. Shi 均任职于中国杭州 310027 浙江大学信息与电子工程学院（电子邮箱： {zijiu yang, qianqianyang20, tangshunpu, shizg}@zju.edu.cn）。T. Yang 任职于中国深圳 518066 鹏城实验室（电子邮箱：yangtt@pcl.ac.cn）。本文的通讯作者为 Q. Yang。

以及 FISTA-Net [21]通过引入改进的更新规则、自适应阈值和基于动量的加速技术，增强了展开式压缩感知框架，从而实现更快的收敛速度和更高的反馈效率。

尽管模型驱动策略具有较好的效果，但由于将迭代算法截断为固定数量的层，仍然面临诸多挑战。这种固定深度的设计使模型无法根据不同的信道条件调整迭代次数。因为当信道高度稀疏或压缩率较低时，需要更多迭代次数，而对于较为简单的情况，较少的迭代次数即可满足需求 [18]。更重要的是，这种截断削弱了原始优化算法在理论上的收敛保证。为缓解这些问题，近期的研究工作（如 Csi-L2O） [22]直接学习优化规则，而不是展开某个特定算法。通过在用户侧采用线性投影，并在基站采用基于LSTM的学习型优化器，该方法支持可变压缩率，并使更新规则能够泛化到不同的问题规模。然而，Csi-L2O仍然依赖逐步迭代更新，可能收敛较慢，并且在快速变化的信道条件下鲁棒性降低。

2）数据驱动方法： 与模型驱动方法不同，数据驱动方法直接从大规模CSI数据集中端到端地学习CSI压缩与重建。在这一方向上，文献 [7] 提出了首个基于端到端学习的CSI压缩框架，即 CSINet，并证明基于CNN的自编码器相比传统压缩感知方法能够显著提升CSI反馈性能。此后，研究者提出了大量扩展方法。例如，为了更好地建模空间多样性，文献 [8], [12], [23]–[25],引入了多分辨率技术，表明较大的感受野对于捕获多尺度传播特性至关重要。此外，ImCSiNet[26], 和 EVCsiNet [27] 强调了利用先进骨干网络提取更具表达力特征的重要性。为提升鲁棒性，文献 [11], [28], [29] 引入了自适应和特定领域的设计，以处理变化的信道条件。然而，这些方法在捕获CSI中的长程依赖关系方面仍然存在困难。尽管可以使用注意力机制来增强全局上下文聚合 [30]–[32],，其效果仍受到底层CNN架构的限制。

鉴于这一局限性，近期研究开始探索基于Transformer的架构 [33], ，因为其自注意力机制能够有效捕获长程相关性。具体而言，文献 [10] 中的作者提出了一种双层Transformer模块，用于提取具有信息量的全局CSI特征。然而，与基于CNN的设计相比，纯Transformer架构通常会带来更高的计算和内存开销，从而限制了其实际部署。为弥合这一差距，一个自然的方向是在统一框架内集成CNN和Transformer架构。然而，文献 [34] 中直接采用的串行CNN-Transformer设计限制了这种混合架构的潜在优势。相比之下，并行集成可能实现更优的复杂度与性能权衡，近期图像压缩领域的研究 [35]也证明了这一点。受此启发，我们旨在设计一种并行混合*Transformer–CNN*架构，用于*CSI*压缩，在全局建模能力与计算效率之间实现平衡。除此之外，我们还旨在利用无损压缩器进一步编码所获得的CSI低维表示，以减少反馈开销。大多数基于学习的方法并未明确考虑这一编码过程，或仅对压缩特征执行固定精度的标量量化或矢量量化 [12], [22], [36],[37],，这限制了其可实现的压缩效率。

因此，引入变长编码至关重要，因为它能够根据特征分布的统计结构分配比特，从而提高压缩效率。然而，其有效性取决于能否准确估计每个符号的概率。先前的工作（如 [38] ）使用因子化模型（FM）来估计此类概率，但FM假设符号之间具有统计独立性，忽略了CSI内部的上下文关系，导致编码效率欠佳。因此，能够捕捉*CSI*中复杂依赖关系的概率模型，是实现更有效变长编码的关键。

随着语言模型在复杂概率分布建模方面不断取得成功，使用语言模型作为强大的数据压缩概率估计器正受到越来越多的关注。例如，LMC [39] 表明，经过训练的语言模型（LM）可以通过将非文本数据展平为符号序列并预测其概率，用作通用压缩器进行熵编码。 [40]中的作者进一步探索了使用大语言模型（LLM）进行图像压缩的方法，引入视觉提示，引导模型更有效地捕捉图像残差结构，从而实现更准确的符号概率估计并提高压缩效率。此外，在密切相关的无线通信任务中，近期研究表明，LM在信道预测 [41]–[43] 和信道估计 [44],[45], 等任务中具有良好潜力，在不同场景下展现出强大的性能和泛化能力。尽管这些工作并未明确考虑CSI编码或压缩，但它们表明LM能够捕捉并理解信道特性。此外，当前已有多款智能手机配备了内置设备端语言模型，例如Vivo BlueLM[1] 以及iPhone上的Apple Intelligence[2] 。这些嵌入式语言模型还可以进一步用于支持和增强物理层通信任务。受上述研究启发，我们旨在利用*LM*构建表达能力更强的概率模型，以实现无损*CSI*编码。

C. 主要贡献

本文研究频分双工大规模MIMO-OFDM系统中的CSI反馈。我们提出了一种名为TCLNet的新型CSI压缩框架，由有损压缩器和无损压缩器组成。具体而言，对于有损压缩器，我们采用并行混合设计，将卷积神经网络与Transformer架构相结合，以捕获长程依赖关系和细粒度局部特征。为此，我们

[1]https://github.com/vivo-ai-lab/BlueLM [2] https://machinelearning.apple.com/research/ 苹果智能基础语言模型

基于一种称为转置卷积 模块的专用特征提取模块设计有损编码器和解码器，该模块将输入特征拆分为局部路径和全局路径。局部路径采用非对称卷积来捕获细粒度细节，而全局路径则采用Swin Transformer来建模长程依赖关系。通过残差连接融合这些特征，该架构降低了纯Transformer的计算开销，并取得了优于传统卷积神经网络的重建效果。

在有损压缩之后，我们提出一种基于语言建模、复杂度可变的无损压缩器，以进一步降低反馈开销。具体而言，我们首先采用经过适当缩放的上下文感知语言模型作为CSI 符号的分布预测器，从而提高熵编码效率；但由于其自回归特性，也会带来较高的计算时延。为缓解这一问题，我们引入一种轻量级因子分解模型（FM） [46], [47] ，并结合一种新颖的符号选择机制，通过自回归语言模型处理信息密集型符号、通过并行因子分解模型处理统计独立的符号，动态分配计算资源，从而明确应对率失真复杂度（RDC）之间的权衡。此外，我们还进一步探索大语言模型作为零样本 CSI 压缩器的潜力，并证明通过设计提示，大语言模型能够预测 CSI 符号。

本文的主要贡献总结如下：

- 我们提出一种并行混合 Transformer–卷积神经网络有损压缩器，充分发挥 Transformer 和卷积神经网络各自的互补优势，有效平衡计算复杂度与 CSI 重建性能。
- 我们提出了一种混合语言模型–因子分解模型无损压缩框架，通过一种新颖的符号选择机制，将符号动态分配给自回归语言模型或并行因子分解模型。该框架能够精细地控制计算复杂度，并实现高效的变长编码。此外，我们还探索了通过提示工程将大语言模型用于CSI无损压缩。
- 我们在真实世界数据集和仿真数据集上进行了大量实验，结果表明，与现有CSI反馈方法相比，所提出的框架在重建性能和压缩效率方面表现更优。此外，我们还证明，大语言模型无需任何特定领域的训练，即可作为零样本CSI无损压缩器。

II. 系统模型

本节介绍本文的系统模型。具体而言，我们首先介绍信道模型，然后说明CSI反馈过程。

A. 信道模型

我们考虑一个典型的FDD大规模MIMO-OFDM系统，其中基站配备了 $N_t \gg 1$ 根发射天线，每个用户设备配备一根接收天线。该系统带宽在频域被划分为 N_c 个子载波。令 $\boldsymbol{h}_n \in \mathbb{C}^{N_t \times 1}$ 表示下行信道增益向量，则第 n个子载波上的接收信号可表示为

$$y_n = \boldsymbol{h}_n^H \boldsymbol{w}_n x_n + \epsilon_n, \tag{1}$$

其中， $x_n \in \mathbb{C}$ 表示传输的数据符号， $\boldsymbol{w}_n$ 为预编码向量， $\epsilon_n \sim \mathcal{CN}(0, \sigma^2)$为加性白高斯噪声。

将空间频率域中的所有子载波堆叠起来后，整体 CSI 矩阵可表示为

$$\boldsymbol{H} = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,N_t-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_c-1,0} & h_{N_c-1,1} & \cdots & h_{N_c-1,N_t-1} \end{bmatrix} \tag{2}$$

其中， $\boldsymbol{H} \in \mathbb{C}^{N_c \times N_t}$ 表示 CSI 矩阵，其中第i行对应第 i 个子载波的信道系数，第 j列表示与第 j根发射天线相关的信道响应。在频分双工系统中，必须将此 CSI 矩阵反馈给基站，从而为每个子载波优化预编码向量 $\boldsymbol{w}_n$ ，以提升接收信号质量和系统频谱效率。然而，由于 CSI 矩阵的实数维度为 $2N_cN_t$，反馈开销会随着子载波数量 N_c 和发射天线数量 N_t的增加而显著增大。如此大量的反馈不仅会耗费大量上行传输资源，在未来的大规模MIMO-OFDM 系统中也难以实现。因此，有必要在反馈前压缩 CSI 矩阵。

