

分类号： TN915

单位代码： 10335

密 级： 公开

申 请 号： 11831035

浙江大学

博士学位论文



中文论文题目： 基于端到端模型驱动深度神经网络
的物理层收发机设计

英文论文题目： **End-to-End Model-Driven Deep Neural Network
for Physical Layer Transceiver Design**

申请人姓名： 胡棋昱

指导教师： 蔡云龙

合作导师： 余官定、刘安

专业名称： 信息与通信工程

研究方向： 无线通信与信号处理

所在学院： 信息与电子工程学院

论文提交日期 2023年6月

基于端到端模型驱动深度神经网络
的物理层收发机设计



论文作者签名: 胡棋昱

指导教师签名: [Handwritten Signature]

论文评阅人1: 匿名

评阅人2: 匿名

评阅人3: 匿名

评阅人4: 匿名

评阅人5: 匿名

答辩委员会主席: 李春光 教授 浙江大学信电学院

委员1: 范建存 教授 西安交通大学

委员2: 沈雷 教授 杭州电子科技大学

委员3: 韩劲松 教授 浙江大学计算机学院

委员4: 李旻 研究员 浙江大学信电学院

答辩日期: 2023.6.6

End-to-End Model-Driven Deep Neural Network
for Physical Layer Transceiver Design



Author's signature: Qiyu Hu

Supervisor's signature: Cao Zhenyong

External Reviewers: Anonymous
Anonymous
Anonymous
Anonymous
Anonymous

Examining Committee Chairperson:
Chunguang Li

Examining Committee Members:
Jiancun Fan
Lei Shen
Jinsong Han
Min Li

Date of oral defence: 2023.6.6

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

胡棋昱

签字日期： 2023 年 6 月 9 日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权浙江大学可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：

胡棋昱

导师签名：



签字日期： 2023 年 6 月 9 日

签字日期： 2023 年 6 月 9 日

致 谢

鲜衣怒马少年时,不负韶华行且知。

岁月青葱，白驹过隙；春光易逝，且行且歌。从初入求是园的欣喜激动，到即将离去的恋恋不舍，九年的浙大生涯即将画上句点。烟花三月的江南，氤氲着雨后清新的泥土气息，宁静庄重的玉泉偶听得夹杂着欢声笑语的激烈学术讨论。蓦然回首间，已翻山越岭走过漫漫征途，一个个坚实的脚步，一串串悠扬的欢歌，一滴滴辛勤的汗水，凝聚成点点珍贵的青春记忆。我曾将人生最美好的青春年华挥洒在求是园，在这片生机勃勃的热土上播撒梦想和希望的种子，谱写属于我的华丽乐章。九载春秋，见证了春天紫金港的百花齐放朝气蓬勃，夏天玉泉的百年古木郁郁葱葱，秋天之江的红叶似火暮鼓晨钟，冬天华家池的大雪寒冰万籁俱寂。曾几何时，在考试周的挑战夜战中感受凌晨安静无人的校园，在项目出差的披星戴月中聆听清晨婉转动听的鸟鸣；为攻克科研难题绞尽脑汁，为学生工作的顺利开展来回奔波，一幅幅满载回忆的画面构成了丰富多彩校园生活。

在读博求学的道路上，我曾迷茫无助，也曾骄傲自负，但更多的是奋力拼搏、风雨兼程，成长的历程或许可以总结为经历了“科研的三重境界”。从彷徨迷惘到“昨夜西风凋碧树，独上高楼，望尽天涯路”，于上下求索中锚定人生的航向；从懈怠踟蹰到“为伊消得人憔悴，衣带渐宽终不悔”，于锲而不舍中重拾前行的动力；从焦虑急躁到“蓦然回首，那人却在，灯火阑珊处”，于静待花开中收获累累的硕果。一路走来，行过千山万水，道尽千言万语。感谢恩师、同窗、家人、朋友，以无微不至的关怀助我直面挑战一路坚持，以朝夕相处的陪伴使我的求学生涯绚丽多姿。

首先由衷地感谢我的导师蔡云龙教授。蔡老师在科研上对我悉心指导，对论文的修改严谨细致，让我感受到了学术研究的魅力所在。蔡老师有着一丝不苟的治学态度、极致认真的做事风格和学富五车的专业知识，以及对于学术锲而不舍的热爱和追求，是我学习的榜样。当我在科研中遇到困难而止步不前时，正是蔡老师及时的鞭策让我顺利走出泥沼。在引导并帮助我的同时，给予我充分的自由，面对我的某些选择给予了莫大的理解和包容，谨此表达对蔡老师的谢意。其次要感谢合作导师余官定教授。余老师学术水平高超，教导我要坚持做基础的、实际的高质量研究，能够切实解决通信系统中的实际问题。在我

遇到科研上的挫折以及面临人生选择时，春风化雨般以丰富的阅历指引正确的航向，并给予莫大的信任和支持。感谢合作导师刘安研究员，在学术上给予的指导提升了对科研的兴趣，加深了对实际系统的理解，在我迷茫时引导走向正确的科研航向。感谢赵民建教授。赵老师谈吐优雅、学识渊博，有着“谦谦君子，温润如玉”的学者气质。在课程及组会上传授的知识，以及对人生方向的选择和指点，让我收获良多。感谢合作导师史清江教授对我学术方面的指点，以及实验室的李旻、雷鸣、李立言研究员和赵明敏教授，在我遇到困难时曾给予无私恳切的帮助。感谢邵红霞、王婵和杨丽萍老师，为保障实验室正常运转、营造和谐的学习工作环境所付出的努力。祝愿各位老师科研顺利、事业有成！

还要感谢在英国留学期间的合作导师秦志金教授和李焯教授，在一年的访学过程中给予我悉心的指导和学术上的启发。钦佩老师们高深的学术造诣、饱满的科研热情和永无止境的探索精神，丰富了我的学术视野。同时，感谢在伦敦共同求学的谢汇强、严蕾、冀泽霖、翁振紫、申文晗、蔡卫彤、林吉等小伙伴给予的帮助和陪伴，为我的访学生活增添了亮丽的色彩。

感谢崔方宇、陈晞涵、赵存茁、刘瑞、王雅霖、刘胜利、王韵佳等师兄师姐的关心支持，你们是我在科研上的引路人。感谢同级的李梦圆、何映晖、盛华联、刘耀、许小可、刘派、李可欣、丘耿鑫、刘诗雨、陈成霞、寇遵莉等小伙伴的携手并肩，陪伴我经历了读博的酸甜苦辣，相互学习，共赴山海。感谢黄哲、陈梓健、丁光耀、刘文亮、黄冠、窦为龙、黄钰鹏、章晓豪、高惠国、万雨博、胡智祥、段伟骏、刘屹豪、陆旻、郑宇涛、张勉一、王珂璇、徐文康、魏涵宇、盛思锴、闻陈祎鸣、丁悦晋、蔡文凯、林柏烁、周珂全、梁飒钰等师弟师妹的陪伴照顾，让我博士生涯的后半段充满温暖与欢乐。特别感谢实验室相同研究方向的课题组成员徐凯迪、刘彦桢、施淑涵、康凯、张光义，每周组会的热烈讨论和思想碰撞，以及对本论文提出的宝贵意见，共同挑战科研难题、勇攀学术高峰的经历将是我一生宝贵的财富。感谢五年来朝夕相处的室友乔程昱、蒲削锬、沈锡聪，曾在读博最艰难的阶段相互鼓励，互诉衷肠。感谢一直以来相互陪伴的梅红艳、朱元军、王兴路、吕呈辉、董峻廷、楼黎明、陈逗、李媛媛、方康恒、张宁珊、徐安迪等好伙伴，已伴我走过十年的风雨兼程。还有一路走来所有相遇相识、或曾给予我关心和帮助的其他伙伴同学，一一罗列不得，感激不尽。祝愿小伙伴们前程似锦，皆得所愿！

感谢一直以来在背后默默付出、支持鼓励我的余晓群、胡勇、朱云华、胡维善、余常富，和已去世的袁贞芳等家人长辈。从呱呱坠地，蹒跚学步，到二十一载栉风沐雨的异地求学，直至博士即将毕业，每一个坚实的脚印背后是你们毫无保留的付出和薪火相传的守望，在我迷茫时给予家的温暖和呵护，在我困顿时给予及时的勉励和帮助。衷心祝愿你们身体健康，幸福快乐！

最后，要感谢坚持一路前行的自己，虽曾深陷泥淖，但不惧风雨，跌跌撞撞走了出来。曾在高考完的那一天给未来的自己写过一封信，擘画了对自己未来的期许和蓝图。倘若能和过去的自己对话，不知现在的我是否成为了九年前的我想成为的那个人。攻读博士学位的过程，除了勇攀学术高峰、勇闯无人区，更是一种向内自省的过程，克服内心的懒惰、自傲、恐惧、焦虑，纵使身处困顿依旧一心向阳。“风物长宜放眼量”，不执著于“一城一池”的得失，目光长远；“梅花香自苦寒来”，在艰难困苦中磨砺自我，知行合一；“不失初来赤子心”，心怀“国之大者”投身当下的学习生活，刻苦钻研。破除对“小我”的执著，在奉献、钻研、利他的过程中感受自我的价值和意义。渐渐地，在面对竺老校长之问：到浙大来做什么、将来毕业后做什么样的人，拨云见日，有了前行的动力和方向。愿面对未来人生的选择，坚守“何妨吟啸且徐行”的淡然，不问得失，无悔对错；面对人生的道阻且长，怀揣“竹杖芒鞋轻胜马”的自信，披荆斩棘，一往无前。长风破浪会有时，直挂云帆济沧海。行文至此，终将告别，这里的一切有始有终，写满了一年又一年的不期而遇和久别重逢。愿千帆阅尽，归来仍是少年；愿万里归途，总会有风。

胡棋昱

2023年4月 于求是园

摘要

第六代移动通信（The 6th Generation, 6G）系统对性能的严格要求，如超高数据传输速率、极高的可靠性和低延迟，激发了对具有先进收发机技术的下一代多用户多输入多输出（Multi-User Multiple-Input Multiple-Output, MU-MIMO）系统的研究，并被广泛应用于自动驾驶、增强现实等场景。在6G系统中，面向下一代MIMO系统的先进物理层收发机设计是一项核心技术，包括发送端和接收端的信号处理架构、混合预编码、波束选择、信道估计、信道反馈和语义通信等方面的创新设计。为了满足6G的要求，采用包含领域知识的迭代优化算法对收发机进行了设计。虽然可实现较好的系统性能，但通常计算复杂度较高，难以在实际系统中部署。近年来，随着人工智能（Artificial Intelligence, AI）的发展，深度学习因其强大的学习能力和处理大数据的优势，为6G系统的收发机设计提供了潜在解决方案。为提高其可解释性和泛化能力，研究者提出一种新颖的模型驱动网络，将包含领域知识的优化算法与深度神经网络（Deep Neural Networks, DNNs）相结合。本文旨在针对物理层收发机的不同模块设计高效的模型驱动网络，并将其融入端到端深度学习中，更好地解决通信信号处理相关的问题。

首先，本文根据物理层收发机的特点，提出了矩阵形式的模型驱动网络框架。随后将该框架用于解决MU-MIMO系统中预编码设计的和速率最大化问题。基于经典加权最小均方误差（Weighted Minimum Mean-Square Error, WMMSE）迭代算法，设计了高效模型驱动网络架构。具体而言，迭代WMMSE算法被展开为类似于DNN的逐层结构，并引入了一系列训练参数来取代高复杂度操作。为了更高效地训练网络，提出了广义链式法则来描述反向传播过程中相邻两层间梯度的递推关系。仿真结果表明，所提出的网络能以更低的计算复杂度，逼近迭代WMMSE算法的性能。

其次，本文设计了基于深度确定策略梯度（Deep Deterministic Policy Gradient, DDPG）的模型驱动网络框架，其针对不同输入样本可自适应调节网络层数。将问题的优化变量、网络训练参数和网络层结构分别设计为DDPG的状态、动作和状态转移。随后，将该框架应用于求解大规模MU-MIMO系统的稀疏信道估计问题。具体而言，将基于稀疏贝叶斯学习的迭代算法展开为逐层网络结构，并采用基于DDPG的模型驱动框架设计可变层信道估

计网络。为实现可变层网络，设计了停止分数来指示网络何时停止。仿真结果表明，该算法在层数显著减少的情况下，实现的性能优于迭代优化算法和固定层数网络。

再次，本文研究了包含离散透镜阵列的MU-MIMO系统中的波束选择和数字预编码矩阵联合设计问题。针对具有离散变量和耦合约束的非凸优化问题，提出了一种高效的联合网络框架。具体而言，该框架由基于深度强化学习（Deep Reinforcement Learning, DRL）的网络和模型驱动网络组成，分别用于优化波束选择矩阵和数字预编码矩阵。对于基于DRL的网络，将波束选择问题建模为马尔可夫决策过程，并设计双深度Q网络进行求解。针对数字预编码矩阵设计，提出了一种基于迭代WMMSE算法的模型驱动网络，将该算法展开为包含训练参数的逐层网络结构。仿真结果表明，该联合训练网络具有较低的计算复杂度和较强的鲁棒性，性能优于现有迭代算法。

此外，本文提出了用于大规模MIMO系统的基于端到端深度学习的联合物理层收发机设计，该网络架构由基于DNN的导频训练信道估计、信道反馈和混合数模混合预编码等模块组成。为了减少信令开销和由于传输时延导致的信道失配，根据该架构进一步设计了双时间尺度网络，其由长时间尺度网络和短时间尺度网络组成。具体而言，长时间尺度网络估计并反馈高维原始信道，并在每帧更新一次模拟预编码矩阵。短时间尺度网络估计并反馈低维等效信道，并在每个时隙更新一次数字预编码矩阵。仿真结果表明，该方案在显著降低信道估计和反馈开销，同时缩短导频序列的情况下，可提升系统误码率性能。

最后，本文设计了一种用于图像传输的鲁棒端到端语义通信系统框架，以对抗语义噪声，提升系统鲁棒性。具体而言，建模并分析了样本依赖和非样本依赖的两种常见语义噪声。为了对抗该语义噪声，提出了基于权值扰动对抗训练方案，将含有语义噪声的样本纳入训练数据集。随后，设计噪声相关的掩蔽策略，并用于设计掩码矢量量化变分自编码器。使用由收发端共享的离散码本来表示经过编码的图像特征，并设计了用于高效传输的星座图。为进一步提高系统鲁棒性，设计了特征重要性模块来抑制噪声相关以及和任务无关的图像特征。仿真结果表明，该方法可应用于许多下游任务，显著提高了系统对语义噪声的鲁棒性，并降低了传输开销。

本文的前两项研究成果针对物理层收发机模块设计了性能较好且计算复杂度较低的模型驱动网络框架，后三项研究成果将模型驱动网络融入端到端深度学习框架中，为通信系统从5G向6G的演进提供了理论基础和技术解决方案。

关键词：物理层收发机技术、端到端深度学习、模型驱动深度神经网络、大规模MU-MIMO系统、混合预编码、波束选择、信道估计和反馈、鲁棒语义通信

Abstract

The stringent performance requirements of 6th generation (6G) systems, such as ultra-high data rates, extremely high reliability and low latency, are spurring worldwide studies on defining the next-generation multi-user multiple-input multiple-output (MU-MIMO) systems with advanced transceivers that can integrate far-reaching applications ranging from autonomous systems to extended reality. In 6G systems, advanced physical layer transceiver design for next-generation MIMO systems is a core technology that comprises innovative designs in transmit and receive processing architecture, hybrid precoding, beam selection, channel estimation, channel feedback, and semantic communications, etc. To satisfy the requirements of 6G, iterative optimization algorithms with domain knowledge are an attempt to design the transceiver. Although these algorithms have exhibited satisfactory system performance, they typically have high computational complexity and make the deployment difficult. Recently, with the development of artificial intelligence (AI), deep learning techniques provide a solution due to the strong learning ability and the superiority of dealing with big data. To improve its interpretability and generalization ability, researches further propose a novel deep learning technique, named model-driven deep-unfolding network. It fully exploits the domain knowledge and combines the optimization algorithms with deep neural networks (DNNs). This thesis aims to propose efficient model-driven networks for different modules in physical layer transceiver, and incorporates them into end-to-end deep learning to effectively solve the problems in communication signal processing.

First, based on the characteristics of physical layer transceiver design, this thesis proposes a framework for model-driven networks in matrix form. Then, we implement the proposed framework to solve the sum-rate maximization problem for precoding design in MU-MIMO systems. An efficient network architecture is developed based on the classic weighted minimum mean-square error (WMMSE) algorithm. Specifically, the iterative WMMSE algorithm is unfolded into a layer-wise structure, where a number of trainable parameters are introduced to replace the high-complexity operations. To train the network, a generalized chain rule is proposed to depict the

recurrence relation of gradients between two adjacent layers in the back propagation. Simulation results show that the proposed IAIDNN efficiently achieves the performance of the iterative WMMSE algorithm with reduced computational complexity.

Second, this thesis develops a framework of deep deterministic policy gradient (DDPG)-based model-driven network with adaptive depth for different inputs. Specifically, the optimization variables, trainable parameters, and architecture of deep-unfolding NN are designed as the state, action, and state transition of DDPG, respectively. Then, this framework is employed to deal with the channel estimation problem in massive MU-MIMO systems. Specifically, we unfold the sparse Bayesian learning (SBL)-based algorithm into a layer-wise structure and employ the proposed DDPG-based framework to solve this channel estimation problem. To realize adaptive depth, we design the halting score to indicate when to stop. Simulation results show that the proposed algorithm outperforms the conventional optimization algorithms and DNNs with fixed depth with much reduced number of layers.

Next, this thesis investigates the joint design of beam selection and digital precoding matrices for MU-MIMO systems with discrete lens array (DLA). The investigated non-convex problem with discrete variables and coupled constraints is challenging to solve and an efficient framework of joint neural network (NN) design is proposed to tackle it. Specifically, the proposed framework consists of a deep reinforcement learning (DRL)-based NN and a deep-unfolding NN, which are employed to optimize the beam selection and digital precoding matrices, respectively. As for the DRL-based NN, we formulate the beam selection problem as a Markov decision process (MDP) and a double deep Q-network (DDQN) is developed to solve it. Regarding the design of the digital precoding matrix, we develop an iterative WMMSE algorithm induced model-driven NN, which unfolds this algorithm into a layer-wise structure with introduced trainable parameters. Simulation results verify that this jointly trained NN remarkably outperforms the existing iterative algorithms with reduced complexity and stronger robustness.

Furthermore, this thesis proposes an end-to-end deep learning-based joint transceiver design algorithm for massive MIMO systems, which consists of DNN-aided pilot training, channel feedback, and hybrid analog-digital precoding. To further reduce the signaling overhead and channel state information (CSI) mismatch caused by the transmission delay, a two-timescale DNN composed of a long-term DNN and a short-term DNN is developed. In particular, the analog precoders are designed by the long-term DNN based on the estimated high-dimensional full CSI matrices and updated once in a frame. In contrast, the digital precoders are optimized by the short-term

DNN based on the estimated low-dimensional equivalent CSI matrices and updated once in a time slot. Simulation results show that our proposed technique significantly outperforms conventional schemes in terms of bit-error rate performance with reduced signaling overhead and shorter pilot sequences.

Finally, this thesis designs a framework for the robust end-to-end semantic communication systems for image transmission to combat the semantic noise. In particular, we analyze sample-dependent and sample-independent semantic noise. To combat the semantic noise, the adversarial training with weight perturbation is developed to incorporate the samples with semantic noise in the training dataset. Then, we design the masked vector quantized-variational autoencoder (VQ-VAE) with the noise-related masking strategy. We use a discrete codebook shared by the transmitter and the receiver for encoded feature representation, where a constellation has been designed for efficient transmission. To further improve the system robustness, we develop a feature importance module to suppress the noise-related and task-unrelated features. Simulation results show that the proposed method can be applied in many downstream tasks and significantly improve the robustness against semantic noise with remarkable reduction on the transmission overhead.

The first two works propose framework of model-driven network for transceiver modules in physical layer communication with satisfactory performance and low computational complexity, while the latter three works incorporate model-driven networks to end-to-end learning framework. They jointly provide some theoretical basis and effective technical solutions for the further evolution of 5G to the 6G.

Keywords: Physical layer transceiver technology, end-to-end deep learning, model-driven deep neural network, massive MU-MIMO systems, hybrid precoding, beam selection, channel estimation and feedback, robust semantic communications.

插图

1.1	6G应用场景	2
1.2	6G系统的部署预计时间表	3
1.3	6G无线通信网络概述	4
1.4	FNN结构示意图	7
1.5	CNN结构示意图	7
1.6	VAE结构示意图	8
1.7	ViT网络	9
1.8	DRL示意图	10
1.9	迭代优化算法和模型驱动网络示意图	11
1.10	深度神经网络在无线通信场景中的应用	16
1.11	端到端物理层收发机	24
1.12	端到端语义通信系统	24
1.13	论文框架示意图	26
2.1	迭代WMMSE算法的流程框图	38
2.2	基于迭代WMMSE算法的模型驱动网络结构。	40
2.3	模型驱动网络的离线训练阶段和在线预测阶段示意图	41
2.4	黑盒网络结构示意图	43
2.5	不同训练批次大小和学习率下系统($N_t = 128, K = 30$)的收敛性能	47
2.6	系统 $N_t = 64$ 的测试数据集中不同样本的和速率性能累积分布函数	47
2.7	系统 $N_t = 128$ 的测试数据集中不同样本的和速率性能分布	48
2.8	不同层数IAIDNN的和速率性能	50
2.9	系统($N_t = 64, K = 20$)非完美信道下的和速率性能	50
3.1	优化算法和网络对比: (a) 迭代次数随样本可变的优化算法; (b) 层数固定的模型驱动/黑盒网络	54

3.2	DDPG网络架构.....	56
3.3	可变层网络框架：(a)可变层DNN框架；(b)基于DDPG的可变层模型驱动网络框架。.....	57
3.4	基于SBL算法的用于信道估计的模型驱动网络结构.....	64
3.5	基于DDPG的用于展开迭代SBL算法的可变层模型驱动网络结构.....	66
3.6	不同学习率下网络的收敛性.....	71
3.7	NMSE性能随SNR的变化.....	72
3.8	NMSE性能随网格点数目 \hat{J} 的变化.....	72
3.9	NMSE性能随导频长度 T 的变化.....	73
3.10	不同信道路径数和SNR下模型驱动网络的最优平均层数.....	74
3.11	不同停止常数 ϵ 下的网络层数分布.....	74
3.12	NMSE性能随网络层数 L 的变化.....	75
3.13	NMSE性能随网络层数的变化：不同网络架构的比较.....	76
3.14	网络在不同系统参数失配场景下的泛化能力.....	77
4.1	带有DLA的毫米波MU-MIMO系统.....	80
4.2	联合波束选择和数字预编码设计的网络框架.....	83
4.3	波束空间信道的降维过程，其中颜色越深表示波束具有越高的能量.....	87
4.4	DRL网络结构：(a)基于DRL网络的波束选择设计；(b)竞争网络结构；(c)残差网络块结构.....	88
4.5	用于数字预编码设计的模型驱动网络结构.....	91
4.6	联合设计网络的收敛性能.....	96
4.7	联合设计网络的和速率收敛性能.....	97
4.8	系统和速率性能随射频链路数 N_{RF} 的变化.....	97
4.9	系统和速率性能随用户数 K 的变化.....	98
4.10	系统和速率性能随天线数 M_s 的变化.....	98
4.11	系统和速率性能随SNR的变化.....	99
4.12	网络在不同系统参数失配场景下的和速率泛化性能.....	101
4.13	每个用户选择波束的能量.....	102
4.14	系统和速率随信道估计误差的变化.....	102
4.15	不同算法的收敛时间随射频链路数 N_{RF} 的变化.....	103
5.1	毫米波大规模MIMO混合预编码系统.....	107

5.2 传统模块化的通信系统：在发送端执行导频训练和混合预编码设计，在接收端完成信道反馈。	108
5.3 双时间尺度混合预编码的帧结构	109
5.4 双时间尺度网络结构：(a)长时间尺度网络；(b)短时间尺度网络。	111
5.5 用于设计混合预编码的网络架构及数据传输阶段的信号流程图	115
5.6 大规模MIMO混合预编码系统的收发端通信过程	118
5.7 大小为 $D = 3$ 的高维原始信道矩阵缓存滑动窗	119
5.8 双时间尺度网络的收敛性能	128
5.9 系统BER性能随SNR的变化	128
5.10 经过LDPC编码的系统BER性能随SNR的变化	129
5.11 系统BER性能随射频链路数 N_{RF} 的变化	129
5.12 系统BER性能随信道反馈比特数 B 的变化	130
5.13 系统BER性能随导频长度 L 的变化	131
5.14 双时间尺度架构下系统BER性能随时延 τ 的变化	132
5.15 不同模拟移相器量化比特数 Q_{RF} 的系统BER性能随SNR的变化	132
5.16 OFDM系统的BER性能	133
5.17 网络对于不同系统参数失配的泛化能力	133
5.18 基于两阶段训练方式的网络对于信道反馈比特数 B 的泛化能力	134
5.19 迁移学习过程中网络损失函数BCE的变化	135
6.1 语义噪声模型及鲁棒语义通信系统框架	140
6.2 带有离散码本设计的掩蔽VQ-VAE的架构	145
6.3 对抗语义噪声的离散码本设计	149
6.4 包含FIM模块的掩蔽VQ-VAE架构	151
6.5 图像分类准确率随SNR的变化	156
6.6 不同码本大小 J 的图像分类准确率随SNR的变化	157
6.7 图像分类准确率随语义噪声功率 ϵ 的变化	158
6.8 不同语义噪声功率 ϵ 下的图像分类准确率随图像保留率的变化	158
6.9 图像分类准确率随接收端信号-语义噪声比例的变化	159
6.10 不同特征的激活频率	160
6.11 码本中基向量的语义相似性	161
6.12 图像检索与图像重建的性能	162

表 格

1.1	不同算法的优势比较.....	12
1.2	相关文献综述.....	16
2.1	迭代WMMSE预编码算法.....	38
2.2	模型驱动网络的训练过程.....	42
2.3	系统天线数为 $N_t = 8$ 和 $N_t = 16$ 的和速率性能.....	45
2.4	系统天线数为 $N_t = 64$ 的和速率性能.....	45
2.5	系统天线数为 $N_t = 128$ 的和速率性能.....	46
2.6	系统天线数为 $N_t = 256$ 的和速率性能.....	46
2.7	系统($N_t = 64, K = 30$)的和速率随SNR的变化.....	48
2.8	系统($N_t = 128, K = 40$)的和速率随训练样本数的变化.....	49
2.9	系统($N_t = 64, K = 30$)的和速率随模型驱动网络层数的变化.....	49
2.10	CPU运行时间.....	51
2.11	IAIDNN的泛化能力.....	51
2.12	用户满载系统中改进IAIDNN的和速率性能.....	52
3.1	基于SBL的迭代优化算法.....	64
4.1	DRL网络的训练过程.....	90
4.2	DRL网络和模型驱动网络的联合训练过程.....	93
4.3	不同算法的计算复杂度比较.....	94
4.4	不同信噪比下联合和分别训练网络的和速率性能.....	100
6.1	接收端非样本依赖语义噪声的生成算法.....	143
6.2	用于求解问题 $\mathcal{P}3$ 的对抗训练算法.....	144
6.3	鲁棒语义通信系统中模型的训练方法.....	153
6.4	模型的网络结构.....	154

6.5 完成不同任务传输一张图像需要的符号数.....	155
6.6 白盒和黑盒语义噪声下的图像分类准确率.....	159

缩写、符号清单、术语表

缩略词	英文全称	中文全称
5G	The 5th Generation	第五代移动通信
6G	The 6th Generation	第六代移动通信
AC-NN	Analog Combiner-Neural Network	接收端模拟预编码网络
AI	Artificial Intelligence	人工智能
AMP	Approximate Message Passing	近似消息传递
AoA	Angle-of-Arrival	到达角
AoD	Angle-of-Departure	离开角
AP-NN	Analog Precoder-Neural Network	发送端模拟预编码网络
ASP	Audio Signal Processing	音频信号处理
AWGN	Additive White Gaussian Noise	加性高斯白噪声
BCD	Block Coordinate Descent	块坐标下降
BCE	Bit-wise Cross Entropy	比特交叉熵
BER	Bit-Error Rate	误码率
CDF	Cumulative Distribution Function	累积分布函数
CMA	Channel Matching Algorithm	信道匹配算法
CNN	Convolutional Neural Network	卷积神经网络
CSI	Channel State Information	信道状态信息
CV	Computer Vision	计算机视觉
D2D	Device-to-Device	终端直通
DC-NN	Digital Combiner-Neural Network	接收端数字预编码网络
DDPG	Deep Deterministic Policy Gradient	深度确定性策略梯度
DDQN	Double Deep Q-Network	双深度Q网络
DE-NN	Demodulator-Neural Network	解调网络

缩略词	英文全称	中文全称
DFT	Discrete Fourier Transformation	离散傅里叶变换
DLA	Discrete Lens Arrays	离散透镜阵列
DNN	Deep Neural Network	深度神经网络
DP-NN	Digital Precoder-Neural Network	发送端数字预编码网络
DQN	Deep Q-Network	深度Q网络
DRL	Deep Reinforcement Learning	深度强化学习
EM	Expectation-Maximization	期望最大化
eMBB	Enhanced Mobile Broadband	增强型移动带宽
FDD	Frequency Division Duplexing	频分双工
FDMA	Frequency Division Multiple Access	频分多址
FGSM	Fast Gradient Sign Method	快速梯度符号法
FIM	Feature Importance Module	特征重要性模块
FNN	Fully-connected Neural Network	全连接神经网络
GCR	Generalized Chain Rule	广义链式法则
IAIDNN	Iterative Algorithm Induced Deep-unfolding Neural Network	迭代算法驱动的深度展开 网络
IID	Independent Identically Distributed	独立同分布
LDPC	Low-Density Parity-Check	低密度奇偶校验
LSTM	Long Short-Term Memory	长短时记忆网络
LoS	Line-of-Sight	直射径
MDP	Markov Decision Process	马尔可夫决策过程
MU-MIMO	Multi-User Multiple-Input Multiple-Output	多用户多输入多输出
MINLP	Mixed Integer Nonlinear Programming	混合整数非线性规划
MM	Majorization-Minimization	最大-最小优化
mMTC	Massive Machine Type Communications	大规模机器通信
mmWave	MilliMeter-Wave	毫米波
MSE	Mean Squared Error	均方误差
NLoS	Non-Line-of-Sight	非直射径
NLP	Natural Language Processing	自然语言处理
NMSE	Normalized Mean Squared Error	归一化均方误差

缩略词	英文全称	中文全称
OFDM	Orthogonal Frequency Division Multiplexing	正交频分多路复用
OMP	Orthogonal Matching Pursuit	正交匹配追踪
PDD	Penalty Dual Decomposition	惩罚对偶分解
PDF	Probability Distribution Function	概率分布函数
PGD	Projected Gradient Descent	投影梯度下降法
RNN	Recurrent Neural Network	循环神经网络
SBL	Sparse Bayesian Learning	稀疏贝叶斯学习
SGD	Stochastic Gradient Descent	随机梯度下降
SINR	Signal-to-Interference plus-Noise Ratio	信干噪比
SNR	Signal-to-Noise Ratio	信噪比
SVD	Singular Value Decomposition	奇异值分解
TDD	Time Division Duplex	时分双工
TDMA	Time Division Multiple Access	时分多址
URLLC	Ultra-Reliable Low Latency Communications	超可靠低时延通信
VAE	Variational AutoEncoder	变分自编码器
ViT	Vision Transformer	视觉转换网络
VQ-VAE	Vector Quantized-Variational AutoEncoder	矢量量化变分自编码器
WMMSE	Weighted Minimum Mean-Square Error	加权最小均方误差

符号	含义	符号	含义
a	标量 a	∇	梯度算子
\mathbf{a}	矢量 \mathbf{a}	\mathcal{O}	计算复杂度
\mathbf{A}	矩阵 \mathbf{A}	\forall	任意
\in	属于	\exists	存在
\notin	不属于	\emptyset	空集
$ \cdot $	绝对值	\lim	取极限
$\ \cdot\ $	l_2 范数	\rightarrow	趋近于
\mathbf{A}^T	矩阵 \mathbf{A} 的转置	$\operatorname{Re}\{\cdot\}$	取实部
\mathbf{A}^H	矩阵 \mathbf{A} 的共轭转置	$\mathbb{E}\{\cdot\}$	数学期望
\mathbf{A}^{-1}	矩阵 \mathbf{A} 的逆	$\operatorname{Tr}(\mathbf{A})$	矩阵 \mathbf{A} 的迹
$\arg \min f(\cdot)$	函数 f 取最小值时的自变量集合	$\min\{\cdot\}$	取最小值运算
$\arg \max f(\cdot)$	函数 f 取最大值时的自变量集合	$\max\{\cdot\}$	取最大值运算
\mathbb{R}^d	实数域的 d 维向量	\sum	求和运算
\mathbb{C}^d	复数域的 d 维向量	\triangleq	定义为
$\mathbb{R}^{M \times N}$	实数域的 $M \times N$ 维矩阵	\prod	求积运算
$\mathbb{C}^{M \times N}$	复数域的 $M \times N$ 维矩阵	\odot	矩阵 Hadamard 积
$\frac{\partial f(\cdot)}{\partial x}$	函数 f 关于 x 的一阶偏导数	\otimes	矩阵 Kronecker 积

目 次

致谢	I
摘要	IV
Abstract.....	VI
插图	IX
表格	XII
缩写、符号清单、术语表.....	XIV
目次	
1 绪论.....	1
1.1 研究背景与技术挑战	1
1.1.1 研究背景	1
1.1.2 技术挑战.....	3
1.2 深度神经网络模型和理论基础	5
1.2.1 深度神经网络基本结构.....	6
1.2.2 深度神经网络训练方法	12
1.2.3 深度神经网络理论基础.....	14
1.3 深度神经网络在物理层收发机中的研究热点	15
1.3.1 物理层预编码	16
1.3.2 物理层信道估计	20
1.3.3 物理层信道反馈.....	22
1.3.4 新一代物理层通信技术和场景	23
1.3.5 端到端收发机设计.....	24
1.4 本论文研究动机、主要创新点和内容安排	25
1.4.1 主要研究结论及创新点.....	26
1.4.2 内容安排.....	29

2	矩阵形式模型驱动网络框架：大规模MU-MIMO系统预编码设计.....	32
2.1	引言	32
2.2	模型驱动网络框架	33
2.2.1	优化问题的一般形式.....	33
2.2.2	模型驱动网络的前向传播	34
2.2.3	模型驱动网络的反向传播.....	34
2.3	迭代WMMSE预编码算法	36
2.3.1	问题建模.....	36
2.3.2	迭代WMMSE预编码算法设计	37
2.4	基于模型驱动网络的数字预编码设计	39
2.4.1	模型驱动网络的结构和前向传播	39
2.4.2	矩阵形式GCR和网络的反向传播训练.....	41
2.5	计算复杂度和泛化性分析	42
2.5.1	基于CNN的黑盒网络.....	42
2.5.2	参数维度和计算复杂度分析.....	43
2.5.3	泛化性分析	44
2.6	仿真结果	45
2.6.1	仿真设置.....	45
2.6.2	和速率性能.....	46
2.6.3	复杂度分析.....	49
2.6.4	泛化性分析.....	50
2.6.5	IAIDNN在用户满载场景中的改进.....	51
2.7	本章小结	52
3	基于深度强化学习的可变层模型驱动网络：稀疏信道估计.....	53
3.1	引言	53
3.2	基于深度强化学习的可变层模型驱动网络框架	55
3.2.1	问题建模.....	55
3.2.2	基于确定性策略梯度的深度强化学习	56
3.2.3	可变层模型驱动网络框架.....	57
3.3	基于稀疏贝叶斯学习的信道估计	58
3.3.1	大规模MIMO信道的离网基底.....	59
3.3.2	稀疏贝叶斯学习框架.....	60