B. CSI反馈

CSI反馈过程通常包括三个主要阶段：压缩、量化和重建。具体而言，根据 [7], ，我们首先使用二维离散傅里叶变换（DFT）将 CSI 从空间频率域变换到角度时延域，表示为

$$\boldsymbol{H}' = \boldsymbol{F}_c \boldsymbol{H} \boldsymbol{F}_t, \tag{3}$$

其中， $\boldsymbol{H}'$ 表示角度–时延域中经过变换的信道状态信息矩阵。此外， $\boldsymbol{F}_c \in \mathbb{C}^{N_c \times N_c}$ 和 $\boldsymbol{F}_t \in \mathbb{C}^{N_t \times N_t}$分别表示沿频率维度和空间维度的离散傅里叶变换矩阵。该变换将大部分信道能量集中到位于前 N_a 行（$N_a \leq N_c$）的少数主导系数上，而其余元素几乎为零。因此，只保留 $\boldsymbol{H}'$ 的前 N_a 行，即可对其进行近似，从而得到低维代表矩阵 $\boldsymbol{H}_a \in \mathbb{C}^{N_a \times N_t}$。接下来，为进一步降低反馈开销，使用基于学习的自编码器框架对低维矩阵 $\boldsymbol{H}_a$ 进行压缩和量化。具体而言，我们分别用 $E_l(\cdot)$ 和 $E_{nl}(\cdot)$ 表示用户设备处的有损编码器和无损编码器，并用 $Q(\cdot)$ 和 $Q_d(\cdot)$ 表示量化器和反量化器。采用预定义的

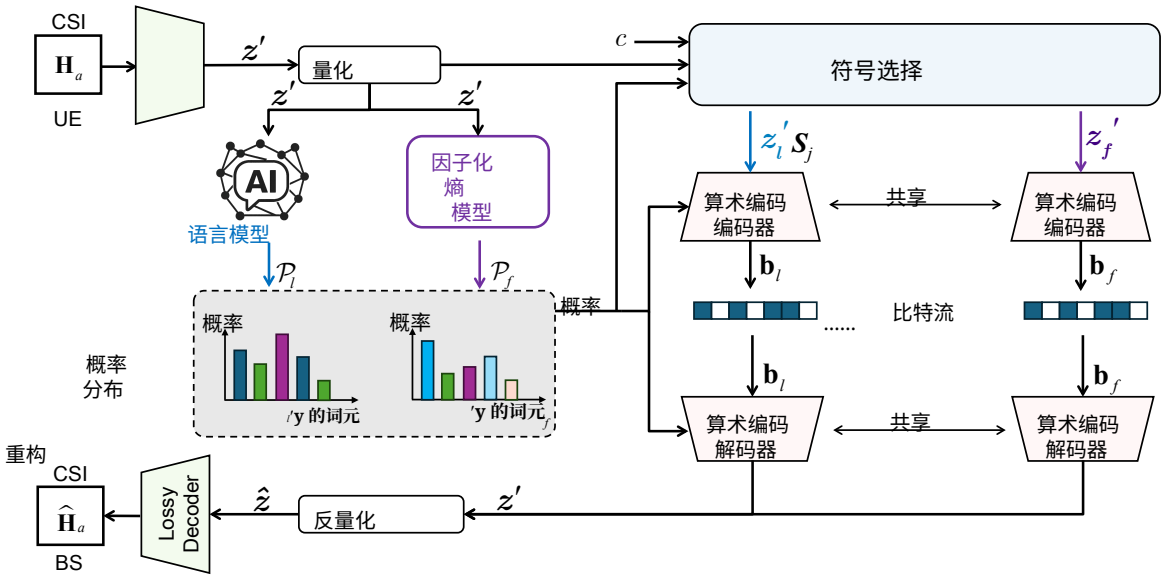


图2. 所提出的TCLNet示意图。该网络由基于Transformer-卷积神经网络混合架构的有损编码器，以及基于语言模型和频率模型的无损编码器组成。有损编码器用于压缩CSI的数量，而无损编码器用于估计信息熵并进行编码。

压缩比 CR 进行压缩以及量化过程可表示如下

$$\boldsymbol{z} = E_l(\boldsymbol{H}_a), \tag{4}$$
$$\boldsymbol{z'} = Q(\boldsymbol{z}), \tag{5}$$
$$\boldsymbol{b} = E_{nl}(\boldsymbol{z'}), \tag{6}$$

其中， $\boldsymbol{z} \in \mathbb{R}^z$ 表示压缩后的CSI特征，其 $z = N_a \times N_t \times$压缩率为CR； $\boldsymbol{z'} \in \mathbb{R}^Z$ 表示量化符号，$\boldsymbol{b}$ 表示用于反馈的量化比特流。

在接收端，采用有损解码器 $D_l(\cdot)$和无损解码器$D_{nl}(\cdot)$，根据接收到的$\boldsymbol{b}$重建CSI，表示为

$$\boldsymbol{z'} = D_{nl}(\boldsymbol{b}), \tag{7}$$
$$\hat{\boldsymbol{z}} = Q^{-1}(\boldsymbol{z'}), \tag{8}$$
$$\hat{\boldsymbol{H}}_a = D_l(\hat{z}), \tag{9}$$

其中， $\boldsymbol{z'}$ 和 $\boldsymbol{H}'_a$ 分别表示解码后的向量和重构的CSI。

采用这种架构，关键挑战在于如何设计$E_l(\cdot)$ $E_{nl}(\cdot)$ 和 $D_l(\cdot)$ $D_{nl}(\cdot)$，以实现高效的CSI压缩与重建。然而，大多数先前研究 [8]-[10], [12], [23] 仅关注定长编码，在固定比特率下改善归一化均方误差，却忽略了速率控制。因此，有必要联合考虑比特率、失真和计算复杂度，以解决CSI反馈中的率失真复杂度权衡。

III. 所提出的TCLNET

本节将介绍所提出的用于高效CSI反馈的TCLNet。具体而言，我们首先概述TCLNet，随后详细介绍其有损压缩器和无损压缩器。接下来，我们将讨论TCLNet的训练策略，并分析所提出方法的复杂度。

A. 概述

如图2所示，所提出的TCLNet由基于Transformer–卷积神经网络的混合有损压缩器和基于LM–FM的混合无损压缩器组成。有损压缩器由有损编码器和有损解码器构成，旨在从高维信道状态信息中提取最具信息量的特征，而无损压缩器则进一步将量化潜在特征压缩为用于传输的比特流。具体而言，在发射端，有损编码器首先对信道状态信息进行编码 $\boldsymbol{H}_a$ ，得到紧凑表示 $\boldsymbol{z}$，随后将其量化为离散符号 $\boldsymbol{z'}$。这些符号随后被输入无损压缩器，该压缩器包括使用LM和FM模块以及算术编码器实现的两个并行分支。

语言模型（LM）和频率模型（FM）分别表示语言模型和因子化模型，它们为熵编码生成概率分布 $\mathcal{P}_l$和 $\mathcal{P}_f$ 。随后，将这些分布（$\mathcal{P}_l$， $\mathcal{P}_f$）、复杂度变量 c和量化特征 $\boldsymbol{z'}$ 输入符号选择模块。该模块生成选择指示符 $\boldsymbol{S}_j$。具体而言， $\boldsymbol{S}_j$ 表示 $\boldsymbol{z'}$ 中由算术编码器使用 $\mathcal{P}_l$ 进行编码、以生成比特流 $\boldsymbol{b}_l$的元素子集，其余的 $\boldsymbol{z'}$ 则使用 $\mathcal{P}_f$ 编码，以生成 $\boldsymbol{b}_f$。在接收端，算术解码器执行解码过程。它以LM为条件自回归解码比特流 $\boldsymbol{b}_l$ ，同时以FM为条件解码 $\boldsymbol{b}_f$ ，从而恢复量化符号 $\boldsymbol{z'}$。最后，对恢复的 $\boldsymbol{z'}$ 进行反量化，并将其传递给有损解码器ˆ，以获得重构的CSI $\boldsymbol{H}_a$。

B. 混合式*Transformer-CNN*有损压缩器

为有效应对现有研究中捕获信道状态信息矩阵的局部空间特征和长程依赖关系所面临的挑战， [9], [10],我们提出了一种新型

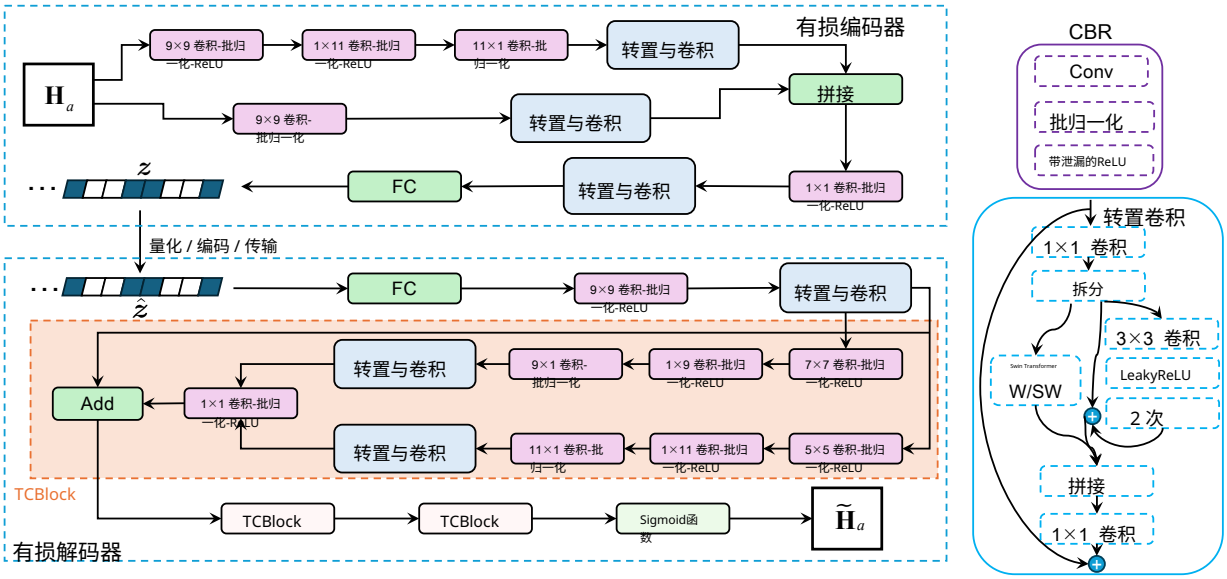


图3. 所提出的混合式Transformer-CNN有损压缩器示意图。该压缩器由卷积神经网络和Swin Transformer组成，分别用于从CSI中提取局部和全局特征。

混合式基于Transformer–CNN的有损压缩器，如图3所示，该压缩器充分发挥卷积神经网络和Transformer各自的优势。所提出的有损压缩器由两个主要组件构成：部署于用户设备的有损编码器和部署于基站的有损解码器。

1）编码器架构：在UE侧，CSI矩阵$\boldsymbol{H}_a$首先输入两个并行的特征提取分支，每个分支均旨在捕获CSI信息的互补特征。这种双分支设计实现了多尺度表示，并增强了整体特征提取能力 [8], [23]。在第一个分支中，依次应用三个带批归一化和ReLU（CBR）操作的卷积块，逐步扩大感受野，并捕获不同尺度的空间相关性。第一个CBR块采用标准的 9×9 卷积核，随后接入两个非对称CBR，其卷积核尺寸分别为 1×11 和11×1。这是因为非对称卷积核是可分离卷积的一种高效替代方案，能够在保持计算效率的同时有效捕获空间依赖关系。这些CBR的输出随后输入转置卷积模块，该模块是用于捕获全局和局部特征的混合式Transformer–CNN块。稍后将详细介绍转置卷积模块的结构。第二个分支负责上下文特征提取，首先经过一个 9×9 CBR，然后接入另一个转置卷积模块。两个分支的输出经拼接后，通过一个 1×1 CBR进行融合，以实现通道级整合；随后接入最终的转置卷积模块，进一步精炼特征。最后，全连接层（FC）根据所需的压缩率CR，将精炼后的特征图压缩为紧凑的特征向量序列$\boldsymbol{z}$。然后对这些特征向量进行量化并传输。

2) 解码器架构： 在BS侧，首先使用全连接层将反量化特征向量投影回其原始大小，随后经过一个 9×9 CBR和一个 TransConv 模块以融合局部和全局信息。此外，我们设计了三个串联的*TCBlock*模块，以逐步重构CSI矩阵。与编码器类似，每个TCBlock模块包含两个并行分支。第一分支依次应用 7×7、 1×9和 9×1 卷积，随后连接一个*TransConv*模块；第二分支采用 5×5、 1×11和11×1 CBR模块，随后连接另一个*TransConv* 模块。两个分支的输出经过拼接后，通过一个 1×1 卷积块进行特征融合。受残差学习启发 [48]，我们在每个TCBlock模块的输入与输出之间添加跳跃连接，以促进信息流动并提高训练稳定性。经过全部三个TCBlock模块后，通过Sigmoid激活函数获得重构的CSI矩阵，从而确保重构值落在原始CSI的范围内。

3) TransConv 模块的详细介绍：作为所提出 TCLNet 的核心组成部分，TransConv 模块在有效捕获 CSI数据中的局部特征和全局特征方面发挥着关键作用。受图像压缩中并行使用 TransConv 的启发， [35], 本文进一步介绍了用于 CSI压缩的 TransConv 模块架构，如图3右侧所示。具体而言，设 TransConv 模块的输入表示为 $\boldsymbol{F} \in \mathbb{R}^{2\times h\times w}$ 其中，两个通道分别来自 CSI特征图的实部和虚部。首先，我们使用一个 1×1 卷积进行特征通道扩展，将特征通道数从 2 增加到 r（$r > 2$），从而增强特征表示能力，表示为

$$\boldsymbol{F}' \in \mathbb{R}^{r\times h\times w} = \mathrm{Conv}_{1\times1}(\boldsymbol{F}), \tag{10}$$

其中，$\mathrm{Conv}_{1\times1}(\cdot)$ 表示通道扩展操作， $\boldsymbol{F}'$表示扩展特征图。随后，特征表示经过通道拆分操作，其中

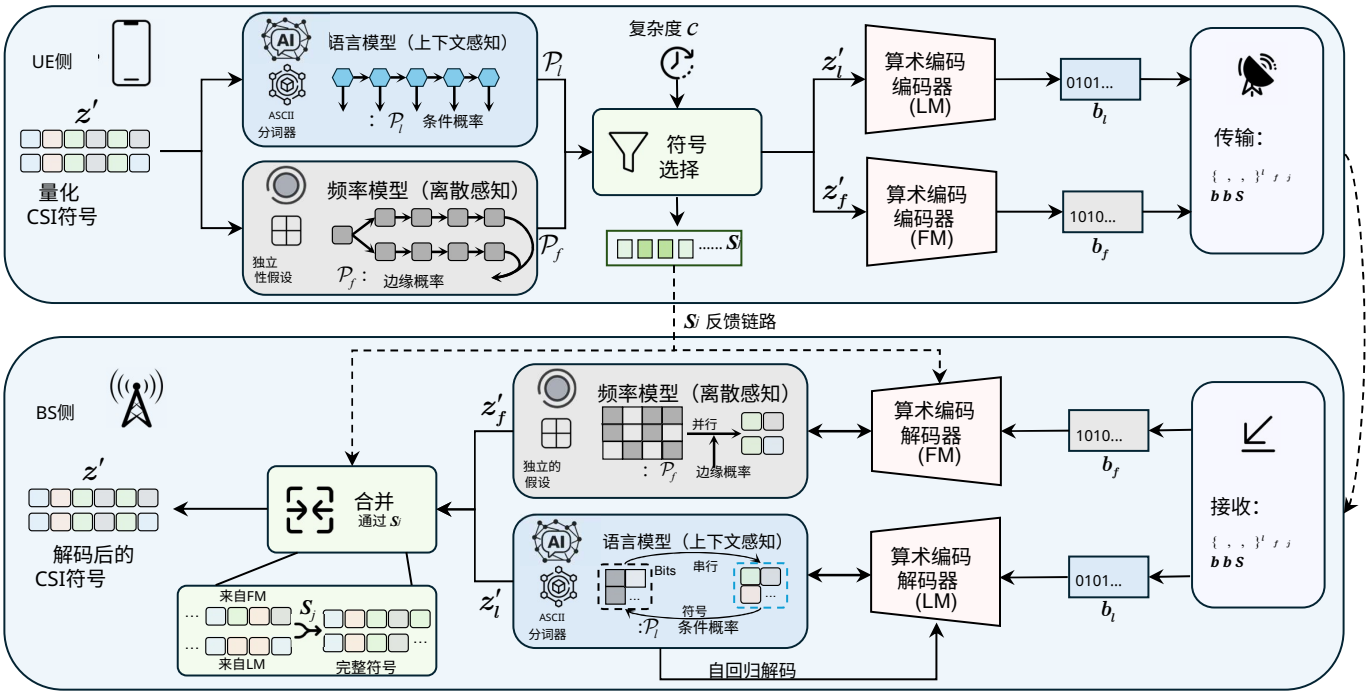


图4. 所提出的基于LM和FM的编码方案示意图，其中LM对具有统计依赖关系的潜在符号进行编码，而FM对相对独立的符号进行编码。

各通道几乎均等地分为两部分，以实现全局和局部特征提取，可表示为

$$\boldsymbol{F}_g, \boldsymbol{F}_l = \text{Split}(\boldsymbol{F}'), \tag{11}$$

其中，$\boldsymbol{F}_l \in \mathbb{R}^{r_l \times h \times w}$，$\boldsymbol{F}_g \in \mathbb{R}^{r_g \times h \times w}$ 分别表示输入局部路径和全局路径的特征。局部分支 $\boldsymbol{F}_l$ 经过两个连续的小卷积核卷积处理，然后与原始输入进行残差连接，以获得局部增强特征 $\boldsymbol{F}_l'$。与此同时，全局分支 $\boldsymbol{F}_g$ 经过Swin Transformer模块，以捕获长程依赖关系和全局上下文，从而获得全局表示 $\boldsymbol{F}_g'$。随后，将两个分支的输出 $\boldsymbol{F}_o \in \mathbb{R}^{r \times h \times w}$ 拼接并通过 1×1 卷积进行投影，再通过另一个残差连接与原始输入融合，表示为

$$\boldsymbol{F}_o = \text{Conv}_{1\times1}(\text{concat}(\boldsymbol{F}_l', \boldsymbol{F}_g')) + \boldsymbol{F}. \tag{12}$$

C. 基于*LM*和*FM*的无损压缩器

通过有损编码器将信道状态信息压缩为紧凑的特征向量后，首先对该表示进行量化，然后将其传递给无损压缩器，由后者生成用于传输的最终比特流。如图4所示，无损压缩器通过两个概率模型处理量化符号，随后进行符号选择，再执行算术编码。下面将详细介绍无损压缩器的各个组成部分。

1）量化与概率模型：有损编码器生成的CSI特征向量 $\boldsymbol{z}$ 首先经过一个 n 比特均匀量化器处理，其定义如下：

$$\begin{aligned} \Delta &= \frac{\boldsymbol{z}_{max} - \boldsymbol{z}_{min}}{2^n - 1}, \\ \boldsymbol{z}' &= Q(\boldsymbol{z}) = \left\lfloor \frac{\boldsymbol{z} - \boldsymbol{z}_{min}}{\Delta} \right\rfloor, \end{aligned} \tag{13}$$

其中，Δ 表示量化步长，$\boldsymbol{z}_{max}$ 和 $\boldsymbol{z}_{min}$分别表示 $\boldsymbol{z}$的最大值和最小值，$\boldsymbol{z}'$ 表示量化后的特征向量，由 2^n 种离散符号组成，序列长度为 z。

随后，这些符号被并行送入LM和FM，从而得到相应的基于上下文的概率分布 $\mathcal{P}_l$ 和离散概率分布 $\mathcal{P}_f$。具体而言，LM记作LM(·)，首先将离散符号转换为ASCII令牌序列，然后预测其依赖于上下文的概率，表示为

$$\mathcal{P}_l = \text{LM}(\boldsymbol{u}) = \left\{\boldsymbol{\rho}_{0|d}, \boldsymbol{\rho}_{1|0,d}, \ldots, \boldsymbol{\rho}_{z|<z}\right\}, \tag{14}$$

其中，$\boldsymbol{u} = \text{ASCII}(\boldsymbol{z}')$是通过ASCII 分词从量化符号 $\boldsymbol{z}'$ 得到的词元序列。此外，符号 $\boldsymbol{\rho}_{z|<z} \in \mathbb{R}^{|\mathbf{A}|}$表示第 z个符号的基于上下文的概率分布，其中第 z个词元是在给定其所有前置 $z-1$个词元的条件下预测的。这里，$\mathbf{A}$ 表示LM的词汇表，满足$|\mathbf{A}| > 2^n$，而 d 表示用于序列初始化或填充的虚拟令牌。