3.3.3	基于稀疏贝叶斯学习的迭代算法	61
3.4	模型驱动信道估计网络	63
3.4.1	基于稀疏贝叶斯学习算法的模型驱动网络	63
3.4.2	模型驱动网络的性能分析	65
3.5	基于深度强化学习的模型驱动信道估计网络	66
3.5.1	基于DDPG的可变层模型驱动信道估计网络	66
3.5.2	基于DDPG的可变层模型驱动网络的改进	67
3.5.3	基于DDPG的可变层黑盒网络拓展	69
3.6	仿真结果	70
3.6.1	仿真设置	70
3.6.2	NMSE性能	71
3.6.3	网络层数	73
3.6.4	泛化能力	76
3.7	本章小结	77
4	基于联合强化学习和模型驱动网络的波束选择和预编码设计	79
4.1	引言	79
4.2	系统模型和问题建模	80
4.2.1	系统模型	80
4.2.2	波束空间信道模型	81
4.2.3	问题建模	82
4.3	网络的联合设计框架	82
4.3.1	网络联合设计	82
4.3.2	信号流程图	84
4.4	基于深度强化学习网络的波束选择	84
4.4.1	马尔科夫决策过程和价值函数	84
4.4.2	基于深度强化学习的波束选择	85
4.4.3	深度强化学习网络结构	88
4.5	基于模型驱动网络的数字预编码设计	89
4.5.1	迭代WMMSE算法	89
4.5.2	基于模型驱动网络的数字预编码设计	91
4.5.3	计算复杂度分析	92
4.6	仿真结果	95

4.6.1	仿真设置	95
4.6.2	联合设计网络的收敛性	96
4.6.3	和速率性能	96
4.6.4	泛化性能	100
4.6.5	公平性、鲁棒性和复杂度对比	101
4.7	本章小结	103
5	基于双时间尺度端到端学习的信道获取和混合预编码设计	105
5.1	引言	105
5.2	系统模型和问题建模	107
5.2.1	端到端毫米波MIMO系统	107
5.2.2	双时间尺度帧结构	109
5.2.3	问题的数学建模	110
5.3	混合预编码端到端深度学习网络架构	111
5.3.1	端到端深度学习框架	111
5.3.2	导频设计和信道估计	112
5.3.3	信道反馈	113
5.3.4	混合预编码设计	114
5.3.5	双时间尺度通信过程	117
5.4	端到端深度学习网络结构和系统部署	118
5.4.1	双时间尺度网络结构	118
5.4.2	模拟预编码矩阵的优化	119
5.4.3	训练方法	119
5.5	性能分析和系统扩展	120
5.5.1	网络的泛化能力	120
5.5.2	通信传输开销	122
5.5.3	扩展到OFDM系统	123
5.6	仿真结果	125
5.6.1	仿真参数设置	125
5.6.2	网络的收敛性能	127
5.6.3	系统的BER性能	127
5.6.4	网络的泛化能力	133
5.7	本章小结	135

6 基于特征重要性码本和星座点设计的端到端鲁棒语义通信系统.....	137
6.1 引言	137
6.2 鲁棒端到端语义通信系统框架	139
6.2.1 端到端语义通信系统.....	139
6.2.2 语义噪声模型.....	139
6.2.3 对抗训练.....	143
6.3 基于掩蔽矢量量化变分自编码器的码本设计	144
6.3.1 掩蔽矢量量化变分自编码器.....	145
6.3.2 离散码本设计.....	146
6.3.3 码本鲁棒性.....	148
6.3.4 基于码本的高效传输通信系统.....	149
6.4 特征重要性模块和训练方法	150
6.4.1 噪声相关的特征抑制.....	150
6.4.2 具有动态信噪比的特征重要性模块.....	151
6.4.3 模型训练.....	152
6.5 仿真结果	154
6.5.1 仿真设置.....	154
6.5.2 传输开销分析.....	155
6.5.3 图像分类准确率.....	156
6.5.4 白盒和黑盒语义噪声.....	159
6.5.5 特征激活.....	160
6.5.6 语义相似度.....	161
6.5.7 其他任务的性能.....	161
6.6 本章小结	162
7 工作总结和展望.....	163
7.1 工作总结	163
7.2 进一步展望	165
参考文献	167
附录	185
A.1 矩阵形式模型驱动网络框架：大规模MU-MIMO系统预编码设计	185
A.1.1 定理2.1的证明.....	185
A.1.2 输出层矩阵变量的梯度推导.....	186

A.1.3 中间层矩阵变量的梯度推导	186
A.1.4 关于公式(A.7)-(A.8)的详细推导	188
A.2 基于深度强化学习的可变层模型驱动网络：稀疏信道估计	189
A.2.1 定理3.1的证明	189
A.2.2 引理A.2的证明	191
A.3 基于联合强化学习和模型驱动网络的波束选择和预编码设计	192
A.3.1 强化学习的策略梯度	192
A.3.2 启发式算法	193
作者简历	195
攻读博士学位期间的研究成果	196

1 绪论

1.1 研究背景与技术挑战

1.1.1 研究背景

科学技术的变革推动着时代巨轮奔腾向前，永不止步。自二十世纪以来，无线通信技术为了满足无线网络中的服务和应用，已从第一代移动通信（The 1th Generation, 1G）系统发展到第五代移动通信（The 5th Generation, 5G）系统^[1]。相较于传统通信，5G将提供更为多样化的服务^[2]。在功能层面，5G将支持包含通信、控制、计算和内容交付在内的4C服务。在应用层面，5G将支持三大应用场景^[3]：增强型移动宽带（Enhanced Mobile Broadband, eMBB），大规模机器通信（Massive Machine Type Communications, mMTC），以及超可靠低时延通信（Ultra-Reliable Low-Latency Communications, URLLC）。为支持上述应用，研究者提出了多种关键技术，用于提升传输速率、网络容量、通信时延等性能指标，如毫米波（MilliMeter-Wave, mmWave）蜂窝通信技术^[4]、大规模多输入多输出（Multiple-Input Multiple-Output, MIMO）等技术^[5-7]。另一方面，随着深度学习算法的演进、计算机硬件设备性能的提升、海量数据产生和获取的速度加快，深度学习已广泛应用于各领域，并对推动各领域的发展和变革产生了至关重要的作用。

第六代移动通信（The 6th Generation, 6G）系统将拥抱深度学习技术，从而涌现了诸多新型应用，例如元宇宙、自动驾驶、智慧物联网、智慧医疗、工业自动化、虚拟现实、增强现实等，如图1.1所示^[8]。相较于5G应用场景，上述新型应用对无线通信系统提出了更为严格的性能要求^[8-10]。此外，5G系统的部署与商用，逐渐暴露出局限性，难以真正实现万物互联的目标。叠加上互联设备数量的不断增加、无线业务的爆发性增长、网络规模的迅速膨胀且异构化程度不断提升等因素，促进了业界和学界对6G技术的探索。为适配上述新型应用场景，未来的6G系统需要满足比5G系统更为严格的性能要求，例如巨大的通信容量、超高的数据速率和能效、极高的可靠性、毫秒级的延迟，以及全球覆盖和海量链接，同时要求通信系统具备高速且精确处理大量数据、识别和动态适应复杂环境的能力^[11-13]。

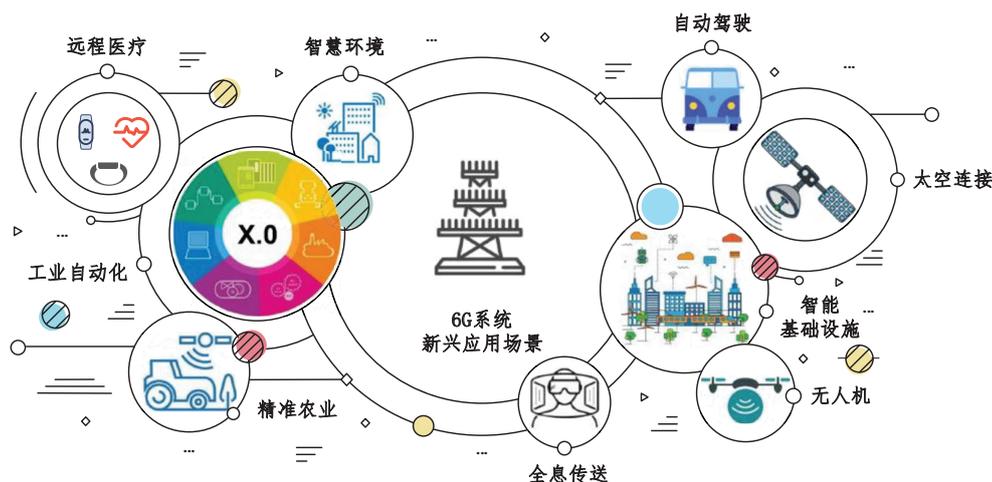


图 1.1 6G应用场景

自2019年以来，国内外多个研究组发布了6G白皮书，包括中国IMT-2030的6G推进组、芬兰6G旗舰项目组、北美电信联盟、国际电信联盟无线电通信组，以及工业界华为2012实验室、诺基亚贝尔实验室、三星等对6G潜在的应用场景和关键技术进行展望^[14-19]。此外，来自国内外多个高校和机构，例如中科院、清华大学、北京邮电大学、东南大学、美国普林斯顿大学、英国帝国理工大学的研究者也提出了6G的发展方向^[20-22]。图1.2展示了5G和6G的发展时间轴及部署进展^[3]。另外，如图1.3所示，6G系统的愿景可概括为：

- 全球覆盖：空天一体化网络，由卫星通信、无人机通信、海洋通信以及陆地通信组成的全方位通信技术，将实现通信网络的全球覆盖。
- 全频谱：为提高传输速率，包括sub-6GHz频段、毫米波、太赫兹等在内的全频谱资源将被充分使用。
- 全应用：随着人工智能（Artificial Intelligence, AI）技术的发展，AI技术与无线通信技术的结合有助于提升网络运行效率并提供新型应用，从三大5G场景扩展到其他新型场景。
- 高度安全：进一步提升网络的安全性，包括物理层和网络层安全。

作为实现上述愿景的基础，AI驱动的新型无线通信系统将成为6G的重要应用场景与关键技术之一。伴随着物联网设备数量的高速增长和AI算法性能的不不断提升，AI将成为6G系统性能提升的重要驱动。为提升通信系统性能并降低复杂度，传统的通信模块，例如：信源编码、信道编码、物理层预编码、信道估计、信道反馈等都可用深度神经网络

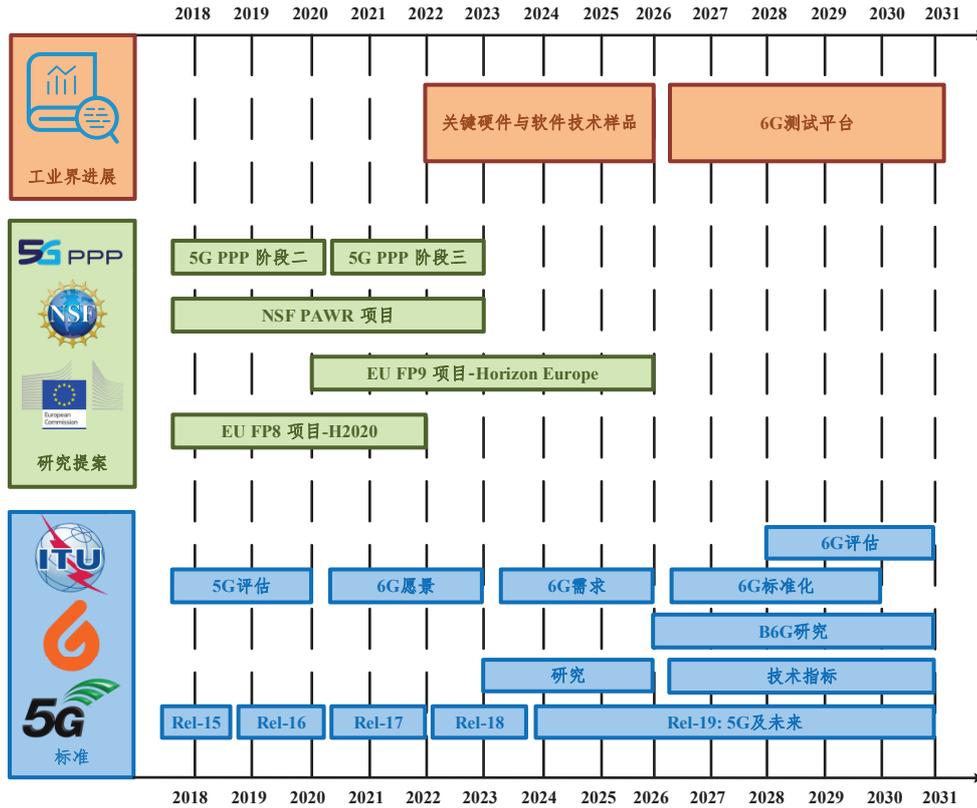


图 1.2 6G系统的部署预计时间表

(Deep Neural Network, DNN) 进行优化。此外，DNN可拟合上述多个模块的功能，实现通信系统的端到端联合设计。根据获取的海量无线通信数据，利用DNN的学习能力从中挖掘物理信道信息并进行信道恢复和重构，从而设计最优的传输方案。AI驱动的物理层技术将打破传统通信系统的模块化设计框架，显著提升无线网络的高效性、可靠性和实时性，在实现针对特定场景的定制化设计方面具有其他方法难以比拟的优势。

1.1.2 技术挑战

为满足6G应用场景的需求，无线通信系统需提供更大的带宽和更高的频谱效率，同时容纳更多的用户。因此，其对算法性能和效率的要求更高。而新一代先进物理层收发机设计是6G系统的关键核心技术，将直接影响通信系统性能和效率的指标。具体而言，物理层收发机设计包括发送机和接收机架构、数模混合预编码设计、信号检测、信道估计和信道反馈等多个收发机模块的设计。传统的迭代优化算法虽可以取得较好的性能，但其通常涉及大量迭代和高复杂度矩阵操作^[23]，计算复杂度较高，导致算法效率不高、实时性不够理想。此外，还存在需对问题进行精确的数学建模、对抗环境变化的鲁棒性较差、收发机各模块独立设计等问题，使其难以在6G应用场景中实时应用和大规模部署。尽管毫米波通

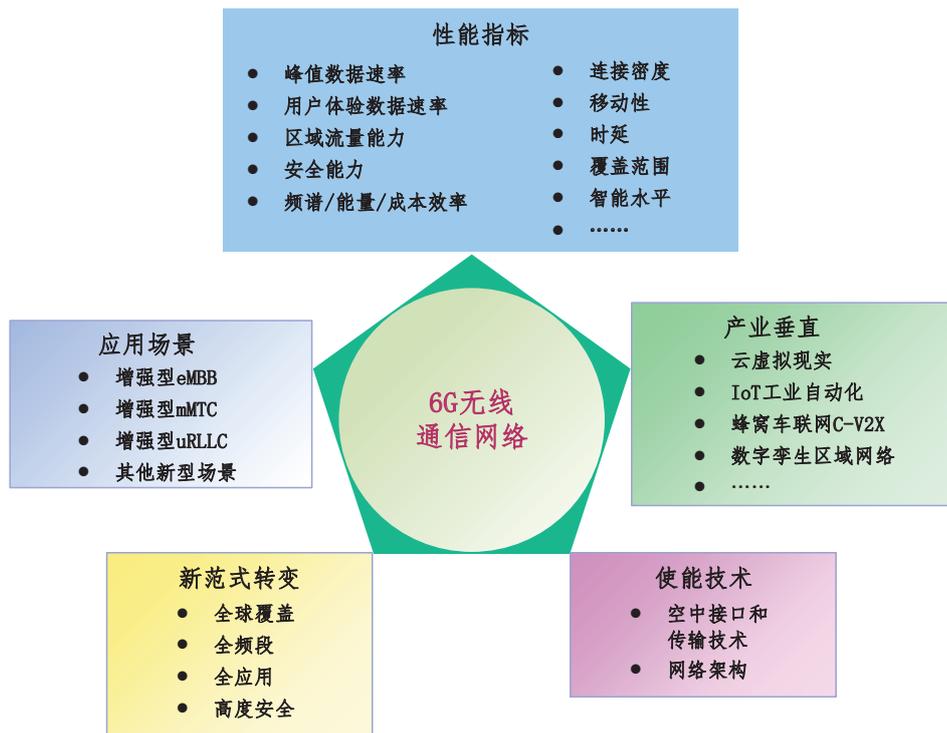


图 1.3 6G无线通信网络概述

信、大规模MIMO等新技术的出现在满足了某些6G场景的通信需求，但在诸多复杂的通信任务中仍存在局限性：

1. 难以进行精确的信道建模：信道条件对通信系统的设计有着至关重要的影响。传统方案依赖于对信道的精确建模，需用数学模型描述信道环境，从而为物理层收发机的设计奠定基础。尽管现有数学模型能够刻画某些通信场景下的信道特征，但由于通信环境复杂多变，大多数场景下的信道难以被精确建模^[24]。
2. 难以满足实时通信传输和信号处理的需求：为提升系统实时性，在通信系统中可采用低复杂度的编译码器、模数转换器等硬件设备。但与此同时，会引入额外的系统误差，需使用具有较高计算复杂度的鲁棒信号处理算法进行补偿^[25]。另外，随着大规模MIMO技术的发展，通信信号矢量的维度增加，使得信号处理算法的计算复杂度和时延显著提升。这些问题导致传统通信算法在实时信号处理问题上遭遇瓶颈^[26]，从而阻碍了在需要快速处理实时数据的通信系统中的部署，例如需针对实时信道进行处理信道估计、信道反馈和预编码算法等收发机模块，以及无人驾驶、智慧医疗等新型应用。
3. 难以实现通信系统各模块的端到端联合设计：传统通信系统由一系列通信信号处理

模块构成，如预编码、信道估计、信道反馈、信号检测等，各模块以分而治之的方式单独构建。在该架构下，研究者通过独立优化各模块来提升系统性能。尽管各模块的算法不断被优化，但其独立设计导致距离实现整个通信系统的最佳性能仍有较大距离。

深度学习为上述问题提供了一种潜在解决方案，从而有效替代迭代优化算法^[27]。相比传统优化算法，深度学习的计算复杂度较低、对抗误差的鲁棒性较好、各通信模块可进行端到端联合设计。具体而言，将迭代优化算法视为一个“黑盒”，并使用DNN学习输入和输出之间的映射关系。尽管DNN实现了较好的性能，但其解释性和泛化能力较差，在动态变化环境的实际场景中难以保证性能下界。此外，DNN大多是基于传统神经网络结构的“黑盒网络”，通常是数据驱动的，高度依赖于对大量带有标签的训练数据，网络训练时间很长。然而，对于实际的通信系统和应用场景，带有标签的训练数据和计算资源都是稀缺的，因此需要对这些方法进一步改进，设计针对通信系统和场景的网络结构。对此，有研究者提出模型驱动网络^[28]，其充分利用已有的先验知识、数学模型和迭代算法，将通信系统中已有的迭代算法展开为类似于DNN的逐层结构。此外，在展开过程中引入一系列训练参数，替换原有迭代算法中的高复杂度操作，加快网络收敛速度，同时提升系统性能。模型驱动网络的性能往往优于黑盒网络，以更低的计算复杂度实现了接近原有迭代算法的性能，同时具有较好的可解释性和泛化能力，在面对复杂动态的通信环境时也表现出了较好的适应性。