此外，我们还采用轻量级频率模型来估计量化符号的无上下文离散概率分布。该频率模型假设符号之间统计独立，

并仅对它们的边缘分布进行建模。因此，频率模型产生的概率分布 $\mathcal{P}_f$ 为

$$\mathcal{P}_f = \mathrm{FM}(\mathbf{z}') = \{\pi_0, \pi_1, \ldots, \pi_{2^n-1}\}, \tag{15}$$

其中，π_i 表示第 i个符号类别的预测边缘概率，$\sum_{i=0}^{2^n-1} \pi_i = 1$。

2）符号选择：接下来，我们介绍所提出的符号选择机制。该机制旨在利用 LM 自适应地选择符号进行基于上下文的压缩，同时将其余符号分配给 FM 进行并行处理。具体而言，首先根据 $\mathcal{P}_l$ 计算 $\boldsymbol{z}'$ 中每个词元的基于上下文的熵，并根据 $\mathcal{P}_f$ 计算整个符号序列的信息熵，如下所示：

$$H_{l,k}(\boldsymbol{z}'_k \mid \boldsymbol{z}'_{<k}) = -\sum_{a\in\mathbf{A}} \rho_{k|<k}(a)\, \log_2 \rho_{k|<k}(a), \tag{16}$$

$$H_f(\boldsymbol{z}') = -\sum_{s=0}^{2^n-1} \pi_s \log_2 \pi_s, \tag{17}$$

其中，$H_{l,\,k}(\boldsymbol{z}'_k \mid \boldsymbol{z}'_{<k})$ 表示第 k个词元的基于上下文的熵，$H_f(\boldsymbol{z}')$ 表示整个符号序列的离散熵，$\rho_{k|<k}(a)$ 表示词元 k 取词汇表条目 a的基于上下文的概率。随后，可以计算熵差，如下所示：

$$D_k = H_f(\boldsymbol{z}') - H_{l,k}(\boldsymbol{z}'_i|\boldsymbol{z}'_{<i}), \tag{18}$$

其中，D_k 表示第 k个词元的熵差。

根据计算得到的熵差，我们可以确定哪些符号应使用语言模型或频率模型进行编码。为此，我们引入复杂度变量 $c \in (0,1)$，用于指定分配给频率模型并行处理的符号比例。首先根据熵差对所有词元进行排序。选取熵差最大的前 $(1-c)$ 比例的词元作为 $\boldsymbol{z}'_l$，并用 $\boldsymbol{S}_j$ 表示所选符号对应的指示变量。其余熵差较小的 c比例词元则归入 $\boldsymbol{z}'_f$。这是因为，如果某个词元满足

$$H_{l,k}(\boldsymbol{z}'_k|\boldsymbol{z}'_{<k}) \approx H_f(\boldsymbol{z}'), \tag{19}$$

其依赖上下文的熵相比无上下文熵几乎没有改善，意味着上下文建模带来的压缩增益微乎其微。因此，这些词元被分配到频率模型分支，以加快解码速度。相比之下，当

$$H_{l,k}(\boldsymbol{z}'_k|\boldsymbol{z}'_{<k}) \ll H_f(\boldsymbol{z}'), \tag{20}$$

由于上下文依赖关系较强，语言模型能够显著提高概率估计精度，因此这些词元将使用语言模型进行编码。

3）算术编码器与反量化：获得概率分布和选定的符号集后，应用统计编码阶段来压缩量化符号。具体而言，我们采用有限精度算术编码器 [39], [49] 对 $\boldsymbol{z}'_f$ 和 $\boldsymbol{z}'_l$进行编码，得到相应的比特流 $\boldsymbol{b}_f$ 和 $\boldsymbol{b}_l$。算术编码过程通过根据预测的符号概率，逐步缩小$[0,1)$ 中的一个区间。在编码开始时，区间初始化为 $I_0 = [0,1)$。在每个编码步骤中，当前区间 $I_{t-1} = [l_{t-1}, h_{t-1})$ 会被划分为一组互不重叠的子-˜区间 $\{I_t(x)\}_{x\in\mathbf{A}}$，其中 l 和 h 分别表示区间的下界和上界。每个子区间都对应预定义字母表中的一个不同符号$\mathbf{A}$。这些子区间的大小与给定前文上下文时各符号的条件概率成正比，可表示为

$$\tilde{I}_t(x) = \Bigg[l_{t-1} + (h_{t-1} - l_{t-1}) \times \sum_{m<x} \rho\,(m \mid x_{<t})\,, \quad l_{t-1} + (h_{t-1} - l_{t-1}) \times \sum_{m\le x} \rho\,(m \mid x_{<t}) \Bigg). \tag{21}$$

编码器随后选取与当前符号对应的子区间，并将其作为下一步更新后的区间。处理完所有符号后，最终区间以二进制小数表示，从而得到压缩后的比特流。接下来，我们可以获得两个比特流 $\boldsymbol{b}_l$ 和 $\boldsymbol{b}_f$，并将它们连同指示变量 $\boldsymbol{S}_j$ 一起传输给接收端。

在解码器端，首先使用指示矩阵 $\boldsymbol{S}_j$ 将两个比特流 $\boldsymbol{b}_l$ 和 $\boldsymbol{b}_f$ 分离。LM 编码流 $\boldsymbol{b}_l$ 由 LM 迭代解码：LM 预测每个字符的概率分布，算术解码器则根据预测的分布进行解码。解码过程从一个预定义的占位符词元开始，以初始化 LM。随后，LM 预测第一个符号的概率分布，并将该分布与接收到的比特流 $\boldsymbol{b}_l$ 一并提供给算术解码器，以恢复第一个 ASCII 字符。恢复出的字符会被追加到上下文中，并由 LM 用于预测下一个字符的分布。此迭代过程持续进行，直至重构出完整的 ASCII 序列，随后进行 ASCII 去词元化，以获得相应的量化值 q_l。对于 FM 编码流 $\boldsymbol{b}_f$，可以使用统一的概率分布进行并行解码，从而获得相应的量化值 q_f。然后，根据指示矩阵 $\boldsymbol{S}_j$，将这些量化值 q_l 和 q_f 合并回量化信道状态信息特征 $\boldsymbol{z}'$。接着，我们进行反量化并重构信道状态信息特征向量，表示为

$$\hat{\boldsymbol{z}} = Q_d(\boldsymbol{z}') = \boldsymbol{z}' \cdot \Delta + \boldsymbol{z}_{min}. \tag{22}$$

ˆ 最后，将 ˆ$\boldsymbol{z}$ 输入有损解码器，即可得到重构的 CSI矩阵 $\boldsymbol{H}_a$。

4) 接收端复杂度讨论：现在简要讨论所提出的 TCLNet 的解码复杂度。由于大部分计算负担来自无损解码过程，而有损解码器可通过 GPU 高效加速，因此我们将分析重点放在无损解码过程上。如图4所示，我们定义 T_{lm} 和 T_{fm} 分别表示 LM 和 FM 对每个词元的推理时间。此外，令 Z 表示待解码词元的总数，c 表示分配给 FM 进行并行解码的词元比例。因此，无损解码过程的时延可表示为

$$T_{\mathrm{lossless}} = \max\Big\{ \underbrace{(1-c)\,Z\,T_{\mathrm{lm}}}_{\mathrm{LM}}, \underbrace{c\,Z\,T_{\mathrm{fm}}}_{\mathrm{FM}} \Big\}. \tag{23}$$

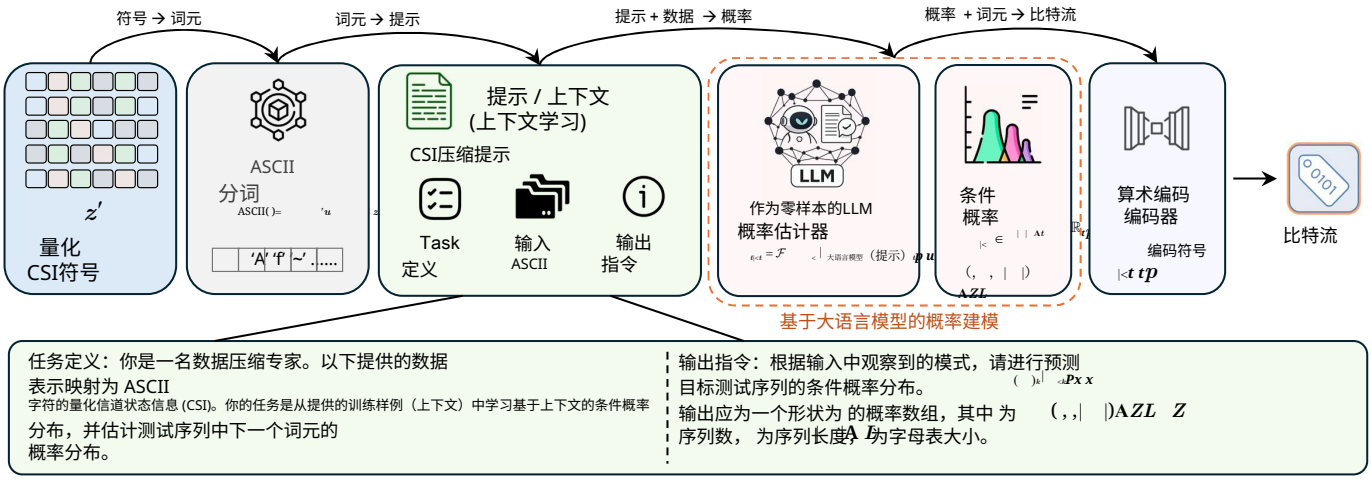


图5. 所提出的基于大语言模型的无损压缩器示意图，其中采用了 ASCII 分词和上下文学习，并结合了大语言模型与算术编码器。

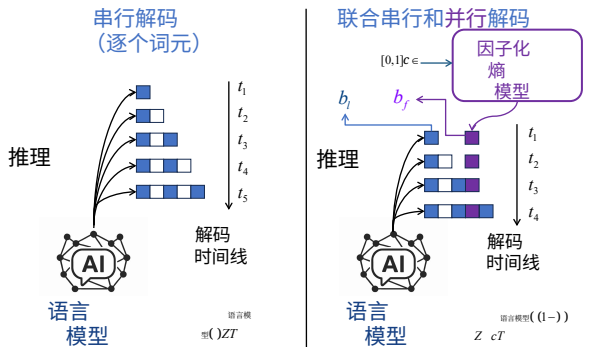


图6. 所提出的串行与并行联合解码方案。频率模型并行解码相应的词元，而语言模型串行解码词元，两者之间的比例由 c决定。

其中，该项表示 LM 与 FM 解码过程之间的最大时延，因为二者并行运行。需要指出的是，FM 以并行方式执行，因此其时延远低于 LM，即 $T_{\mathrm{fm}} \ll T_{\mathrm{lm}}$。因此，我们可以将整体无损解码时延简化为

$$T_{\mathrm{lossless}} \approx (1-c)\, Z\, T_{\mathrm{lm}} \leq Z\, T_{\mathrm{lm}}. \tag{24}$$