1.2 深度神经网络模型和理论基础

深度学习起源于一个三层或三层以上的DNN，从大量数据中学习关键特征，构建复杂的概念并理解高层语义信息，从而更好地完成相应的任务。深度学习的研究有着悠久的历史。自从文献[29]于1958年提出单层感知器，DNN受到了广泛的关注和研究。1990年，文献[30]提出了循环神经网络（Recurrent Neural Network, RNN）。1997年，文献[31]提出了长短时记忆网络（Long Short-Term Memory, LSTM）。1998年，文献[32]提出了卷积神经网络（Convolutional Neural Network, CNN）。但受限于当时硬件技术和计算机算力，这些网络仅能完成一些简单的图像处理问题。随着技术的发展和革新，大规模并行计算日趋成熟；在大数据和万物互联的时代背景下，带标签的训练数据集不断增加。由此，DNN重新进入学界的视野成为热点研究话题，且网络规模越来越大。随着研究的深入，DNN解决了越来越复杂的问题，且性能指标不断提升。在计算机视觉（Computer Vision, CV）领域，DNN在图像识别^[33]、图像去噪^[34]、语义分割^[35]等应用场景中都取得了较好的性能。

在自然语言处理（Natural Language Processing, NLP）领域，DNN已广泛应用于机器翻译^[36]、人机交互^[37]、机器阅读理解^[38]等场景。近期大模型chatGPT的诞生对NLP领域产生了革命性的影响，显著提升了相关领域内各任务的效率和性能。在音频信号处理（Audio Signal Processing, ASP）领域，DNN被广泛应用于语音识别^[39]、语音信息检索^[40]等相关任务。

在推动深度学习的发展和变革中，海内外科技公司和研究机构起到了至关重要的作用。在国际上，Google、IBM、Microsoft、Netflix、NVIDIA 等各大科技公司开展了深度学习相关的研究。在中国，华为、百度、阿里先后成立相关实验室和研究院，并开发了AI平台。在学术界，国内外名校都将深度学习作为重要战略方向，其中斯坦福、加州大学伯克利分校等名校的诸多教授都是相关领域研究的先驱。接下来，本节将从DNN的基本结构、训练方法和理论基础等三个方面进行介绍。

1.2.1 深度神经网络基本结构

深度神经网络可以大体分为两类：数据驱动的黑盒网络和模型驱动网络，下面将分别展开介绍。

1.2.1.1 数据驱动的黑盒网络

顾名思义，数据驱动的黑盒网络是指将DNN看作一个黑盒，依靠DNN强大的学习能力和大量带标签的训练数据实现特定的功能（为描述方便起见，下称“黑盒网络”）。数据驱动的黑盒网络通常是CV和NLP领域最流行的解决方案，这主要是因为这些任务难以用数学工具进行精确建模，且存在大量带标签的训练数据可供DNN学习。著名的网络包括VGG-Net^[41]、GoogleNet^[42]、ResNet^[43]、ChatGPT等，常见的网络结构有全连接神经网络（Fully-connected Neural Network, FNN）、CNN、变分自编码器（Variational AutoEncoder, VAE）、RNN、LSTM、视觉转换网络（Vision Transformer, ViT）、深度强化学习（Deep Reinforcement Learning, DRL）等。在不依赖数学模型和领域知识的情况下，研究者可针对各种各样的问题设计具有不同结构的黑盒网络，从大量带标签的训练数据中学习不同的机制。黑网络的训练过程和参数调整可通过工程实践，借助相关深度学习平台完成。

FNN是一种最基本的DNN，具有多层的感知机结构^[44]。它由三层或以上的全连接层组成，相邻两层的每个神经元都互相连接。该多层结构使其具备学习复杂映射关系的能力。FNN由输入层、隐藏层和输出层组成，其中隐藏层可以更好地提取数据的特征，但过

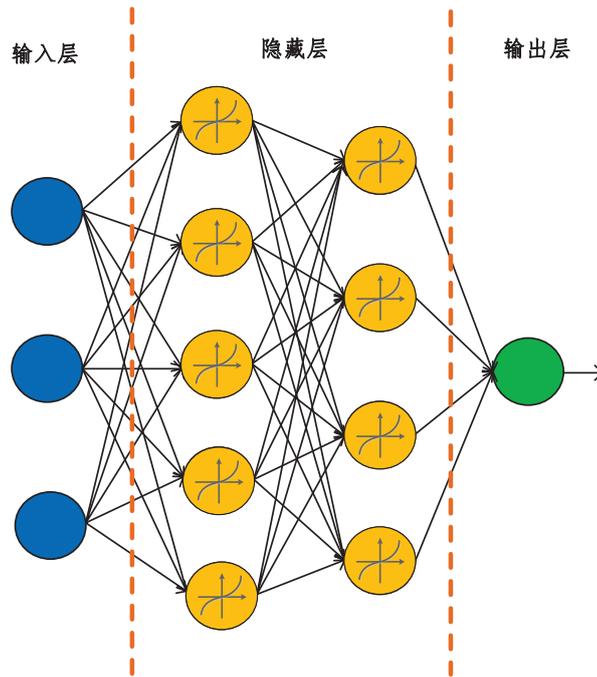


图 1.4 FNN结构示意图

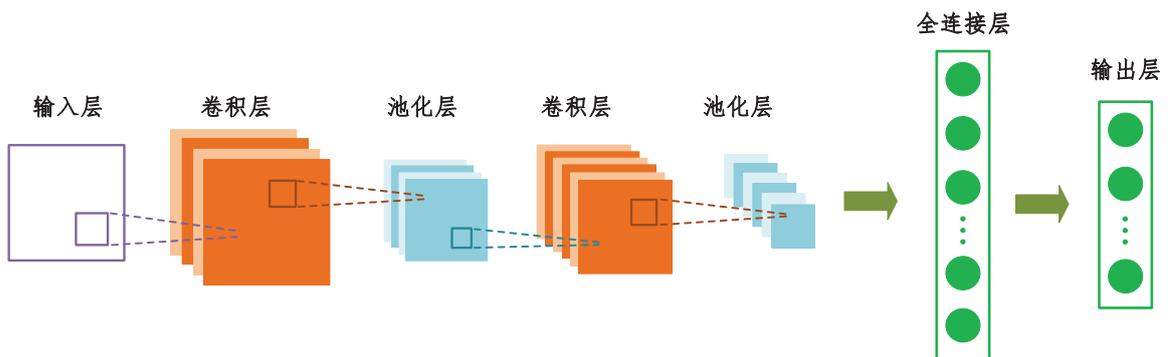


图 1.5 CNN结构示意图

多的隐藏层会导致过拟合问题。如图1.4所示，FNN通过输入层输入样本，在每个隐藏层与神经元的权重相乘、偏置相加，对不同神经元的输出结果进行求和，随后通过激活函数输出结果，最终通过输出层输出结果。

CNN是目前应用最为广泛的DNN结构之一，在图像处理中发挥着重要的作用^[32]。其包含的卷积层和池化层，有助于CNN学习数据的高维特征。如图1.5所示，CNN采用卷积层和池化层交替连接的方式提取样本特征，随后通过全连接层和输出层，将从卷积层提取出来的高维特征映射到样本-标签空间，最终得到CNN的输出结果。具体而言，卷积层由若干特征图组成，每个特征图由若干神经元组成。同一特征图中的神经元共享权值，该共享权值被称为卷积核。由于不同卷积核的权值不同，因此可提取数据的不同特征。尽管相

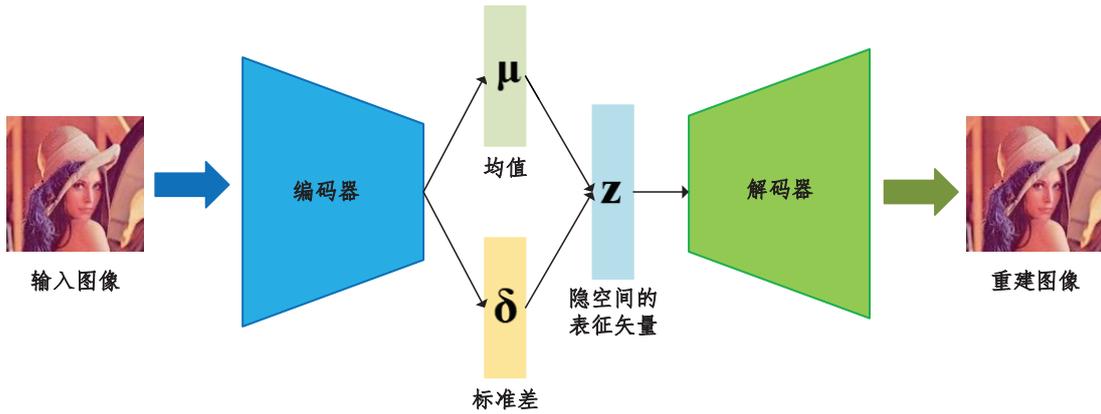


图 1.6 VAE结构示意图

比于全连接层，卷积核的设计减少了网络参数数量，但由于特征图的数量较大，网络参数的维度仍较高，存在过拟合的风险。而池化层可对不同位置的特征进行选择 and 聚合，从而进一步减少网络参数数量，有助于改善过拟合现象。较常采用的池化方式有均值池化和最大值池化。卷积和池化操作能够显著降低网络的计算复杂度，这使得CNN被广泛用于设计大型网络。

VAE是一类用于数据重建的DNN架构，并可以用于生成任务和数据恢复任务^[45]。如图1.6所示，它可以实现针对高维数据的降维，并实现较高精度的数据重建。具体而言，它将输入编码为隐空间上的某一分布，即编码器的输出是均值向量 μ 和标准差向量 δ 。隐空间中的表征矢量 z 从一个预先定义的分布采样得到：

$$z = \mu + \delta\epsilon, \quad (1.1)$$

其中， $\epsilon \sim \mathcal{N}(0, I)$ 。随后，将隐空间的表征矢量 z 发送到解码端，用于重建原始数据。VAE的训练有两个目标：重建原始数据、确保隐空间中的表征矢量 z 是正态分布。因此，其损失函数为重建损失与相似损失之和。重建损失一般采用网络输入与输出之间的均方误差损失（Mean Squared Error, MSE），而相似损失采用标准高斯分布和隐空间分布之间的Kullback-Leibler散度。

RNN是另一种常见的DNN结构，它在时间序列信号处理任务中发挥着举足轻重的作用^[30]。不同于FNN和CNN，RNN的神经元存在时间上的共享特性。在FNN和CNN中，相邻两层的神经元相互连接，网络每一层内部的神经元之间没有连接。因此，不同输入样本之间相互独立，难以处理时间序列样本。而在RNN中，网络层内部的节点相互之间也存在连接，每层网络当前时刻的输入由两部分组成：该层的上一时刻输出和上一层当前时刻的输出。因此，RNN能记住过去时刻的输入数据及其特征，并使其对当前时刻输出的计算结

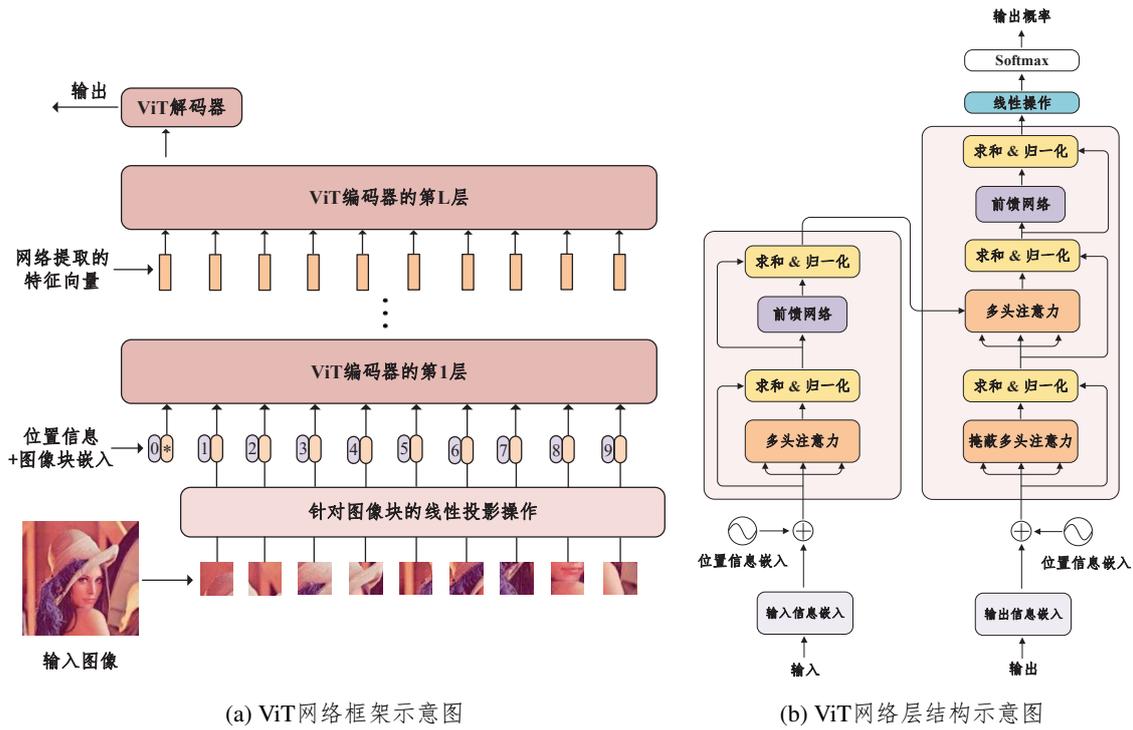


图 1.7 ViT网络

果产生影响。为解决一般的RNN存在的长期依赖问题，即梯度随着时间长度的增加而逐渐消失，在RNN的基础上提出了LSTM。LSTM由遗忘门、更新门、输出门、门控循环单元等结构组成，适合处理时间序列中间隔和延迟较长的输入^[31]。

然而，RNN和LSTM仅能从左往右或从右往左依次按序计算，该机制带来了两个问题：(1) t 时刻的输出依赖于 $t - 1$ 时刻的输出，限制了网络的并行能力；(2) 顺序计算的过程存在信息丢失和长期依赖现象，尤其是当时间序列较长时。尽管LSTM一定程度上解决了该问题，但对于特别长期的依赖现象，LSTM表现不佳。由此，提出了ViT架构，如图1.7(a)所示。首先，不同于RNN的顺序结构，ViT的架构具备更好的并行特性，更适配现有的GPU框架，便于快速高效训练；其次，设计了注意力机制，将时间序列数据中任意两个时刻的间隔转换为常量，从而更好地解决长期依赖问题。在操作过程中，将原始图像分割为若干图像块，结合块位置编码信息，输入ViT编码器进行特征提取，随后通过ViT解码器输出最终结果。图1.7(b)描述了ViT网络详细的层结构。

DRL是深度学习的一个分支，是一种融合了DNN和强化学习的方法，能高效处理可建模为马尔可夫决策过程（Markov Decision Process, MDP）的问题。相较于深度学习中经典的监督学习、无监督学习问题，DRL最大的特点是在交互中学习。智能体在与环境的交互过程中根据获得的奖励或惩罚不断地学习知识，从而更好地适应环境。DRL的范式类似

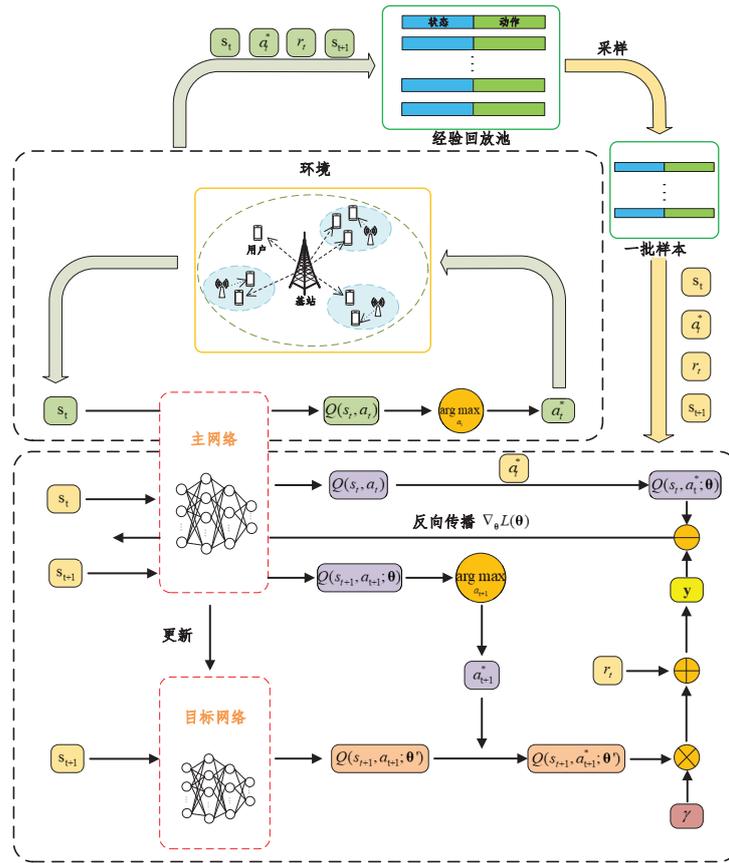


图 1.8 DRL示意图

于人类学习知识的过程，因此被视为实现通用AI的重要途径。如图1.8所示，DRL的学习步骤可概括为：（1）在当前时刻，智能体处于某状态，并利用DNN来感知当前状态，以得到该状态的特征表示；（2）在当前状态下，根据预期奖励来评估各动作的价值，并通过策略将当前状态映射为相应的动作；（3）智能体执行该动作与环境交互，得到相应的奖励或惩罚，转移到下一个状态。通过重复执行上述步骤，获得实现目标的最优策略。

双深度Q网络（Double Deep Q-Network, DDQN）是DRL中较为成功且应用广泛的一种方案，如图1.8所示。其针对一般的DRL方案做了进一步的改进：（1）采用DNN进行Q函数拟合，进行端到端的模型训练。（2）经验回放：在DNN取得重大进展的监督学习中，不同输入样本间都是独立同分布（Independent Identically Distributed, IID）的。而DRL中的样本往往是高度关联且非静态的，因此训练的结果难以收敛。经验回放机制构建一个用于存储训练样本经验回放池，通过随机采样方式去除样本间的相关性。（3）双网络结构：一般的DRL网络结构中，评估动作和选择动作采用相同的网络，这使得模型震荡和发散可能性变大。构建目标网络用于选择动作，主网络用于评估动作，从而避免网络训练过程中的震荡和发散，使得训练过程更加稳定。关于DRL的详细介绍请见附录A.3.1。

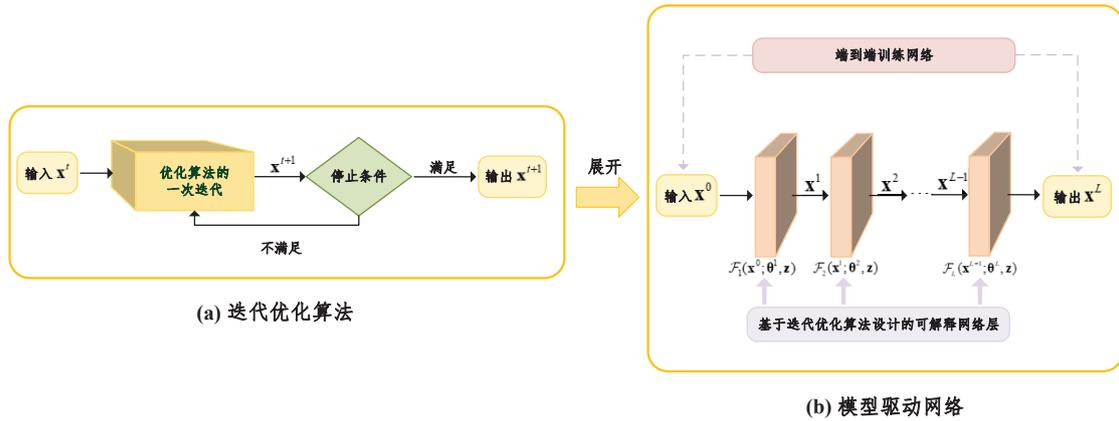


图 1.9 迭代优化算法和模型驱动网络示意图

1.2.1.2 模型驱动网络简介

训练数据驱动的黑盒网络需要大量的计算和存储资源、较高的时间成本和大规模的带标签训练数据集，而这几种资源在通信系统中较为紧缺。当计算资源受限时，黑盒网络训练时间过长；当训练数据集较小时，黑盒网络难以学习到有效信息且泛化能力较差。此外，现阶段缺少对黑盒网络拓扑结构与性能关系的理论分析框架，且缺乏针对特定场景的设计和相应的可解释性，性能往往无法得到理论上的保证。这使得黑盒网络在无线通信领域的应用受到了一定的限制。为了克服这些缺点，文献[46–49]提出一种新型网络结构，即将迭代算法展开为类似于黑盒网络的逐层结构，被称为模型驱动网络^[28]。不同于黑盒网络，模型驱动网络将领域知识与深度学习方法结合，以低于传统迭代算法和黑盒网络的计算复杂度实现了更好的性能，且所需训练时间较短、对计算资源的需求较低、训练过程不依赖于大量带标签的数据。模型驱动网络主要由三部分组成：问题的数学模型、针对该问题的迭代算法和基于该迭代算法设计的网络^[50]。

- 数学模型：根据领域相关的专业背景知识，将现实世界中的问题抽象为相应的数学模型；
- 迭代优化算法：根据数学模型和领域知识，设计迭代优化算法求解；
- 模型驱动网络：根据迭代优化算法，在相应位置引入训练参数，设计模型驱动网络，采用反向传播算法训练。该网络将迭代算法展开为信号流图，因此也被称为深度展开（Deep-Unfolding）网络^[51]。

图1.9展示了迭代优化算法和对应的模型驱动网络。相比数据驱动的黑盒网络，模型驱动网络往往更适用于无线通信系统。无线通信领域的学者经过数十年的深入研究而得到的

表 1.1 不同算法的优势比较

算法名称	性能	运行效率	参数数量	鲁棒性	可解释性	泛化性
迭代优化算法	高	低	低	低	高	高
模型驱动网络	高	高	中	高	中	中
黑盒神经网络	低	高	高	中	低	低

领域知识和数学模型，可用于模型驱动网络的设计。其继承了传统迭代优化算法的优点，同时避免了对问题精确建模的要求。此外，由领域知识和数学模型的不准确所带来的误差，可通过深度学习来弥补。模型驱动网络还具有所需要的训练样本少、过拟合风险小、训练速度快等优点，且已逐步应用于物理层收发机设计，有望成为实现6G系统指标的高效解决方案。

表1.1总结了各种方法的特性。针对不同的通信问题和场景，需要根据不同方法的特点，设计能满足需求的网络结构，从而对问题进行高效求解。

1.2.2 深度神经网络训练方法

激活函数、损失函数和学习策略是DNN设计的关键，下面将分别进行介绍。

1.2.2.1 激活函数的选择

激活函数将非线性引入DNN，显著提升了DNN的学习能力，同时也会影响DNN的训练和稳定性。因此，激活函数的选择非常重要，一般需具备如下特点：

- 非线性：是激活函数的本质特征，也是DNN具备强大学习能力的基础；
- 连续且可微：保证DNN在训练过程中，参数梯度的反向传播过程顺利进行；
- 单调性：保证激活函数的梯度方向不会频繁改变，提升DNN训练过程的收敛速度和稳定性；
- 输出范围有限：保证了DNN的输出平稳，有利于训练过程的稳定，防止因输入过大导致网络输出的波动和震荡；
- 接近恒等变换：满足 $f(x) \approx x$ ，保证激活函数的输出不会随网络深度的增加而显著增加，从而使网络输出更稳定，易于训练。

常用的激活函数都满足上述特征，如Sigmoid、Tanh、L-ReLU函数等，具体表达式分别为：

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \in (0, 1), \quad (1.2a)$$

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \in (-1, 1), \quad (1.2b)$$

$$\text{L-ReLU}(x) = \begin{cases} x, & x \geq 0, \\ \alpha x, & x < 0. \end{cases} \quad (1.2c)$$

1.2.2.2 损失函数的选择

损失函数将一个或多个变量的值映射为一个实数，用于计算DNN实际输出和预期输出的差值，表达与输入样本相关的损失。下面给出三个最常见的损失函数：

- 均方误差损失：常用于回归任务中，计算网络输出和标签之差的 l_2 范数，表示为：

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N (y_n - \hat{y}_n)^2, \quad (1.3)$$

其中 \hat{y}_n 表示网络输出， y_n 表示标签。由于该损失函数对误差平方化，因此训练收敛速度较快。

- 均绝对误差损失：计算网络输出和标签之差的 l_1 范数，表示为：

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^N |y_n - \hat{y}_n|. \quad (1.4)$$

平均绝对误差损失的收敛速度虽然没有均方差损失快，但其对异常点更加鲁棒。

- 交叉熵损失：是一种对数似然函数，常用于分类任务中，表示为：

$$\mathcal{L} = -\frac{1}{N} \sum_{n=1}^N [y_n \ln(\hat{y}_n) + (1 - y_n) \ln(1 - \hat{y}_n)]. \quad (1.5)$$

使用交叉熵损失时，网络参数更新的速度会受误差影响，当误差越大，参数更新越快。

1.2.2.3 学习算法的选择

反向传播算法是深度学习的基础，结合了链式法则和梯度下降法。可用于训练绝大多数的DNN，使其可较好地拟合从输入到输出的函数。该算法计算损失函数关于网络每层训

练参数的梯度，并用梯度下降法更新参数，从而最小化损失函数。目前较为常用的DNN学习算法有：

- 随机梯度下降（Stochastic Gradient Descent, SGD）算法^[52]：每次从训练数据集中随机选择一个样本学习，收敛速度较快，也可用于在线训练。
- Momentum算法^[53]：能解决SGD易陷入局部最优、易震荡等问题，加速SGD在正确方向的下降并抑制震荡。
- RMSprop算法^[54]：对二阶动量使用指数移动平均，在参数空间平缓的方向加快学习速度，而在陡峭的方向放慢学习速度。因此，在损失函数非凸时能取得较好的训练效果，且能加快收敛速度。
- ADAM算法^[55]：对一阶动量使用指数移动平均，能够自适应地调整学习率，可看作RMSprop算法和Momentum算法的结合。

1.2.3 深度神经网络理论基础

DNN本质上是一种通用的非线性拟合器。文献[56]在数学上严格地证明了具有单隐藏层的标准三层神经网络在理论上能够以任意精度拟合任意非线性连续函数。

定理 1.1. 对于任意输入空间，一个具有任意激活函数的单隐藏层神经网络能近似任意可测函数^[56]。

该定理可以推广到多层DNN。注意到DNN的拟合能力来源于其训练参数（即权重和偏置）可以取到足够小的数。例如，逼近一个三维函数，DNN能够通过调节训练参数的值，来模拟出任意位置、高度和宽度的长方体，当足够多个这样不同位置、高度和宽度的长方体堆叠起来就能够近似出任意三维函数。当DNN拥有足够的深度和宽度时，上述学习模式可拓展至任意维的空间中，用于拟合任意函数。如若没能实现理想的拟合效果，通常是由如下原因导致：（1）训练数据和时间不足；（2）神经元数量不足；（3）输入与目标之间缺乏确定性函数关系。

此外，文献[44]指出：三层神经网络足以表示任何函数，但当该层所包含的神经元数量非常多时，难以达到较好的学习效果。一般来说，更深的网络相比更宽的网络，为实现相同的性能所需的神经元数量更少，且拟合能力和泛化性能更好。隐藏层数和神经元数都会影响DNN的收敛速度，文献[56]指出DNN的收敛速度会随神经元数量的增加而提升。尽

管深层网络和浅层网络的学习能力几乎相同，但深层网络更容易训练^[42]。相比浅层网络，深层网络通常对高维特征具有更强的提取和表征能力，从而可以更好地对复杂的非线性系统建模。因此，DNN的层数日益增多，并被广泛应用于各学科的不同场景中^[44]。

1.3 深度神经网络在物理层收发机中的研究热点

近年来，深度学习技术的迅速发展，使得利用DNN解决无线通信领域的复杂需求，充分赋能未来6G移动通信已成为必然趋势^[57]。与传统的迭代优化算法相比，DNN能够从训练数据中提取相关特征和关键信息^[58]，且可以实现高效并行计算以加速网络训练，通过网络参数的训练和调整实现端到端系统设计^[59]。因此，DNN凭借其强大的学习能力、处理大数据的优势，在物理层和网络层等无线通信场景中有着广泛的应用，如图1.10所示^[8]。值得注意的是，DNN在物理层收发机关键模块的设计也受到了广泛关注，如信号调制识别和检测^[60]、资源分配^[61]、信道建模和识别^[62]、编码和解码^[63]、信道估计和反馈^[64]等。在通信系统的物理层收发机设计中，DNN有着广阔的应用前景：

1. 理论已证明，DNN拥有强大的学习能力，可模拟任意函数^[65]。因此，DNN有望高效求解传统算法难以解决的复杂通信问题，突破已有的性能界限。
2. 无需精确的数学建模，能减少对领域内专业知识的需求，例如信道建模^[24]、端到端系统设计^[66]等。
3. 在未来通信场景中，大量用户同时在线产生海量数据，这要求通信系统能够对大量数据进行实时处理。另外，大规模MIMO系统中收发机配备的天线数量较大，导致通信算法需处理高维数据，这显著增加了通信信号处理的计算复杂度和时延。由于DNN可采用高效的并行计算架构、从大数据中学习重要特征，因此DNN能够保证计算效率、高效处理大数据，满足未来通信场景的需求。
4. 得益于AI技术的快速演进，已建立成熟的深度学习框架，如PyTorch^[67]、Tensorflow^[68]、Caffe^[69]、Theano^[70]等，可以加速深度学习算法的实验和部署。
5. 分别根据各模块的性能指标设计的模块化通信系统架构，难以保证系统实现整体最优的性能。相比之下，DNN能够打破模块化通信架构的局限，通过收发机的端到端联合设计，进一步提升通信系统的时延、传输速率、误码率等方面的性能^[24]。

本节将综述用迭代优化算法、黑盒神经网络、模型驱动网络等工具解决物理层预编码、信道估计、信道反馈、新一代物理层通信技术和场景以及端到端收发机设计等方面的

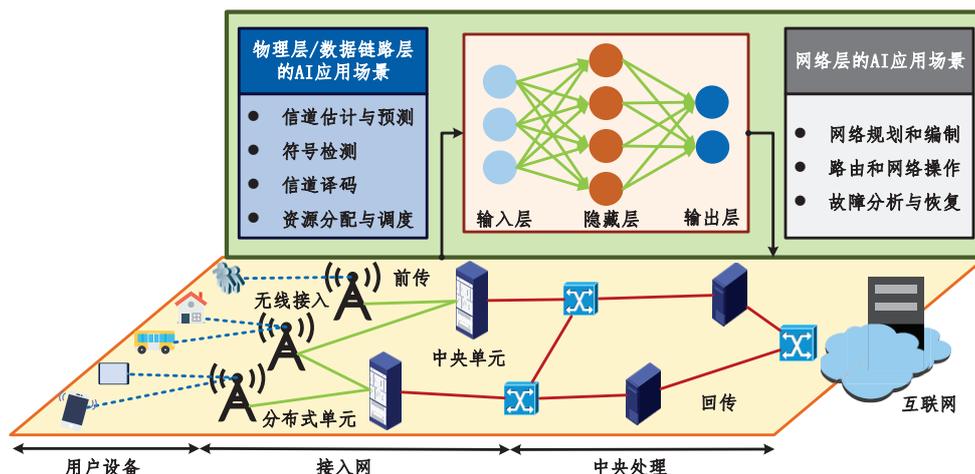


图 1.10 神经网络在无线通信场景中的应用

表 1.2 相关文献综述

通信问题	细分场景	迭代优化算法	黑盒神经网络	模型驱动网络
物理层预编码	数字预编码	[80–89]	[90–96]	[97–102]
	混合预编码	[103–112]	[113–119]	[120–123]
	波束选择	[124–132]	[133–140]	[141]
物理层信道估计	基于导频序列	[143–145], [152–165]	[166–169]	[172–174]
	盲信道估计	[146–148]	[170], [171]	[175], [176]
	半盲信道估计	[149–151]		
物理层信道反馈	基于压缩感知	[177–179]	[184–200]	[201–203]
	基于码本	[180–183]		
新一代物理层技术和场景	新一代物理层技术	[60–64]	[211–213]	[204–208]
	新一代物理层场景	[57–59]	[214–218]	[209–211]
端到端收发机设计	端到端物理层收发机	-	[219–227]	
	端到端语义通信系统	-	[228–254]	

已有文献，具体分类请见表1.2。接下来，将对这些文献进行详细介绍和归类。

1.3.1 物理层预编码

近年来，毫米波（MilliMeter-Wave, mmWave）通信由于其巨大的带宽，被认为是能满足下一代无线通信高数据传输速率要求的关键技术之一^[71]。毫米波较短的波长使得在实际通信系统中部署大规模天线阵列成为可能，从而实现定向传输^[72–74]。另外，毫米波大规模多用户多输入多输出（Multi-User Multiple-Input Multiple-Output, MU-MIMO）系统可为空间多路复用提供了较大的阵列增益，从而有望提高系统容量并缓解频谱短缺问题^[75]。为实现上述目标，毫米波通信系统中用于提高频谱效率的高效预编码算法设计至关

重要^[76-78]。预编码的核心思想是，通过矩阵运算把经过调制的发送符号和信道状态信息（Channel State Information, CSI）进行有机结合，变换成适合于当前信道传输的数据流，通过天线发送至接收端，使得被服务目标能获得尽可能大的信噪比（Signal-to-Noise Ratio, SNR）。传统通信系统往往采用全数字预编码架构，即需要为每根天线配备一条射频链路。这导致了昂贵的硬件制造成本和极高的能耗，成为实际系统部署的主要障碍之一。为了解决该问题，研究者提出了混合预编码结构，其包含数字预编码和模拟预编码。具体而言，大量的天线连接着较少的射频链路，因此其中的模拟预编码矩阵带有恒模约束，从而大大降低了硬件成本和能耗。为进一步降低通信开销和能耗，研究者提出了离散透镜阵列（Discrete Lens Arrays, DLA）硬件结构^[79]。其改进了混合预编码架构，而采用波束选择架构，即从一系列波束中选择使得系统通信效率最高的波束作为预编码矩阵。本小节将介绍数字预编码、混合预编码、波束选择相关的文献。

1.3.1.1 数字预编码

基于传统优化理论和信息论设计的针对MU-MIMO系统的数字预编码方案分为线性预编码^[80-86]和非线性预编码^[87-89]。其中，文献[80]提出了迫零预编码算法，文献[81]提出了匹配滤波预编码算法，文献[82]提出了MMSE预编码算法。在上述经典线性预编码算法的基础上，文献[83]提出了用于最大化广播信道下MIMO系统和速率的线性预编码方案。文献[84]提出了用于MIMO干扰系统的迭代注水预编码方案。文献[85]使用半定松弛算法设计MIMO广播系统的预编码矩阵。文献[86]为了求解MU-MIMO系统中的和速率最大化问题，将和速率最大化的问题被等价转换为最小均方误差问题，结合块坐标下降（Block Coordinate Descent, BCD）的思路设计了加权最小均方误差（Weighted Minimum Mean-Square Error, WMMSE）预编码算法求解该问题。关于非线性预编码方面的研究，文献[87]提出恒包络预编码算法，各天线均发送恒包络信号，符号信息包含在相位中。文献[88]提出脏纸编码，证明了当基站端完美已知信道信息时，使用脏纸预编码可实现最优性能，该性能可作为线性预编码算法的参考上界。文献[89]提出了非线性预编码技术，可均衡由下行链路子信道产生的干扰。相较于线性预编码，非线性预编码的性能虽然有一定增益，但复杂度往往较高，在通信系统中实际应用得较少。

虽然这些迭代优化算法得到的预编码矩阵实现了较好的性能，但由于大规模MU-MIMO系统的优化算法中存在高维矩阵求逆等高复杂度操作，且需要大量的迭代，导致算法的计算复杂度较高，阻碍了在实际系统中的应用。为降低预编码算法的复杂度，文献[90-96]提出了基于黑盒网络的预编码算法。其中，文献[90]和[91]分别设计了多层感知器和CNN近似迭代WMMSE算法，并采用了监督学习。为了提升系统性能并解决训练过程