与需要$Z\,T_{\mathrm{lm}}$ 时间的完全自回归解码相比，所提出的串行与并行联合解码方案有效地将解码时延降低了与 c成比例的倍数。这提升了所提出的 TCLNet 在平衡压缩性能与解码速度方面的效率。

D. 大语言模型作为零样本无损压缩器

尽管基于语言模型的无损压缩器具有较高的压缩效率，但它需要在特定领域的信道状态信息数据集上进行训练。受大语言模型强大的上下文学习能力启发，我们进一步研究了它们作为信道状态信息反馈零样本无损压缩器的潜力，使其能够对信道状态信息符号的统计依赖关系进行建模无需进行任何参数更新。所提出的基于大语言模型的无损压缩器的详细信息见图5。

1) 工作流程与提示工程：为适应大语言模型的文本输入空间，我们采用与基于语言模型的压缩器相同的ASCII分词方案，将量化信道状态信息 z' 转换为离散的字符词元。概率估计任务被表述为下一个词元预测，由大语言模型推断目标序列的统计结构。为使大语言模型充当概率估计器而非文本生成器，我们设计了一个简单、结构化的提示，说明任务、输入格式以及所需的输出。

2) 概率提取与编码：接下来，我们使用大语言模型估计算术编码所需的条件概率分布。具体而言，给定输入词元序列 $\boldsymbol{u}$，在每个时间步 t，大语言模型会根据前文上下文 $\boldsymbol{u}_{<t}$ 和精心设计的提示，输出条件概率向量 $\boldsymbol{p}_t$ ，即

$$\boldsymbol{p}_t = \mathcal{F}_{\mathrm{LLM}}(\boldsymbol{u}_{<t} \mid \mathrm{Prompt}), \tag{25}$$

其中， $\mathcal{F}_{\mathrm{LLM}}$ 是大语言模型的前向函数。输出向量 $\boldsymbol{p}_t \in \mathbb{R}^{|\mathbf{A}|}$ 表示词汇表$\mathbf{A}$ 上位置t处下一个词元的预测概率分布。最后，这些概率向量 $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_L\}$ 依次输入算术编码器，将整个序列编码为紧凑的比特流以供传输。

IV. 实验结果

本节通过实验评估所提出的TCLNet框架的性能。我们首先介绍实验设置，然后从重建精度和计算复杂度两方面展示并分析实验结果。此外，我们还进行了消融研究，以验证TCLNet中关键组件的有效性。

A. 实验设置

1) 数据集：我们在真实和仿真的多输入多输出信道上评估所提出的 TCLNet 的性能

表 I ARGOS 室内场景下的性能比较。粗体表示最佳结果，下划线表示次佳结果。

CR		1/8		1/16		1/32		1/64		1/128	
方法	架构	浮点运算次数	NMSE	浮点运算次数	NMSE	浮点运算次数	NMSE	浮点运算次数	NMSE	浮点运算次数	NMSE
CRNet	CNN	41.38M	-18.12	28.92M	-16.01	22.69M	-11.80	19.58M	-10.44	18.02M	-8.36
CLNet	CNN	36.81M	-21.80	24.36M	-16.93	18.13M	-13.39	15.02M	-9.16	13.46M	-6.10
TransNet	Transformer	822.80M	-19.50	810.34M	-17.45	804.11M	**-15.95**	801.00M	-13.52	799.44M	-11.84
TCLNet	Trans&CNN	310.92M	**-22.64**	298.46M	**-17.70**	292.23M	-15.31	289.12M	**-14.27**	285.18M	**-11.93**
7 bit-CRNet	CNN	$+O(Q_7)$	-18.07	$+O(Q_7)$	-15.97	$+O(Q_7)$	-11.78	$+O(Q_7)$	-10.28	$+O(Q_7)$	-8.22
7 位 CLNet	CNN	$+O(Q_7)$	-21.52	$+O(Q_7)$	-16.93	$+O(Q_7)$	-13.36	$+O(Q_7)$	-9.19	$+O(Q_7)$	-6.09
7 位 TransNet	Transformer	$+O(Q_7)$	-19.37	$+O(Q_7)$	-17.42	$+O(Q_7)$	**-15.80**	$+O(Q_7)$	-13.63	$+O(Q_7)$	-11.82
7 bit-TCLNet	Trans&CNN	$+O(Q_7)$	**-22.41**	$+O(Q_7)$	**-17.64**	$+O(Q_7)$	-15.26	$+O(Q_7)$	**-14.22**	$+O(Q_7)$	**-11.91**
6 bit-CRNet	CNN	$+O(Q_6)$	-17.91	$+O(Q_6)$	-15.87	$+O(Q_6)$	-11.74	$+O(Q_6)$	-10.24	$+O(Q_6)$	-8.18
6位CLNet	CNN	$+O(Q_6)$	-20.76	$+O(Q_6)$	-16.73	$+O(Q_6)$	-13.28	$+O(Q_6)$	-9.16	$+O(Q_6)$	-6.08
6位TransNet	Transformer	$+O(Q_6)$	-19.06	$+O(Q_6)$	-17.21	$+O(Q_6)$	**-15.67**	$+O(Q_6)$	-13.58	$+O(Q_6)$	-11.78
6位TCLNet	Trans&CNN	$+O(Q_6)$	**-21.80**	$+O(Q_6)$	**-17.47**	$+O(Q_6)$	-15.18	$+O(Q_6)$	**-14.15**	$+O(Q_6)$	**-11.84**
5位CRNet	CNN	$+O(Q_5)$	-17.34	$+O(Q_5)$	-15.45	$+O(Q_5)$	-11.59	$+O(Q_5)$	-10.09	$+O(Q_5)$	-8.02
5 bit-CLNet	CNN	$+O(Q_5)$	-18.61	$+O(Q_5)$	-15.99	$+O(Q_5)$	-12.97	$+O(Q_5)$	-9.02	$+O(Q_5)$	-6.02
5 bit-TransNet	Transformer	$+O(Q_5)$	-17.93	$+O(Q_5)$	-16.17	$+O(Q_5)$	**-15.61**	$+O(Q_5)$	-13.38	$+O(Q_5)$	**-11.62**
5 bit-TCLNet	Trans&CNN	$+O(Q_5)$	**-20.06**	$+O(Q_5)$	**-16.85**	$+O(Q_5)$	-14.87	$+O(Q_5)$	**-13.87**	$+O(Q_5)$	-11.58

注释： 为说明有损压缩器的有效性，此处我们不使用所提出的无损压缩器 .

数据集。对于真实数据，我们采用了 Argos数据集[50]，，该数据集是在各种室内场所通过 2.4 GHz 频段采集的。对于仿真数据，我们使用 COST 2100 模型生成的信道 [7] ，涵盖室内和室外环境，并采用标准参数设置。

2）实现细节： 我们使用 PyTorch 实现所提出的 TCLNet。有损压缩网络使用 Adam 优化器和余弦退火学习率调度进行训练，初始学习率和最低学习率分别设为 2×10^{-3} 和 5×10^{-5}。训练过程持续 500 个训练轮次，其中包括 20 个训练轮次的预热。LM 实例化为仅解码器 Transformer，嵌入维度为 256，词汇表大小为 256。它由四个 Transformer 块组成，每个块配备八个注意力头，前馈网络维度为 1024。LM 使用 Adam 进行训练，学习率固定为 10^{-4} ，批量大小为 16，共迭代 8000 次。

3）基线方法： 我们将所提出的 TCLNet 与若干具有不同复杂度的代表性深度学习 CSI 反馈方法进行比较。对于低复杂度方法，我们与三种基于 CNN 的网络进行比较：CsiNet [7],CRNet [8], 和 CLNet [9]。对于更高复杂度的方法，我们将其与 TransNet [10] 和 CsiNet+ [12]进行比较。

4）评估指标：我们从三个方面评估CSI反馈性能：重建精度、计算复杂度和传输效率。重建精度采用归一化均方误差（NMSE）进行衡量，其定义如下：

$$\mathrm{NMSE}=\mathbb{E}\left[\frac{\|\boldsymbol{H}_a-\hat{\boldsymbol{H}}_a\|_2^2}{\|\boldsymbol{H}_a\|_2^2}\right]. \tag{26}$$