中标签难以获取的问题,文献[92]提出了一种使用无监督学习的高效功率分配算法。不同于上述文献需要已知信道信息来设计预编码矩阵,文献[93]设计了基于CNN的预编码网络,且不需要事先已知信道信息。不同于一般根据信道设计的预编码矩阵,文献[94]提出了基于深度学习的符号预编码,即针对不同符号设计不同的预编码矩阵。文献[95]提出了基于深度学习的自适应预编码算法。文献[96]基于迁移学习和元学习设计了能快速适应动态环境的预编码算法。

为提升网络的可解释性和泛化能力,研究者提出了基于模型驱动网络的数字预编码算法^[97-102]。文献[97]提出了基于干扰消除预编码的模型驱动网络。文献[98]提出一种模型驱动网络,利用最优下行预编码结构设计有效的混合学习策略,以最大化和速率性能。文献[99]提出了一种混合网络结构,结合了黑盒网络和模型驱动网络的优势。文献[100]提出了一种基于共轭梯度算法的模型驱动网络,用于恒包络预编码。文献[101]提出了一种用于大规模MU-MIMO系统中有限字母预编码的模型驱动网络。文献[102]设计了可用于频分双工(Frequency Division Duplex, FDD)系统中非完美信道场景下预编码的模型驱动网络。

1.3.1.2 混合预编码

文献[103-109]设计了针对混合预编码的迭代优化算法,联合设计数字和模拟预编码矩阵。其中,文献[105]提出了一种新颖的惩罚对偶分解(Penalty Dual Decomposition, PDD)优化算法,用于求解MIMO系统下的混合预编码问题。文献[106]证明,混合预编码系统如果具有两倍于数据流的射频链路数,则可实现接近于全数字预编码系统的性能。文献[107]提出了一种基于流形的混合预编码迭代算法。文献[109]的作者提出了一系列低复杂度近似求解算法,用混合预编码系统实现逼近全数字预编码系统的性能。不同于上述文献直接优化预编码矩阵,文献[108]提出了码本驱动的混合预编码迭代算法,设计了一个用于预编码的码本。上述研究在相同的时间尺度优化数字和模拟预编码矩阵,但这会导致较大的信道估计和反馈开销,且算法复杂度往往较高。为解决该问题,文献[110]提出了一种双时间尺度框架,使得模拟预编码矩阵的更新频率慢于数字预编码矩阵,从而降低系统信道估计和反馈开销,以及算法的计算复杂度,且通过随机优化算法的设计实现了较好的性能。基于该思想,文献[111]和文献[112]分别提出了双时间尺度混合预编码优化算法用于MU-MIMO广播信道场景和全双工多中继通信场景。

为进一步降低传统迭代优化算法的计算复杂度,便于应用在实际系统中,文献[113-119]针对不同混合预编码场景设计了深度神经网络。具体而言,文献[113]较早提出了使用FNN,采用反向传播的网络训练方式,端到端联合设计数模混合预编码矩阵。文

文献[114]提出了一种基于深度学习的混合预编码框架，将预编码矩阵的每次选择都被视为DNN中的一个映射关系。文献[115]设计了基于CNN的混合预编码框架，以不完美的信道矩阵作为网络输入，且运行时间小于传统方案。文献[116]和文献[117]设计了基于非监督深度学习的混合预编码网络，可以较好地处理非完美信道场景。文献[118]联合设计了用于估计信道到达角（Angle-of-Arrival, AoA）和混合预编码的网络。文献[119]设计的混合预编码网络只需已知信道的统计特性，从而大大降低了信道估计和反馈的开销。为提升网络的可解释性和泛化能力，研究者提出了基于模型驱动网络的混合预编码算法^[120-123]。文献[120]提出了逼近奇异值分解（Singular Value Decomposition, SVD）算法的模型驱动网络。文献[121]设计了经过网络剪枝的轻量级模型驱动网络，以非完美CSI作为输入，输出大规模MU-MIMO系统的模拟预编码矩阵。文献[122]提出了一种基于深度生成模型和模型驱动网络的方案，以较低的计算复杂度实现接近最优性能的混合预编码。文献[123]设计了基于流形优化混合预编码算法的模型驱动网络，实现了较快的收敛速度。

1.3.1.3 波束选择

文献[124-132]研究了带有DLA的毫米波MU-MIMO系统。DLA一般由两部分组成：电磁透镜和匹配的天线阵列。它将一般的空间域MIMO信道转换为聚焦相关角度能量的波束空间信道。在实际应用中，由于波束空间信道的稀疏性，只需要选择若干具有较大性能增益的波束。此外，每条射频链路选择一个波束，因此所需的射频链路数可大大减少。此外，可以用波束选择的转换网络替代传统数模混合结构中的移相器网络，从而大大降低了硬件成本和能耗预算。尽管带有DLA的系统具有上述优点，文献[125]已证明，波束选择问题是NP难的，因此如何高效地对该问题进行求解仍然是一个悬而未决的难题。文献[126]提出了基于干扰感知的启发式算法。文献[127]提出了一种PDD优化算法，用于联合优化波束选择和数字预编码矩阵。文献[128]研究了高维信道的多用户场景下的波束选择问题。文献[129]提出采用最大模值波束选择方案和最大信干噪比（Signal-to-Interference-Ratio, SINR）波束选择方案。文献[130]针对大规模MU-MIMO系统提出了能量最大波束选择算法和基于连续干扰消除的预编码方法。

文献[133-141]提出了使用深度学习算法来求解波束选择问题。具体而言，文献[133]将波束选择问题建模为可以由支持向量机求解的多类分类问题。文献[134]设计了一个用于研究移动用户和基站通信过程中的波束选择问题的数据集。文献[135]通过充分利用AoA信息，设计了多层感知器来优化波束选择矩阵。文献[136]根据超分辨率技术，提出了一种基于深度学习的低复杂度波束选择方案，且该网络可根据部分波束观测样本评估波束质量。文献[137]为了降低网络对数据集大小的要求，设计了基于迁移学习技

术的深度学习波束选择网络。文献[138]提出了一种基于用户位置和方向的波束选择方案,以实现基于前后时刻信息的波束对齐。文献[139]提出了一种基于深度学习的视觉辅助波束选择用于室内用户通信,所提出的方案旨在降低多用户场景下毫米波窄波束选择问题的开销。文献[140]研究了基于图神经网络的联合波束选择和链路激活问题,以有效地控制通信信号间的干扰。针对波束选择网络的训练过程中存在的隐私问题,文献[141]提出了一种隐私保护分割学习方案,在训练和推理过程中可以不上传原始数据从而保护用户隐私。

1.3.2 物理层信道估计

在通信信号的传输过程中,信道会导致发送信号幅度和相位的失真。为正确解码发送信号,接收机需消除由信道引起的信号失真。因此,对信道系数或CSI进行估计的“信道估计”领域显得尤为重要。物理层收发机设计中的其他环节,例如:信号检测^[75]和预编码矩阵设计等^[76-78],都依赖于信道估计获取的CSI。因此,信道估计算法的性能会影响物理层收发机的整体性能。根据输入的数据类型,已有的信道估计算法可分为时域和频域两大类。时域算法适用于大多数通信系统^[142],而频域算法一般应用于多载波正交频分复用(Orthogonal Frequency Division Multiplexing, OFDM)系统^[143]。针对多载波MU-MIMO场景,文献[144]根据不同频率间信道参数的相关性,设计了基于大规模矩阵运算的信道估计算法。另外,根据信道估计算法依赖的先验信息,可进行如下分类:

- 基于导频序列的信道估计^[143-145]:接收端根据传输的的导频序列估计信道,所需的导频数量被称为信道训练开销。该方法应用范围广泛,适用于突发传输和连续传输。
- 盲信道估计^[146-148]:通过分析调制信号的结构特征和统计特性来估计CSI。由于无需导频序列,能显著降低信道训练开销,提升系统的频谱效率。但其性能通常不如基于导频序列的信道估计算法,且容易出现误差传播、相位模糊和收敛慢等问题,因此使用场景较为受限。
- 半盲信道估计^[149-151]:结合上述两种方法的特点,通过发送较短的导频序列估计信道,在信道估计的性能和开销之间达到较好的平衡。

近年来,随着无线通信技术的发展和大规模MIMO系统的部署,新型通信场景不断涌现。在大规模MIMO系统中,天线阵列带来的空间分辨率提高,提升了系统的性能和频谱效率^[72]。为获得该巨大的性能增益,基站需要获得准确的CSI以进行预编码、资源分配等模块的设计。然而,由于信道矩阵的维度较大,获取准确CSI需要的开销过高。为解决上

述问题, 文献[152]针对时分双工 (Time Division Duplex, TDD) 系统, 利用上下行信道的互易性, 通过估计上行CSI获取下行CSI。针对FDD系统, 文献[153]提出了一种基于开环和闭环训练的信道估计算法, 显著降低了信道训练开销。当收发机存在由器件非理想导致的信号畸变时, 文献[154]针对上行和下行系统分别设计了信道估计算法, 并推导了性能界。

此外, 为进一步降低信道估计开销、提升算法效率, 需充分利用信道的稀疏性。在毫米波大规模MIMO系统中, 由于无线传播环境中散射体的数量有限, 在离散傅里叶变换 (Discrete Fourier Transformation, DFT) 基底, 毫米波波束域具有稀疏性特征, 因此大规模MIMO信道具有近似稀疏的表示^[155-157]。利用这种稀疏性, 有许多高效的信道估计算法^[158-160]。其中, 压缩感知相关算法已被广泛应用于求解信道估计问题, 例如: 近似消息传递 (Approximate Message Passing, AMP) 算法^[161]和正交匹配追踪 (Orthogonal Matching Pursuit, OMP) 算法^[162]等。然而, 利用信道在DFT基底具有稀疏性的信道估计算法往往存在能量泄漏问题, 从而会造成一定的性能损失^[163]。本质上, DFT基底提供了一种角度域信道的离散固定采样网格点。由于信号的方向往往是随机的, 因信号方向和网格点的失配造成的能量泄露难以避免。为了减少能量泄露, 文献[163]考虑了一种过完备的DFT基底, 它提供了角度域信道更为密集的采样网格点。然而, 当网格点的密度不够时, 过完备的DFT基底仍会导致严重的方向失配。而如果采用非常密集的采样网格点, 不同基向量将会高度相关, 从而导致算法的性能下降。为了解决上述问题, 文献[164]提出了用于表征稀疏信道的离网模型, 而文献[165]基于该离网模型提出了稀疏贝叶斯学习 (Sparse Bayesian Learning, SBL) 算法。

由于深度学习具有较强的拟合能力和处理大量数据的能力, 有研究将黑盒网络^[166-171]和模型驱动网络^[172-176]应用于信道估计领域。具体而言, 文献[166]和文献[167]设计了CNN用于联合信道估计和信号检测。在TDD大规模MIMO系统中, 文献[168]提出了基于CNN的信道估计方案来缓解导频污染的影响。文献[169]利用DNN高效学习信道的统计特性和空间结构。针对无小区毫米波大规模MIMO系统, 文献[170]将毫米波信道矩阵视为图像, 设计了基于去噪CNN的信道估计方案。文献[171]使用深度残差网络来隐式地学习信道估计的噪声特性, 从带有噪声信道观测样本中恢复CSI。在模型驱动网络方面的研究, 文献[172]提出了一种基于AMP的模型驱动网络, 用于MIMO系统的信道估计, 文献[173]设计了用于波束空间信道估计的模型驱动网络, 文献[174]设计了用于联合信道估计和活跃用户检测的模型驱动网络。文献[175]提出的模型驱动网络, 无需发送下行导频仍能取得较好的信道估计性能。文献[176]设计的模型驱动网络利用角度时延域信道的稀疏性来降低系统开销。

1.3.3 物理层信道反馈

目前文献中的信道反馈方案主要分为两类：(1) 基于压缩感知的，利用CSI的空间或时间相关性的信道反馈方案^[177-179]；(2) 基于码本的信道反馈方案^[180-183]。基于压缩感知的方案充分利用信道矩阵在某些域，例如空时域、空频域的稀疏性，降低信道反馈的开销^[177]。在大规模MIMO系统中，由于散射簇的数量远小于发送天线的数量，通过DFT可将空间域信道转化为稀疏的角度域的信道，从而可用更少的参数来表征信道矩阵。然而，基于压缩感知的信道反馈需要以信道稀疏性假设为前提，这在实际系统中未必成立，且迭代算法的复杂度太高，无法满足实时性要求。基于码本的信道反馈核心思想：假定信道反馈比特数为 B ，收发端共享的信道反馈码本则包含 2^B 个码字，接收端将所选码字对应的标号反馈给发送端，发送端根据该标号恢复出对应的码字，从而恢复出信道矩阵^[182]。随着 B 的增大，信道反馈精度增加，但码本维度随之增加，从而码字搜索的计算复杂度也会明显增加。虽然已提出许多高效的自适应码本信道反馈算法，但反馈精度、码本复杂度和信道反馈开销等仍有待改进。

由于深度学习在特征提取和信息压缩等方面实现了较好的性能，因此可以用于设计高效的信道反馈方案。为更高效地进行信道压缩，文献^[184-190]提出了新颖的信道反馈网络结构，根据信道特性设计了更具有针对性的网络。具体来说，文献^[184]基于CNN的结构增加了卷积核感受野的范围，文献^[185]提出了针对不同信道特征采用多分辨率网络结构，文献^[186]借鉴了CNN的结构并设计了对称的卷积模块，文献^[187]引入了注意力机制并采用了两层ViT结构，文献^[188]针对网络的预处理模块进行了设计，文献^[189]和文献^[190]分别设计了用于信道反馈的对抗生成网络和变分自编码器。

在结构新颖的网络基础上，为充分发挥深度学习的特征提取优势并降低信道反馈开销，需充分利用信道之间的相关性。由此，基于深度学习的信道反馈可以大体分为四类，分别利用不同域的信道相关性：时域相关性^[191-193]、信道互易性^[194-196]、频域相关性^[197-199]、邻近用户相关性^[200]。具体而言，文献^[192]采用LSTM，利用相邻帧信道的时域相关性设计了不同的压缩率，针对不同帧的信道矩阵进行压缩。文献^[194]利用信道互易性，根据上下行信道的幅度和相位特性，分别设计反馈网络。文献^[198]设计了双向LSTM模块，用于提取邻近子载波对应信道的相关信息，从而进一步压缩信道，降低反馈开销。文献^[200]设计了联合提取并反馈邻近用户信道幅度和相位信息的网络，并在接收端设计了解码器用于恢复相邻用户的信道。

此外，为提升深度学习方案的可解释性和泛化性，文献^[201-203]提出了基于模型驱动网络的信道反馈方案。具体而言，文献^[201]展开快速迭代收缩阈值算法，并设计了一

种基于注意机制的收缩阈值网络，能够自适应地选择阈值，以实现更高效的信道反馈。文献[202]利用信道的时延相关性，基于马尔可夫模型和CNN，设计了马尔可夫网络，以提高网络学习精度和信道反馈效率。文献[203]提出了一种自信息模型，从结构特征角度动态度量每个信道矩阵所包含的信息量，并利用该自信息模型设计了一种模型驱动的信道压缩和反馈网络。

1.3.4 新一代物理层通信技术和场景

新一代物理层通信系统包括多种多样的通信技术和通信场景。除了前文所述的预编码、信道估计、信道反馈，还有大规模MIMO系统下的干扰消除、信号检测、信道编码等通信技术，而新兴的通信场景则包括无人机通信、空天地协同通信、智能反射面、联合通信感知一体化、语义通信等。而新颖的模型驱动网络和深度强化学习是解决这些场景下相关问题的重要工具。相比一般黑盒网络，模型驱动网络可解释性、泛化性、可靠性更强，需要的训练时间和样本更少、复杂度更低、更容易训练，因此针对上述新一代通信技术和复杂的通信场景有着广阔的应用前景^[204-211]。针对新一代物理层通信系统中的通信技术，文献[204]设计了基于投影梯度算法的模型驱动网络用于解决大规模MIMO系统中的信号检测问题，文献[205]提出了一种基于消息传递的模型驱动网络用于求解稀疏线性逆问题，文献[206]提出了基于维特比算法的模型驱动网络用于符号检测，文献[207]设计了用于干扰消除的模型驱动网络，文献[208]提出了一种基于模型驱动网络的极性码混合解码器用于高效信道编码。另外，针对新一代物理层通信系统中的新兴通信场景，文献[209]设计了模型驱动网络用于智能反射面的信道建模和估计，文献[210]将模型驱动网络用于无人机通信场景，文献[211]调研了将模型驱动网络用于联合通信感知一体化的研究。

不同于上述网络结构一般用于求解带有连续变量的问题，深度强化学习被广泛应用于求解通信系统中具有离散变量的混合整数非线性规划（Mixed Integer NonLinear Programming, MINLP）问题、高维非凸优化问题、NP难问题，以及可建模成马尔可夫过程的时间序列问题^[212-218]。基于深度神经网络设计的深度Q网络（Deep Q-Network, DQN）已广泛应用于动态信道接入^[212]，无线接入^[213]，无人机通信系统^[214]，智能反射面^[215]，联合通信感知一体化^[211]，资源分配和移动边缘计算^[216-218]。

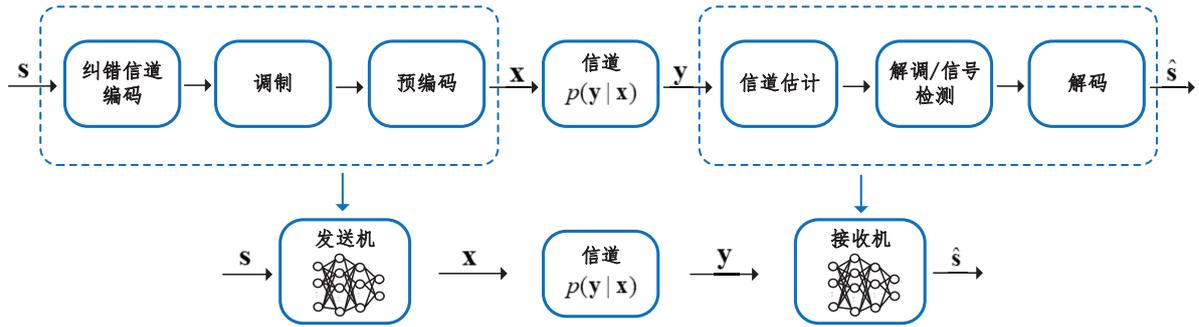


图 1.11 端到端物理层收发机

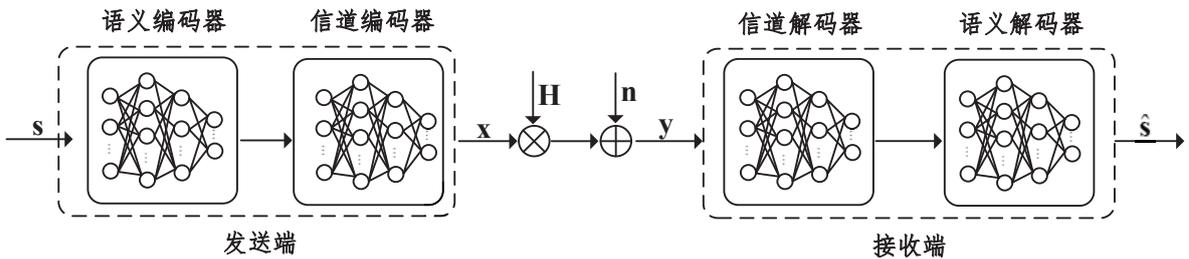


图 1.12 端到端语义通信系统

1.3.5 端到端收发机设计

1.3.5.1 端到端物理层收发机设计

图1.11展示了端到端物理层收发机设计，其核心思想是采用DNN，将物理层通信系统中的信道编码、调制、预编码、信道估计、信道反馈、解调、信号检测等模块进行联合设计，从而打破传统物理层通信系统中模块与模块之间相互独立、分别设计的局限性^[219-227]。为使频分双工系统的和速率最大化，文献[219]联合设计了多用户场景下的信道反馈方案和数字预编码矩阵。文献[220]提出了联合信道编码和星座图设计。文献[221]设计了基于对抗生成网络的端到端无线通信系统，可用于未知信道场景下的通信信号传输。文献[222]和文献[223]端到端联合设计了物理层通信中的信道估计和信道反馈模块，文献[224]和文献[225]端到端联合设计了物理层通信中的信道反馈和预编码模块，文献[226]和文献[227]端到端联合设计了物理层通信中的信道估计、信道反馈和预编码模块。

1.3.5.2 端到端语义通信系统

上述端到端通信系统设计是针对信道进行的联合信道编码、信道估计、信道反馈和预编码等模块的端到端联合设计。而语义通信则可被视为是一种针对信源信道联合编码

的,用于提取和传输信源数据中与具体待执行任务相关特征的关键技术^[228]。如图1.12所示,语义通信系统包含语义编码器、信道编码器、语义解码器、信道解码器等模块。不同于传统通信系统将原始信源数据映射为相互独立的比特进行传输,语义通信系统考虑传输语义信息而不仅仅是原始数据的比特流,从而提高传输的准确性和效率^[229-231]。例如,在用于目标检测任务的图像传输系统中,图像中目标的位置和形状与该任务相关,而图像中的背景与该任务无关,因此无需传输图像背景相关的特征和信息。此外,语义通信对信道环境较差的低SNR场景具有较强的鲁棒性,适用于对可靠性要求较高的应用^[232]。

现有的语义通信研究可大致分为两类:用于数据重构^[232-237]和任务执行^[238-242]的语义通信系统。对于数据重构语义通信系统,其首先提取原始信源数据的语义信息,随后根据接收到的语义信息重构任务相关的数据。文献[232]提出的联合信源信道编码方案,从文本中提取语义信息进行传输,并在接收端进行文本重建,而文献[234]利用DRL来恢复文本。文献[235]提出基于信道反馈的语义通信系统,可提升图像重建的质量。文献[236]设计了一种基于注意力机制的语义通信系统,用于语音信号的传输。而对于特定的任务和应用场景,语义通信系统在发送端提取和编码与任务相关的语义信息,并将这些传输到接收端用于执行任务。文献[238]中面向图像分类任务的语义通信可以较低的传输开销实现较高的分类准确率,而文献[239]针对行人重识别任务设计了具有较高检索精度的语义通信系统。文献[240]考虑了边缘推理和分类任务。文献[241]中的语义通信系统是针对多模态数据中的图像问答任务而设计的。

虽然端到端深度学习的系统实现了较好的性能,其鲁棒性和稳定性往往较差,容易受到噪声的干扰。输入数据样本的微小扰动会导致严重的性能恶化,并导致不正确的输出结果。针对上述系统的干扰噪声大致可以分为两类:(1)特殊性噪声,该噪声针对每个样本单独设计,只能对某一个样本产生干扰,具有一定的特殊性^[243-246];(2)普遍性噪声,该噪声可以较高的概率同时对一组样本产生干扰,具有一定的普遍性^[247-249]。其中,常用的产生第一类噪声的方法包括快速梯度符号法(Fast Gradient Sign Method, FGSM)^[243]、投影梯度下降法(Projected Gradient Descent, PGD)^[244]等。为解决该问题,研究者提出了一些可用于提升端到端深度学习系统鲁棒性的方法,例如输入去噪法^[250]、防御蒸馏法^[251]、梯度正则化^[252]、权重扰动^[253]和对抗训练^[254]等。

1.4 本论文研究动机、主要创新点和内容安排

在下一代移动通信中,网络的规模、用户的传输数据量显著增大,对算法实时性和效率等提出了更高的要求。而通信系统中传统优化算法虽然可以取得较好的性能,但仍有诸

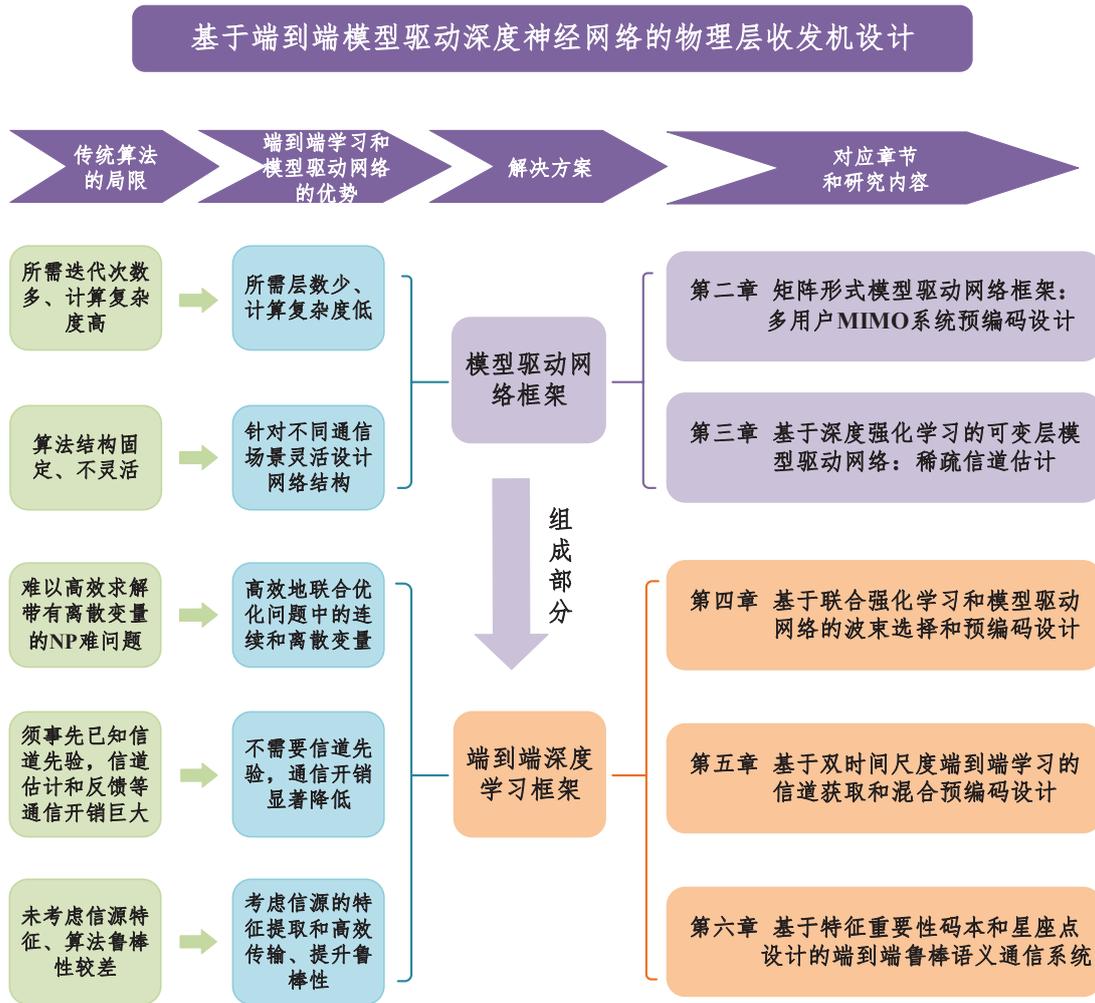


图 1.13 论文框架示意图

多不足：计算复杂度高、需要对问题进行精确的数学建模、鲁棒性较差、通信模块之间的割裂等。深度学习的出现为上述问题提供了一种解决方案。相比传统迭代优化算法，深度学习的计算复杂度较低、不需要对问题进行精确的数学建模、对抗误差的鲁棒性较好、各通信模块可联合设计。论文主要研究了基于端到端模型驱动深度神经网络的物理层收发机设计。该领域仍处于起步阶段，有较好的发展前景。

1.4.1 主要研究结论及创新点

图1.13描述了本文的结构及各章之间的内在联系，以及各章解决的问题。论文结合受到广泛关注的无线通信技术和场景，包括毫米波通信、大规模MU-MIMO、离散透镜天线、语义通信等，研究了物理层收发机设计中至关重要的若干问题：数字模拟预编码设计、波束选择、信道估计、信道反馈、鲁棒传输、星座点设计等。上述问题在下一代移动

通信系统的设计中将面临全新的挑战，例如：传统算法计算复杂度高、结构不灵活、性能不佳、难以优化离散变量、鲁棒性较差、传输时延和通信开销大等。面对上述挑战，本文将探究如何更好地结合无线通信领域的专业知识，利用端到端学习、模型驱动网络、强化学习等先进的深度学习方法，解决物理层收发机设计的关键问题。本文的主要研究结论及创新点总结如下：

1. 矩阵形式模型驱动网络框架：物理层收发机预编码设计

针对无线通信物理层收发机设计，提出了矩阵形式的模型驱动网络框架，以及用于网络训练的广义链式法则（**Generalized Chain Rule, GCR**）。由于已有的FNN和CNN等网络架构并不是针对物理层收发机设计的，且可解释性和泛化性能较为欠缺，所以需要结合已有无线通信中迭代优化算法的模型驱动网络架构，针对物理层收发机中的问题进行更为高效的设计。而提出的模型驱动网络往往包含复数矩阵求逆、行列式等操作，已有的深度学习平台难以支撑。另外，深度学习平台一般采用结合数值梯度的网络训练方法，不够精确且训练效率低。而所提出的广义链式法则可以推导出模型驱动网络中每个矩阵训练参数的精确闭式解，训练效率和精度都更高。随后，将该框架用于预编码设计，根据迭代WMMSE算法设计了模型驱动网络，提出了矩阵的非线性函数，并结合矩阵一阶泰勒展开形式替换高复杂度的矩阵求逆操作。此外，根据GCR推导了网络训练参数的梯度闭式表达式。在保证性能损失很小的前提下，可以显著降低计算复杂度。此框架为物理层收发机的设计提供了新思路。

2. 基于深度强化学习的可变层模型驱动网络框架：稀疏信道估计

针对前一章提出的模型驱动网络框架，为解决其存在的针对不同输入样本，网络层数相同的问题，提出了基于DRL的可变层模型驱动网络框架。即模型驱动网络的层数根据不同的输入样本有所变化，具有自适应性。具体而言，由于一个数据集的不同样本之间存在差异性，当处理较为简单的输入样本时，采用较少的层数即可达到较好满意的性能。而处理较为困难的输入样本时，采用较多的网络层数才能达到满意的性能。提出的框架采用将模型驱动网络的变量、参数、结构分别设计为DRL的状态、动作、状态转移，从而避免由于模型驱动网络层数过多而导致的梯度爆炸现象，使网络训练过程更为高效。另外，通过设计奖励函数和停止分数，来实现针对不同样本的可变层模型驱动网络。随后，将该框架用于展开基于SBL的迭代算法，并设计了用于求解稀疏信道估计问题的可变层模型驱动网络。该框架实现了针对不同样本的动态网络层数分配，在显著减少网络平均层数的前提下，实现了优于固定

层网络的性能。此框架为针对物理层收发机的可变层模型驱动网络设计提供了新思路。

3. 基于端到端联合强化学习和模型驱动网络的波束选择收发机设计

拓展了所提出的模型驱动网络框架，与其他DNN进行联合设计，以求解更为复杂的收发机设计问题。为解决带耦合约束的MINLP问题，提出了联合DRL网络和模型驱动网络框架，分别用于优化离散变量和连续变量。随后，将该框架用于求解带有DLA的毫米波MU-MIMO系统中的波束选择问题。由于存在耦合约束和离散变量，一般的网络难以直接进行处理，因此需要对联合网络架构进行特殊设计以保证约束满足。此外，在高效处理离散变量的同时，需保证网络训练过程顺利进行，避免梯度中断。具体而言，DRL网络将波束选择问题建模为MDP，通过对其状态、动作和奖励函数进行设计，使得在满足波束、功率等相关约束的前提下，选择能够提升用户性能的波束。而针对数字预编码矩阵的设计，设计了模型驱动网络，并引入训练参数替换高复杂度操作。由于两个模块之间相互耦合，而两个网络在训练过程中的梯度并不能直接相互传递，因此提出了一种新颖的训练方式联合交替训练两个网络，且该联合网络框架性能优于现有迭代算法和不同模块分别设计的算法。此框架为物理层收发机设计中包含多个模块的复杂MINLP问题的高效求解提供了新方案。

4. 基于端到端双时间尺度学习的信道获取和数模混合收发机设计

针对现有毫米波大规模MIMO系统的收发机模块分开训练的高复杂度和性能瓶颈问题，提出了一种基于端到端深度学习的联合收发机网络设计框架。设计了信道估计、信道反馈和数模混合预编码等模块相关的深度神经网络，并进行联合训练，以打破收发机各模块之间的壁垒，从而突破性能瓶颈。为进一步降低信令开销和由传输延迟导致的信道失配，基于上述框架设计了一种双时间尺度网络架构，其由长时间尺度网络和短时间尺度网络组成。具体而言，长时间尺度网络估计并反馈高维原始信道，并在每帧更新一次模拟预编码矩阵。短时间尺度网络估计并反馈低维等效信道，并在每个时隙更新一次数字预编码矩阵。由于高维原始信道的估计和反馈频率远低于低维等效信道，同时高维模拟预编码矩阵的更新频率远低于低维数字预编码矩阵，通信系统的复杂度和信令开销可以大大降低。此外，由于该网络架构存在信道反馈模块，因此包含二值层，为了避免梯度中断从而更高效地训练网络，同时提升网络的泛化性能，提出了两阶段训练方法。该方案在显著降低信道估计和反馈开销，同时缩短导频序列的情况下，可提升系统误码率性能，为数模混合收发机的联合设计提供了新思路。

5. 基于特征重要性码本和星座点设计的端到端鲁棒语义通信系统设计

基于前述章节针对信道因素的物理层收发机设计, 进一步考虑信源信道联合编码问题和信源传输过程中的系统鲁棒性问题。首先设计了端到端联合信源信道编码的图像传输收发机, 而在传输过程中存在一种语义噪声, 不易被人眼发现但会对系统性能造成极大的影响。根据不同场景, 对语义通信系统中的样本依赖和非样本依赖语义噪声进行了建模。随后, 提出了一种鲁棒的端到端语义通信系统框架来对抗语义噪声, 从而提升图像传输系统的鲁棒性。具体而言, 提出了基于权值扰动对抗训练方式, 将含有语义噪声的样本纳入训练数据集。接着, 提出对语义噪声出现概率较高的输入图像块进行掩蔽, 并根据该掩蔽策略设计掩蔽矢量量化变分自编码器 (Vector Quantized-Variational AutoEncoder, VQ-VAE)。此外, 提出由发送端和接收端共享的离散码本来对图像特征进行编码, 并将其和星座图建立映射关系。通过对信源的矢量量化编码和星座点联合设计, 更好地适配实际数字通信系统同时提升系统的传输效率。为了进一步提升系统的鲁棒性, 提出了特征重要性模块 (Feature Importance Module, FIM) 来抑制噪声相关的且和图像任务无关的特征。因此, 发送端只需要传输码本中与图像任务相关的重要特征的标号索引。该系统架构可应用于多种不同的图像任务, 显著提高系统的鲁棒性, 并降低传输开销。

1.4.2 内容安排

值得一提的是, 本文提出的技术并非深度学习在物理层收发机设计的简单应用。而是基于如下核心思想进行的实际系统创新型设计: 直面下一代通信系统对高可靠、高速率和低延迟等方面更为严格的要求, 利用深度学习强大的拟合和学习能力, 结合物理层收发机具体问题的结构特征, 针对存在的技术瓶颈, 设计高效算法, 降低通信开销和计算复杂度, 同时改善系统性能。本论文的结构如图1.13所示。全文共分为七章, 其中第二章至第六章为主要研究内容, 分别对应物理层收发机设计的五种不同场景。具体而言, 第二章和第三章设计了模型驱动网络分别用于物理层收发机中的预编码矩阵设计和信道估计。第四章和第五章设计了端到端深度学习网络架构, 将模型驱动网络融入端到端学习框架, 结合DRL网络、自编码器等网络结构联合设计了多个物理层收发机模块, 用于物理层收发机中更为复杂的波束选择、混合预编码、信道估计和反馈等模块的联合设计。第六章根据前述章节的物理层收发机设计, 设计了基于信源特征提取和星座图设计的鲁棒语义通信系统, 进一步提升了收发机的性能。本文的第二章至第六章均来源于作者本人已发表的论文。