计算复杂度通过推理过程中所需的浮点运算次数（ FLOPs）进行量化。传输效率通过每个 CSI 值所传输的比特数进行评估。令 N 表示 CSI 元素数量， [7]表示传输的比特总数，则每个 CSI 值的比特率可表示为

$$\text{Bit Rate}=\frac{B}{N}. \tag{27}$$

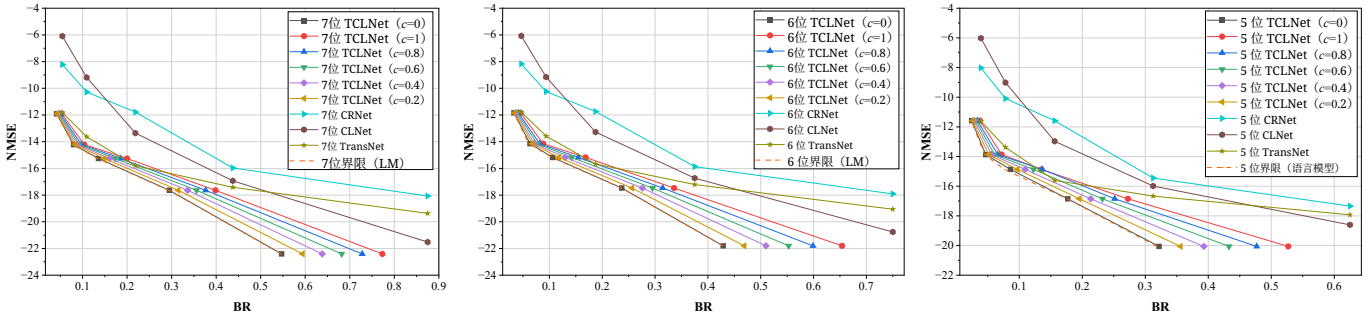
B. 真实世界数据集上的结果

在表I中，我们在真实世界的Argos数据集上比较了TCLNet与基线方法的性能，比较指标为NMSE和浮点运算次数（FLOPs），压缩率（CR）设为1/8、1/16、1/32、1/64和1/128。输入CSI矩阵的维度为 96×52，与先前工作中常用的32×32 表示形式相比，它更能代表实际的大规模MIMO系统，例如 [10]。从表中可以看出，在所有CR设置下，所提出的TCLNet始终优于基于CNN的方法；在大多数场景下，它也优于纯Transformer方法TransNet，且计算复杂度显著更低。具体而言，在CR为1/8时，与基于CNN的方法相比，所提出的TCLNet最多可将NMSE改善4分贝。此外，与采用纯Transformer架构的TransNet相比，所提出的TCLNet也取得了1–2分贝的提升，而其FLOPs仅约为TransNet的1/3。在较低的CR范围内，例如CR=1/64和1/128，所提出的TCLNet仍能以低得多的复杂度，比TransNet提升约0.5分贝。这是因为混合式转置与卷积架构能够有效捕获CSI的全局和局部特征，同时避免深层Transformer模型带来的高计算开销。这些结果证明了所提出的TCLNet的有效性，以及其在实际大规模MIMO系统中部署的适用性。

在表II中，我们比较了所提出的 TCLNet 与传统定长编码方法在 Argos数据集上采用 7 位量化时的无损编码性能，其中 CR 从 1/8 变化到 1/128，且 c设为 0、0.2、0.4、0.6、0.8 和 1。我们报告了各方法的解码时间复杂度以及

表 II TCLNet 与传统定长编码在采用 7 位均匀量化的 ARGOS 数据集上的编码结果。

方法	编解码器	解码时间	压缩率／比特率				
			1/8	1/16	1/32	1/64	1/128
CRNet CLNet TransNet	定长编解码器	$\approx T_{\mathrm{AD}} \times Z$	0.8750	0.4375	0.2188	0.1094	0.0546
TCLNet ($c=1$)	熵编解码器	$T_{\mathrm{AD}} \times Z + T_{\mathrm{fm}}$	0.7732	0.3985	0.2000	0.1038	0.0534
TCLNet（$c=0.8$）		$T_{\mathrm{AD}} \times Z + T_{\mathrm{fm}} \times 0.8 + T_{\mathrm{lm}} \times 0.2$	0.7280	0.3765	0.1860	0.0976	0.0497
TCLNet（$c=0.6$）		$T_{\mathrm{AD}} \times Z + T_{\mathrm{fm}} \times 0.6 + T_{\mathrm{lm}} \times 0.4$	0.6820	0.3560	0.1733	0.0930	0.0471
TCLNet（$c=0.4$）		$T_{\mathrm{AD}} \times Z + T_{\mathrm{fm}} \times 0.4 + T_{\mathrm{lm}} \times 0.6$	0.6380	0.3360	0.1608	0.0885	0.0452
TCLNet（$c=0.2$）		$T_{\mathrm{AD}} \times Z + T_{\mathrm{fm}} \times 0.2 + T_{\mathrm{lm}} \times 0.8$	0.5940	0.3150	0.1485	0.0840	0.0432
TCLNet（$c=0$）		$T_{\mathrm{AD}} \times Z + T_{\mathrm{lm}}$	**0.5470**	**0.2950**	**0.1358**	**0.0799**	**0.0417**
	估计熵（下界，$c=0$）		0.5459	0.2939	0.1332	0.0790	0.0416
	压缩效率（$c=0$）		**99.79%**	**99.32%**	**98.09%**	**98.87%**	**99.76%**



(a) 7 位量化。 (b) 6 位量化。 (c) 5 位量化。

图7. Argos数据集上采用不同位数进行量化的RDC权衡曲线。 c 的值越高，表示解码复杂度越低。

各方法达到的比特率。解码时间以算术解码时间 T_{AD}、T_{fm} 和 T_{lm} 表示，分别代表每个词元的算术解码时间、FM推理时间和LM推理时间。从表中可以看出，在所有 CR 和 c 设置下，所提出的采用熵编码的 TCLNet 在比特率方面始终优于定长编码方法。当 c 接近 1 时，表示编码主要使用 FM，所提出的 TCLNet 仍能显著降低比特率，相较于定长编码方法具有明显优势。当接收端拥有充足的计算资源时，选择较小的 c 可以让 LM 更深度地参与概率建模，从而进一步提升压缩效率。特别是当 $c=0$ 时，所提出的 TCLNet 达到了最低比特率：在 CR 为 1/8 时，比特率为 0.5470，非常接近估计熵下界。这些结果表明，所提出的无损压缩器能够有效利用 CSI数据中的上下文依赖关系，实现优异的压缩性能；此外，所提出的复杂度可控方案能够灵活适应接收端不同的计算资源约束。

在图7中，我们进一步展示了 Argos数据集上 TCLNet 在不同量化设置下的率失真复杂度权衡曲线，其中 c 从 0 到 1 变化，以表示不同的解码复杂度。我们还提供了基线方法的性能，以及训练后的LM估算的理论压缩界，以供比较。从图中可以看出，在不同量化设置下，TCLNet 始终实现了良好的率失真复杂度权衡，并且优于 CRNet、CLNet 和 TransNet 等基线方法，且比特率至多相同。此外，TCLNet 的性能始终更接近理论熵界，尤其是在 c 较小时，这表明 LM 能有效利用 CSI 中的上下文依赖关系。另一项重要观察是，c 的选择应根据量化级别进行调整。具体而言，对于高比特量化，使用较小的 c 可带来显著的压缩增益，因为 LM 能有效建模细粒度的符号相关性。对于低比特量化，$c=0$ 与 $c=1$ 之间的性能差距会缩小，这表明较高的 c （较低的复杂度）已足够，且更适合用于降低解码开销。这些结果进一步验证了所提出的 TCLNet框架在实现良好率失真复杂度权衡方面的灵活性和有效性。

C. 仿真数据集上的结果

我们进一步在广泛使用的模拟COST2100数据集上评估所提出的TCLNet [7]，，输入CSI矩阵的维度设为 32×32。如表III所示，我们首先展示了在压缩率为1/4至1/64的不同设置下，TCLNet与基线方法相比的室内NMSE性能和计算复杂度。可以看出，尽管CLNet和CRNet等轻量级基于CNN的方法具有最低的复杂度，但其重建性能往往有限。在性能较强的模型中，与TransNet和CsiNet相比，所提出的TCLNet取得了具有竞争力的NMSE结果+。具体而言，在压缩率=1/4时，TCLNet的NMSE达到-29.29

表III COST2100数据集上的归一化均方误差（分贝）与计算复杂度（百万浮点运算次数）比较。

方法	指标	压缩率				
		1/4	1/8	1/16	1/32	1/64
CLNet [9]	NMSE 浮点运算次数	-29.16 4.05M	-15.60 3.01M	-11.15 2.48M	-8.95 2.22M	-6.34 2.09M
CRNet [8]	NMSE 浮点运算次数	-26.99 5.12M	-16.01 4.07M	-11.35 3.55M	-8.93 3.29M	-6.49 3.16M
CsiNet+ [12]	NMSE 浮点运算次数	-27.37 24.57M	-18.29 23.52M	-14.14 23.00M	**-10.43** 22.74M	– –
TransNet [10]	NMSE 浮点运算次数	-29.24 35.72M	**-22.84** 34.70M	**-14.96** 34.14M	-10.35 33.88M	-6.08 33.75M
TCLNet	NMSE 浮点运算次数	**-29.29** 31.47M	-21.67 30.42M	-14.77 29.89M	-10.19 29.64M	**-7.81** 29.50M

分贝，优于TransNet的-29.24分贝，并显著优于CsiNet+的-27.37分贝。在其他压缩率设置下也呈现出相同趋势。值得注意的是，在压缩率=1/64时，TCLNet取得了最佳NMSE，为-7.81分贝，表明其在高压缩场景下具有良好的鲁棒性。这些结果凸显了TCLNet架构在平衡重建精度与模型复杂度方面的有效性。

在图8和图9中，我们分别给出了TCLNet与基线方法在COST2100室内和室外数据集上的比特率与归一化均方误差权衡曲线。可以观察到，在低比特率下，TCLNet的性能优于CsiNet+和TransNet，并且显著优于CLNet和CRNet。随着比特率的提高，TCLNet的性能增益更加明显。这可归因于TCLNet增强的表征能力，以及所提出的熵编码器实现的有效无损压缩。此外，通过设置不同的 c，我们可以观察到，当使用语言模型（LM）解码更多符号时，TCLNet的性能与上界有多接近，例如取较低的c值。虽然较高的 c 值会降低复杂度，但所提出的TCLNet仍能保持有竞争力的性能。此外，在室外场景中也观察到类似趋势：在各种比特率下，TCLNet始终优于基线方法，并接近理论性能极限。这些结果证明了所提出的TCLNet架构的稳健性和有效性，以及其在不同场景下自适应平衡压缩效率与计算复杂度的能力。

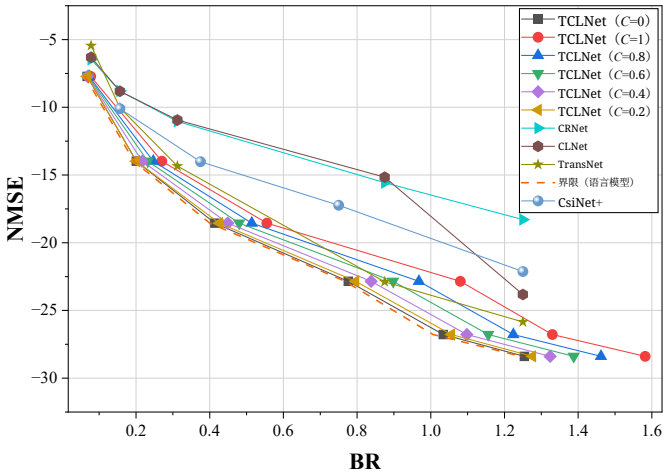


图8. COST2100室内数据集上的比特率-归一化均方误差权衡比较。TCLNet 的重建精度更高，尤其是在比特率较高时。

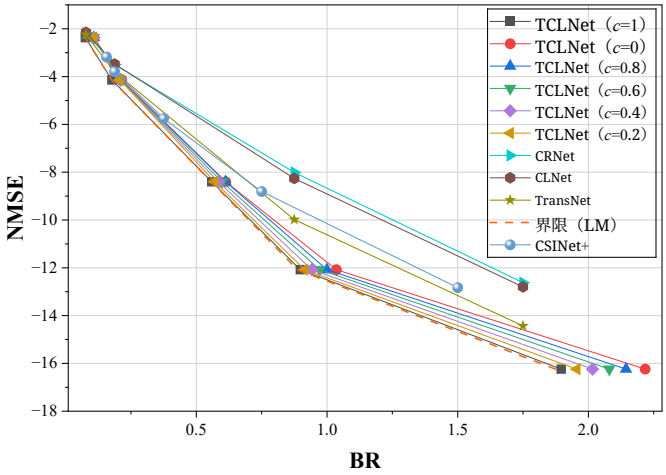


图9. COST2100室外数据集上的比特率-归一化均方误差权衡比较。即使在低比特率限制下，TCLNet 也表现出良好的鲁棒性。

表 IV 抗信道估计误差的鲁棒性

方法	浮点运算次数	归一化均方误差/分贝（完美信道状态信息）	归一化均方误差/分贝（含噪信道状态信息）
CRNet	19.579M	-10.44	-10.18
CLNet	15.016M	-9.159	-8.863
TransNet	800.996M	-13.52	-13.45
TCLNet	289.117M	**-14.27**	**-13.67**

D. 信道估计误差鲁棒性 s

在表 IV中，我们在 Argos数据集上评估了不同方法对下行信道估计误差的鲁棒性。用户侧估计的下行信道状态信息建模为

受到输入信噪比为 5 分贝的加性白高斯噪声（加性高斯白噪声）污染，且压缩率设为 1/64。从结果中可以看出，以含噪 CSI 作为输入时，所有方法的重建精度均有所下降，而所提出的 TCLNet 仍取得了所有竞争方法中最低的归一化均方误差，在性能上优于基于卷积神经网络和基于 Transformer 的基线方法。这些结果表明

TCLNet 对信道估计误差具有很强的鲁棒性。

E. 窗口大小对 *TCLNet* 的影响

在图10中，我们研究了所提出的TCLNet的Transformer模块中不同窗口大小设置对Argos数据集的影响，其中压缩率从1/16到1/128不等。可以观察到，窗口大小对TCLNet的重建性能有显著影响

表V 使用提示启用的大语言模型作为无损压缩器时的压缩性能比较。

模型	定长方案	FM	LM	ChatGPT-5	ChatGPT-4o
模型状态	–	训练所用CSI符号	使用.....训练CSI符号（ASCII）	预训练使用文本	预训练使用文本
方法	固定比特率	训练	训练	提示	提示
参数	–	≈ 20K	≈ 12.5M	＞ 1T	≈ 200B
比特率	0.0546	0.0534	0.0417	0.0513	0.0515
估计熵	0.0546	0.0520	0.0416	0.0498	0.0501

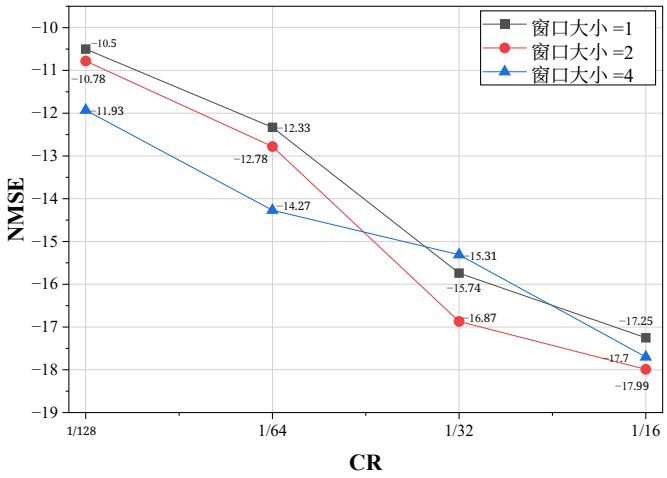


图10. 不同窗口大小下TCLNet的归一化均方误差比较，其中压缩率范围为1/16至1/128。

。具体而言，窗口大小为4时，在大多数压缩率下都能持续获得最佳归一化均方误差，因为它通过自注意力机制有效捕获了CSI矩阵中的空间依赖关系。然而，这也带来了过拟合的风险，如CR= 1/32时所见，此时窗口大小为4的表现略逊于较小的窗口大小。窗口大小减小到2时，模型仍能受益于自注意力机制，但效果有所减弱，因此性能中等。相反，窗口大小为1会将自注意力机制有效地简化为卷积操作，可能无法充分利用CSI数据中的空间相关性，因此在三种配置中表现最差。

F. 大语言模型作为*CSI*无损压缩器的潜力

在表V中，我们最终探讨了大语言模型作为CSI反馈无损压缩器的潜力。我们将两个预训练大语言模型 ChatGPT- 5 和 ChatGPT-4o 的压缩性能，与传统定长编码方案、基于FM的编码，以及使用 ASCII 分词在 CSI 符号上训练的 Transformer解码器进行比较。我们使用 Argos数据集进行评估，其中压缩率为 128 的有损压缩器会生成长度为 78 的符号序列。从结果中我们观察到，ChatGPT-5 和 ChatGPT-4o 的比特率分别为 0.0513 和 0.0515，优于定长编码方案和基于FM的编码，但略逊于在 CSI 符号上训练的 Transformer解码器。它们优于定长编码方案和基于FM的编码，但略逊于在 CSI 符号上训练的 Transformer解码器。这些结果表明，尽管通用大语言模型并未专门针对 CSI数据进行训练，但其强大的上下文建模能力已使其具备出色的压缩性能，展现出作为未来 CSI反馈系统通用无损压缩器的巨大潜力。

V. 结论

本文提出了一种新颖的CSI反馈网络TCLNet。借助混合Transformer-CNN架构，与以往基于Transformer的模型相比，TCLNet在实现高效CSI压缩的同时，显著降低了计算复杂度。此外，我们将基于语言模型和频率模型的无损编码方案整合到TCLNet中，使其在比特级CSI压缩方面实现了更优的率失真复杂度权衡。我们还探讨了大语言模型作为CSI反馈通用无损压缩器的潜力。我们在多个数据集上进行了大量实验，证明所提出的TCLNet优于现有的CSI反馈数据驱动方法和定长编码方法。结果还凸显了大语言模型在这一新兴应用领域中的巨大潜力。未来的工作可从多个方向着手，进一步提升系统性能。首先，可采用LoRA等参数高效微调技术，使大语言模型专用于CSI概率建模，同时避免显著增加训练开销。其次，可探索交叉注意力等先进注意力机制，以进一步改善对CSI数据中空间依赖关系的建模。

参考文献

[1] Z. Yang 和 Q. Yang，“TCNet：利用语言模型作为无损压缩器的 CSI反馈压缩统一框架”，载于机器学习无线通信与网络研讨会论文集（*ML4Wireless*），2025 年。[2] Y. Dong、H. Wang 和 Y.-D. Yao，“利用条件生成对抗网络进行单比特多用户大规模MIMO信道估计”，*IEEE*通信快报，第 25 卷，第 3 期，第 854–858 页，2020 年。[3] R. Shafin 和 L. Liu，“多小区多用户大规模全维MIMO：下行预编码与吞吐量分析”，*IEEE*无线通信汇刊，第 18 卷，第 1 期，第 487–502 页，2018 年。[4] P. W. Chan、E. S. Lo、R. R. Wang、E. K. Au、V. K. Lau、R. S. Cheng、W. H. Mow、R. D. Murch 和 K. B. Letaief，“4G网络的发展路径：频分双工还是时分双工？”，*IEEE*通信杂志，第 44 卷，第 12 期，第 42–50 页，2006 年。

[5] Z. Gao、L. Dai、S. Han、C.-L. I、Z. Wang 和 L. Hanzo，“面向下一代无线通信的压缩感知技术”，IEEE Wireless Commun. Mag.，第 25 卷，第 3 期，第 144–153 页，2018 年。[6] P.-H. Kuo、H. T. Kung 和 P.-A. Ting，“面向空间相关大规模天线阵列的压缩感知信道反馈协议”，载于 IEEE Wireless Commun. Netw. Conf. (WCNC) 论文集，2012 年，第 492–497 页。[7] C.-K. Wen、W.-T. Shih 和 S. Jin，“用于大规模MIMO CSI反馈的深度学习”，IEEE Wireless Commun. Lett.，第 7 卷，第 5 期，第 748–751 页，2018 年。[8] Z. Lu、J. Wang 和 J. Song，“大规模MIMO系统中基于深度学习的多分辨率CSI反馈”，载于 IEEE Int. Conf. Commun. (ICC)，2020 年，第 1–6 页。[9] S. Ji 和 M. Li，“CLNet：面向大规模MIMO CSI反馈的复数输入轻量级神经网络”，IEEE Wireless Commun. Lett.，第 10 卷，第 10 期，第 2318–2322 页，2021 年。[10] Y. Cui、A. Guo 和 C. Song，“TransNet：用于频分双工大规模MIMO系统CSI反馈的全注意力网络”，IEEE Wireless Commun. Lett.，第 11 卷，第 5 期，第 903–907 页，2022 年。[11] J. Guo、C.-K. Wen、M. Chen 和 S. Jin，“利用人工智能增强环境知识辅助的大规模MIMO反馈码本”，IEEE Trans. Commun.，第 70 卷，第 7 期，第 4527–4542 页，2022 年。[12] J. Guo、C.-K. Wen、S. Jin 和 G. Y. Li，“基于卷积神经网络的多速率压缩感知大规模MIMO CSI反馈：设计、仿真与分析”，IEEE Trans. Wirel. Commun.，第 19 卷，第 4 期，第 2827–2840 页，2020 年。[13] W. Chen、W. Wan、S. Wang、P. Sun、G. Y. Li 和 B. Ai，“CSI-PPPNet：面向大规模MIMO CSI反馈的单侧通用深度学习框架”，IEEE Trans. Wirel. Commun.，第 23 卷，第 7 期，第 7599–7611 页，2024 年。[14] Z. Qin、J. Fan、Y. Liu、Y. Gao 和 G. Y. Li，“无线通信中的稀疏表示：一种压缩感知方法”，IEEE Commun. Mag.，第 35 卷，第 3 期，第 40–58 页，2018 年。[15] B. Makki 和 T. Eriksson，“准静态衰落信道中的混合自动重传请求与量化CSI反馈方案”，IEEE Trans. Commun.，第 60 卷，第 4 期，第 986–997 页，2012 年。[16] X. Rao 和 V. K. N. Lau，“频分双工多用户大规模MIMO系统中的分布式压缩感知 CSIT估计与反馈”，IEEE Trans. Signal Processing，第 62 卷，第 12 期，第 3261–3271 页，2014 年。[17] C. K. Au-Yeung、D. J. Love 和 S. Sanayei，“网格编码线打包：大维度波束成形向量量化与反馈传输”，IEEE Trans. Wirel. Commun.，第 10 卷，第 6 期，第 1844–1853 页，2011 年。[18] W. Yu、Y. Shen、H. He、X. Yu、S. Song、J. Zhang 和 K. B. Letaief，“面向太赫兹超大规模MIMO信道估计的自适应鲁棒深度学习框架”，IEEE J. Sel. Topics Signal Process.，第 17 卷，第 4 期，第 761–776 页，2023 年。[19] Y. Wang、X. Chen、H. Yin 和 W. Wang，“基于可学习稀疏变换的大规模MIMO CSI恢复网络”，IEEE Commun. Lett.，第 24 卷，第 7 期，第 1468–1471 页，2020 年。[20] Z. Hu、G. Liu、Q. Xie、J. Xue、D. Meng 和 D. G¨und¨uz，“面向大规模MIMO CSI反馈的可学习优化与正则化方法”，IEEE Trans. Wirel. Commun.，第 23 卷，第 1 期，第 104–116 页，2024 年。[21] J. Guo、L. Wang、F. Li 和 J. Xue，“基于大规模MIMO系统模型驱动深度学习的CSI反馈”，IEEE Commun. Lett.，第 26 卷，第 3 期，第 547–551 页，2021 年。[22] Y. Ma、H. He、S. Song、J. Zhang 和 K. B. Letaief，“通过学习优化实现低复杂度CSI反馈的频分双工大规模MIMO系统”，IEEE Trans. Wirel. Commun.，第 24 卷，第 4 期，第 3483–3498 页，2025 年。[23] S. Tang、J. Xia、L. Fan、X. Lei、W. Xu 和 A. Nallanathan，“基于膨胀卷积的大规模MIMO系统CSI反馈压缩”，IEEE Trans. Veh. Technol.，第 71 卷，第 10 期，第 11216–11221 页，2022 年。[24] Y. Wang、Y. Zhang、J. Sun、G. Gui、T. Ohtsuki 和 F. Adachi，“一种基于深度学习的频分双工大规模MIMO系统新型压缩CSI反馈方法”，载于 IEEE Wirel. Commun. Netw. Conf. (WCNC)，2021 年，第 1–5 页。[25] X. Liang、H. Chang、H. Li、X. Gu 和 L. Zhang，“基于深度学习的CSI反馈可变速率与新型量化方法”，IEEE Trans. Wirel. Commun.，第 21 卷，第 12 期，第 10100–10114 页，2022 年。[26] M. Chen、J. Guo、C.-K. Wen、S. Jin、G. Y. Li 和 A. Yang，“基于深度学习的大规模MIMO隐式CSI反馈”，IEEE Trans. Commun.

，第70卷，第2期，第935–950页，2021年。

[27] W. Liu、W. Tian、H. Xiao、S. Jin、X. Liu 和 J. Shen，《Evcsinet：3GPP链路级信道下基于特征向量的CSI反馈》，IEEE Wireless Commun. Lett.，第10卷，第12期，第2688–2692页，2021年。[28] S. Jo 和 J. So，《面向大规模MIMO系统的自适应轻量级基于CNN的CSI反馈》，IEEE Wireless Commun. Lett.，第10卷，第12期，第2776–2780页，2021年。[29] H. Ye、F. Gao、J. Qian、H. Wang 和 G. Y. Li，《面向频分双工大规模MIMO系统CSI反馈的基于深度学习的去噪网络》，IEEE Commun. Lett.，第24卷，第8期，第1742–1746页，2020年。[30] Q. Cai、C. Dong 和 K. Niu，《面向大规模MIMO CSI压缩反馈与恢复的注意力模型》，发表于 IEEE Wireless Communications and Networking Conference（WCNC），2019年，第1–5页。[31] P. Liang、J. Fan、W. Shen、Z. Qin 和 G. Y. Li，《面向频分双工大规模MIMO系统的基于深度学习和压缩感知的CSI反馈》，IEEE Trans. Veh. Technol.，第69卷，9222，2020年8，9217 no. pp. – 。[32] Y. Xu、M. Zhao、S. Zhang 和 H. Jin，《DFECsiNet：利用多样化信道特征实现大规模MIMO CSI反馈》，发表于 IEEE International Conference on Wireless Communications and Signal Processing（WCSP），2021年，第1–5页。[33] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polosukhin，《注意力机制就是你所需要的一切》，发表于 Advances in Neural Information Processing Systems（NeurIPS），第30卷，2017年。[34] R. Zhao、Z. Liu、T. Song、J. Jin、G. Jin 和 L. Fan，《用于高效CSI反馈的混合CNN-Transformer网络》，Physical Commun.，第66卷，第102477页，2024年。[35] J. Liu、H. Sun 和 J. Katto，《采用混合Transformer-CNN架构的学习式图像压缩》，发表于 IEEE/CVF Conference on Computer Vision and Pattern Recognition（CVPR），2023年，第14388–14 397页。[36] J. Shin、Y. Kang 和 Y.-S. Jeon，《面向大规模MIMO系统中基于深度学习的CSI反馈的向量量化》，IEEE Wireless Commun. Lett.，第13卷，第9期，第2382–2386页，2024年。[37] J. Shin、J. Park 和 Y.-S. Jeon，《面向基于深度学习的CSI反馈的熵约束VQ-VAE》，IEEE Trans. Veh. Technol.，第74卷，第6期，第9870–9875页，2025年。[38] M. B. Mashhadi、Q. Yang 和 D. G¨und¨uz，《面向大规模MIMO CSI反馈的分布式深度卷积压缩》，IEEE Trans. Wirel. Commun.，第20卷，第4期，第2621–2633页，2021年。[39] G. Del´etang、A. Ruoss、P. Duquenne、E. Catt、T. Genewein、C. Mattern、J. Grau-Moya、L. K. Wenliang、M. Aitchison、L. Orseau、M. Hutter 和 J. Veness，《语言建模即压缩》，发表于 International Conference on Learning Representations（ICLR），2025年。[40] J. Du、C. Zhou、N. Cao、G. Chen、Y. Chen、Z. Cheng、L. Song、G. Lu 和 W. Zhang，《利用视觉提示进行无损图像压缩的大语言模型》，arXiv:2502.16163，2025年。[41] B. Liu、X. Liu、S. Gao、X. Cheng 和 L. Yang，《LLM4CP：面向信道预测的大语言模型适配》，J. Commun. Inf. Netw.，第9卷，第2期，第113–125页，2024年。[42] Z. Li、Q. Yang、Z. Xiong、Z. Shi 和 T. Q. Quek，《弥合模态鸿沟：通过语义对齐的大语言模型和知识蒸馏增强信道预测》，arXiv:2505.12729，2025年。[43] K. He、L. He、L. Fan、X. Lei、T. X. Vu、G. K. Karagiannidis 和 S. Chatzinotas，《SCA-LLM：基于频谱注意力大语言模型的无线世界建模，面向智能体通信》，arXiv:2509.08139，2025年。[44] X. Liu、S. Gao、B. Liu、X. Cheng 和 L. Yang，《LLM4WM：将大语言模型适配于无线多任务处理》，IEEE Trans. Machine Learn. Commun.，第3卷，第835–847页，2025年。[45] T. Yang、P. Zhang、M. Zheng、Y. Shi、L. Jing、J. Huang 和 N. Li，《WirelessGPT：面向无线通信的生成式预训练多任务学习框架》，IEEE Netw.，第39卷，65，2025 5，58no. pp. – 。[46] D. Minnen、J. Ball´e 和 G. D. Toderici，《学习式图像压缩的联合自回归与分层先验》，发表于 Advances in Neural Information Processing Systems（NeurIPS），第31卷，2018年。[47] J. Ball´e、D. Minnen、S. Singh、S. J. Hwang 和 N. Johnston，《带尺度超先验的变分图像压缩》，发表于 International Conference on Learning Representations（ICLR），2018年。[48] K. He、X. Zhang、S. Ren 和 J. Sun，《用于图像识别的深度残差学习》，发表于 IEEE/CVF Conference on Computer Vision and Pattern Recognition（CVPR），2016年，第770–778页。[49] I. H. Witten、R. M. Neal 和 J. G. Cleary，《用于数据压缩的算术编码》，Commun. ACM，第30卷，第6期，第520–540页，1987年。[50] C. Shepard、J. Ding、R. E. Guerra 和 L. Zhong，《理解真实的多天线MU-MIMO信道》，发表于 Asilomar Conference on Signals, Systems, and Computers（ACSSC），2016年，第461–467页。