第一章为绪论，介绍了论文的研究背景、端到端深度学习和模型驱动网络的关键技术和挑战，以及已有相关研究的综述，并给出了论文的研究内容和结构。

第二章针对现有的无线通信系统中迭代优化算法复杂度高的问题，提出了一种矩阵形式的模型驱动网络框架，以更好地解决通信系统中的问题。具体而言，基于优化算法的迭代结构，在前向传播中引入训练参数替代其中原有的高复杂度操作。针对网络训练的反向传播，提出了GCR，用于推导训练参数在训练过程中的闭式梯度，从而提升网络训练效率。随后，将所提出的模型驱动网络框架用于解决MU-MIMO系统中预编码设计的和速率最大化问题。在经典迭代WMMSE算法的基础上，设计了一种高效的迭代算法驱动的深度展开网络（Iterative Algorithm Induced Deep-unfolding Neural Network, IAIDNN）。具体而言，迭代WMMSE算法被展开为类似于DNN的逐层结构，并引入训练参数来替代其中的高复杂度矩阵求逆操作。为了更高效地训练网络，采用了GCR来描述反向传播过程中网络相邻两层间梯度的递推关系，从而得到每一层网络训练参数的梯度闭式表达式。此外，还讨论了该方案的计算复杂度和泛化能力。仿真结果表明，所提出的IAIDNN在降低计算复杂度的前提下，能够实现逼近迭代WMMSE算法的性能。

第三章为解决模型驱动网络对不同输入样本的层数相同的问题，提出了一种基于DRL的模型驱动网络框架。该框架实现了对不同输入样本的可变层模型驱动网络，其中模型驱动网络的训练参数由DRL学习得到，而不是直接由SGD更新。具体而言，模型驱动网络的优化变量、训练参数和层结构被分别设计为DRL的状态、动作和状态转移。随后，将该框架应用于解决MU-MIMO系统的信道估计问题。具体来说，首先建模了稀疏信道估计问题并提出使用带有离网参数的基于SBL的迭代算法求解。随后，在该迭代算法中引入训练参数，并将其展开为具有逐层结构的模型驱动网络。根据该得到的网络结构，采用上述可变层模型驱动网络框架解决该信道估计问题。为了更好地实现针对信道估计问题的可变层网络并使得其估计精度更为可控，设计了停止分数来指示何时网络停止运行并输出最终结果，而该停止分数为关于信道估计误差的函数。在此基础上，对所提出的可变层网络框架进行扩展，实现了一般DNN的可变层。仿真结果表明，该算法能够在减少网络层数的前提下，实现优于固定层网络的性能。

第四章将模型驱动网络与其他DNN相结合，针对无线通信中具有离散变量和耦合约束的MINLP问题，提出了一种高效的联合网络设计框架。具体来说，该框架由DRL网络和模型驱动网络组成，分别用于优化离散变量和连续变量。随后，将该框架用于设计带有DLA的毫米波MU-MIMO系统收发机，以在发射功率约束和波束选择矩阵结构约束下使得系统和速率最大化。具体而言，DRL网络和模型驱动网络分别用于设计波束选择矩阵和数字预编码矩阵。对于DRL网络，波束选择问题被建模为MDP，并采用DDQN架构进行

求解。基站被设计为智能体，通过对其状态、动作和奖励函数的设计，使得在满足约束的前提下，选择能够提升用户性能的波束。而针对数字预编码矩阵的设计，提出了一种基于迭代WMMSE算法的模型驱动网络，将该算法展开为逐层结构，并引入训练参数替换高复杂度操作。仿真结果表明，该联合训练网络具有较低的计算复杂度和较强的鲁棒性，在求解MINLP问题时，显著优于现有迭代算法。

第五章联合设计毫米波大规模MIMO系统的物理层收发机模块，提出了一种基于端到端深度学习的联合收发机网络设计框架。其针对信道估计、信道反馈和混合预编码等模块设计了DNN。该框架在接收端将收到的导频信号映射为信道反馈的信息比特，在发送端进一步将收到的信息比特映射为混合预编码器矩阵。为了进一步降低信令开销和由传输延迟导致的信道失配，基于上述框架设计了一种双时间尺度网络架构，其由长时间尺度网络和短时间尺度网络组成。具体而言，首先划分了帧结构，一个超帧由多个帧组成，而一个帧由多个时隙组成。在一个超帧内信道统计特性不变，而在一个时隙内信道不变。长时间尺度网络估计并反馈高维原始信道，并在每帧更新一次收发端模拟预编码矩阵。短时间尺度网络估计并反馈低维等效信道，并在每个时隙更新一次收发端数字预编码矩阵。该网络架构由于存在信道反馈模块，因此包含二值层，为了避免梯度中断从而更高效地训练网络，同时提升网络的泛化性能，本章提出了一种两阶段训练方法。随后，分析了网络的泛化能力和信令开销。该双时间尺度网络架构可以通过简单地修改训练数据的结构，扩展到多载波OFDM通信系统。仿真结果表明，该方案在显著降低信令开销和缩短导频序列的情况下，在系统误码率性能方面明显优于传统方案。

第六章针对图像传输通信系统存在的鲁棒性问题，提出了一种鲁棒的端到端语义通信系统框架，提升图像传输系统的鲁棒性。具体而言，对语义通信系统中的样本依赖和非样本依赖语义噪声进行了建模。为对抗语义噪声，提出了基于权值扰动对抗训练方式，将含有语义噪声的样本纳入训练数据集。随后，提出对语义噪声出现概率较高的输入图像块进行掩蔽，并根据该掩蔽策略设计掩蔽VQ-VAE。此外，为了更好地适配实际数字通信系统同时提升系统传输效率，设计由发送端和接收端共享的离散码本来对图像特征进行编码，并将其和星座图建立映射关系。为了进一步提升系统的鲁棒性，提出了FIM来抑制噪声相关的且和图像任务无关的特征。因此，发送端只需传输码本中与图像任务相关的重要特征的标号索引。最后，仿真结果表明，该方法可以应用于多种不同的图像任务，显著提高了系统的鲁棒性，同时降低了传输开销。

第七章是全文的总结和展望部分，总结了本论文的研究内容和成果，并对未来的研究方向进行了展望。

2 矩阵形式模型驱动网络框架：大规模MU-MIMO系统预编码设计

2.1 引言

大规模MU-MIMO系统能提供分集增益和复用增益，并显著提高频谱效率，因此在无线通信领域受到广泛关注。为尽可能提升频谱效率，已有针对大规模MU-MIMO系统设计的基于优化理论的迭代预编码算法。尽管这些迭代预编码算法实现了较好的性能，但由于迭代过程中存在大规模矩阵求逆等高复杂度操作，且需要的迭代次数较多，导致计算复杂度非常高。这难以满足通信系统的低时延要求，阻碍了其在实时通信系统中的应用。为解决上述问题，有研究者提出采用深度学习方法，主要思想是将迭代算法视为一个黑盒，利用DNN学习输入和输出之间的映射关系。然而，黑盒网络的可解释性和泛化能力较差，且性能没有理论保证。数据驱动的黑盒网络需要大量的训练样本，导致网络训练时间较长。为了克服这些缺点，有学者提出将迭代优化算法展开为类似于DNN的逐层结构，被称为模型驱动网络。

在无线通信领域，尤其是大规模MU-MIMO系统中，模型驱动网络还未被很好地研究。此外，现有模型驱动网络的设计主要集中在针对标量变量的优化问题上，而大规模MU-MIMO系统中的问题涉及矩阵变量。由此，本章将重点研究以下两个问题：（1）如何设计一个针对大规模MU-MIMO无线通信系统的通用模型驱动网络框架？（2）如何基于该框架设计针对大规模MU-MIMO系统预编码的高效模型驱动网络？

针对问题一，提出了通用的矩阵形式模型驱动网络框架，针对性地解决物理层通信中的问题。具体而言，在迭代优化算法的基础上，设计了模型驱动网络结构，用于网络的前向传播，并引入训练参数来替换原迭代算法中的高复杂度操作。在训练参数梯度反向传播过程中，提出用于训练矩阵形式模型驱动网络的GCR，其描述了相邻两层网络间训练参数梯度的递推关系。由此，可以方便地推导出网络各层训练参数的梯度闭式表达式。该GCR扩展了一般的黑盒网络标量形式的链式法则，可以证明现有的标量链式法则是本章所提出GCR的一种特殊情况。

针对问题二，基于该模型驱动网络框架和经典迭代WMMSE算法设计了IAIDNN，¹用于求解大规模MU-MIMO系统中预编码设计的和速率最大化问题。具体而言，通过把功率约束融合到目标函数中，将原带功率约束的优化问题等价地转化为无约束和速率最大化问题，其目标函数可被设计为IAIDNN无监督训练阶段的损失函数。接着，将迭代WMMSE算法展开为具有一系列矩阵乘法和非线性操作的逐层网络结构。一方面，使用更少的迭代次数，即IAIDNN中的层数，来逼近原始迭代WMMSE算法，并避免矩阵求逆操作以降低计算复杂度；另一方面，通过引入训练参数来提升IAIDNN的性能。在前向传播过程中，设计了非线性函数和矩阵求逆的一阶泰勒展开结构来近似高维矩阵求逆运算。在反向传播过程中，采用所提出的GCR来计算训练参数的梯度，并使用SGD来更新参数。此外，提出使用黑盒网络作为基线算法，并讨论了所提出IAIDNN、黑盒网络和迭代WMMSE算法各自的计算复杂度和泛化能力。

本章内容安排如下。2.2节提出了一种通用的基于矩阵形式的模型驱动网络框架，并提出了GCR用于训练网络。2.3节给出了预编码问题的数学建模，并简要介绍了用于频谱效率最大化的经典迭代WMMSE预编码设计算法。2.4节设计了基于迭代WMMSE算法的模型驱动网络。2.5节介绍了一种基于黑盒的深度学习网络基线算法，并分析了模型驱动网络的计算复杂性和泛化能力。2.6节给出了仿真结果并验证所提算法的有效性。最后，2.7对本章进行了总结。

2.2 模型驱动网络框架

本节提出了通用矩阵形式的模型驱动网络框架，并推导了矩阵形式的反向传播链式法则，用于计算训练参数的梯度，便于高效训练网络。

2.2.1 优化问题的一般形式

优化问题的一般形式

$$\min_{\mathbf{X}} f(\mathbf{X}; \mathbf{Z}) \quad \text{s.t.} \quad \mathbf{X} \in \mathcal{X}, \quad (2.1)$$

其中 $f: \mathbb{C}^{m \times n} \mapsto \mathbb{R}$ 为连续的目标函数， $\mathbf{X} \in \mathbb{C}^{m \times n}$ 为优化变量， \mathcal{X} 是可行域， $\mathbf{Z} \in \mathbb{C}^{p \times q}$ 为问题的已知参数。问题(2.1)可以使用迭代算法进行求解，迭代算法的一般形式

$$\mathbf{X}^t = F_t(\mathbf{X}^{t-1}; \mathbf{Z}), \quad (2.2)$$

¹IAIDNN即模型驱动网络，为和已发表论文保持一致，故保留IAIDNN的说法。

其中 $t \in \mathcal{T} \triangleq \{1, 2, \dots, T\}$ 表示迭代次数, T 为最大迭代次数, 函数 F_t 表示在第 t 次迭代, 基于问题参数 \mathbf{Z} 将变量 \mathbf{X}^{t-1} 映射为 \mathbf{X}^t 。

2.2.2 模型驱动网络的前向传播

根据迭代算法的一般表达式(2.2), 引入训练参数 $\boldsymbol{\theta} \in \mathbb{C}^{a \times b}$ 来降低迭代算法的复杂度, 同时提升其性能。因为 \mathbf{Z} 通常为随机变量, 通过对 \mathbf{Z} 求期望, 问题(2.1)可表示为

$$\min_{\mathbf{X}} \mathbb{E}_{\mathbf{Z}} \{f(\mathbf{X}; \boldsymbol{\theta}, \mathbf{Z})\} \quad \text{s.t.} \quad \mathbf{X} \in \mathcal{X}. \quad (2.3)$$

随后, 迭代表达式(2.2)可以被转换成如下网络形式:

$$\mathbf{X}^l = \mathcal{F}_l(\mathbf{X}^{l-1}; \boldsymbol{\theta}^l, \mathbf{Z}), \quad (2.4)$$

其中 $l \in \mathcal{L} \triangleq \{1, 2, \dots, L\}$ 表示网络层数, L 表示网络的总层数, \mathcal{F}_l 代表第 l 层的网络结构, \mathbf{X}^{l-1} 和 \mathbf{X}^l 分别表示第 l 层的输入和输出, \mathbf{Z} 为已知问题的参数或者网络的输入, $\boldsymbol{\theta}^l$ 表示第 l 层的训练参数。此外, 优化问题(2.1) 中的 $f(\mathbf{X}; \mathbf{Z})$ 可以作为网络(2.4)的损失函数。最后, 将网络的输出 \mathbf{X}^L 代入目标函数 $f(\mathbf{X}; \mathbf{Z})$ 以获得最终结果。

2.2.3 模型驱动网络的反向传播

为训练学习参数 $\boldsymbol{\theta}^l$, 需计算目标函数 $f(\mathbf{X}; \boldsymbol{\theta}, \mathbf{Z})$ 关于 $\boldsymbol{\theta}^l$ 的梯度, 并使用链式法则计算训练参数 $\boldsymbol{\theta}^l$ 的梯度, 通过反向传播来更新训练参数 $\boldsymbol{\theta}^l$ 。为了描述相邻层训练参数之间梯度的递推关系, 定理2.1提出了矩阵形式GCR。注2.1比较了提出的矩阵形式GCR和现有标量形式链式法则, 其中后者为前者的一种特殊情况。

定理 2.1. (矩阵形式GCR) 考虑一般形式的模型驱动网络结构(2.4): $\mathbf{X}^l = \mathcal{F}_l(\mathbf{X}^{l-1}; \boldsymbol{\theta}^l, \mathbf{Z})$ 。从第 l 层变量 \mathbf{X}^l 到第 $l-1$ 层变量 \mathbf{X}^{l-1} 之间的递推关系可以描述为

$$\text{Tr}(\mathbf{G}^l d\mathbf{X}^l) = \text{Tr} \left(\mathbf{G}^l \mathbf{A}^l (\mathbf{B}^l \odot (\mathbf{C}^l d\mathbf{X}^{l-1} \mathbf{E}^l)) \mathbf{F}^l \right) = \text{Tr} \left(\mathbf{E}^l ((\mathbf{F}^l \mathbf{G}^l \mathbf{A}^l) \odot (\mathbf{B}^l)^T) \mathbf{C}^l d\mathbf{X}^{l-1} \right), \quad (2.5)$$

其中 \mathbf{G}^l 和 \mathbf{G}^{l-1} 分别为变量 \mathbf{X}^l 和 \mathbf{X}^{l-1} 的梯度, 其他矩阵例如 \mathbf{A}^l , 与网络的结构相关, 比如 $\boldsymbol{\theta}^l$ 和 \mathbf{Z} 。DNN的结构常包含逐元素的非线性激活函数, 因此引入 \odot 表示矩阵逐元素相乘, 即Hadamard积。由此得到矩阵形式GCR

$$\mathbf{G}^{l-1} = \mathbf{E}^l ((\mathbf{F}^l \mathbf{G}^l \mathbf{A}^l) \odot (\mathbf{B}^l)^T) \mathbf{C}^l. \quad (2.6)$$

证明。见附录A.1.1。 \square

注 2.1. (矩阵形式GCR和标量形式链式法则的对比) 一般FNN的结构: $\mathbf{x}^l = \varphi(\mathbf{z}^l)$, 其中 $\mathbf{z}^l = \mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^l$, \mathbf{W}^l 和 \mathbf{b}^l 分别代表第 l 层的权重矩阵和偏置矩阵, \mathbf{z}^l 和 \mathbf{x}^l 分别表示第 l 层非线性函数 φ 的输入和输出, φ 表示逐元素非线性函数。因此, 标量形式链式法则为

$$\mathbf{g}^{l-1} = ((\mathbf{W}^l)^T \mathbf{g}^l) \odot \varphi'(\mathbf{z}^l), \quad (2.7)$$

其中 \mathbf{g}^l 表示第 l 层变量 \mathbf{x}^l 的梯度。通过比较(2.6)和(2.7), 令 $\mathbf{F} = \mathbf{W}^T$, $\mathbf{B} = \varphi'(\mathbf{z})^T$, $\mathbf{A} = \mathbf{E} = \mathbf{C} = \mathbf{I}$, 其中 \mathbf{I} 表示单位矩阵, 则(2.6)等价于(2.7)。因此, (2.7)是(2.6)的一个特例。

接着, 为了阐述如何使用定理2.1, 提供一个具有关于变量二次方结构和矩阵逐元素非线性函数的模型驱动网络结构。根据定理2.1中的(2.6), 推导了相邻层之间变量的梯度递推关系。

引理 2.1. 当网络包含如下关于输入变量的二次方 (两个变量 \mathbf{X}^l 相乘) 结构:

$$\mathbf{X}^l = \bar{\mathbf{A}} \mathbf{X}^{l-1} \bar{\mathbf{B}} \mathbf{X}^{l-1} \bar{\mathbf{C}} + \varphi(\bar{\mathbf{A}} \mathbf{X}^{l-1} \bar{\mathbf{B}} \mathbf{X}^{l-1} \bar{\mathbf{C}}) \bar{\mathbf{D}}, \quad (2.8)$$

其中 $\varphi: \mathbb{C}^{m \times n} \mapsto \mathbb{C}^{m \times n}$ 为逐元素非线性函数, \mathbf{X}^l 为第 l 层的输出变量, 其他矩阵, 比如 $\bar{\mathbf{A}}$, 为常数矩阵或训练参数矩阵。为了更清楚地表达, 略去这些矩阵的角标 l 。网络的损失函数为 $f(\mathbf{X})$, $f: \mathbb{C}^{m \times n} \mapsto \mathbb{R}$ 。关于变量 \mathbf{X}^{l-1} 的矩阵微分可表示为

$$\begin{aligned} \text{Tr} \left\{ \mathbf{G}^l d\mathbf{X}^l \right\} \stackrel{(2.5)}{\cong} & \text{Tr} \left\{ \left(\bar{\mathbf{B}} \mathbf{X}^{l-1} \bar{\mathbf{C}} \mathbf{G}^l (\bar{\mathbf{D}} \odot \varphi'(\bar{\mathbf{A}} \mathbf{X}^{l-1} \bar{\mathbf{B}} \mathbf{X}^{l-1} \bar{\mathbf{C}})^T + \mathbf{I}) \bar{\mathbf{A}} \right. \right. \\ & \left. \left. + \bar{\mathbf{C}} \mathbf{G}^l (\bar{\mathbf{D}} \odot \varphi'(\bar{\mathbf{A}} \mathbf{X}^{l-1} \bar{\mathbf{B}} \mathbf{X}^{l-1} \bar{\mathbf{C}})^T + \mathbf{I}) \bar{\mathbf{A}} \mathbf{X}^{l-1} \bar{\mathbf{B}} \right) d\mathbf{X}^{l-1} \right\}, \end{aligned} \quad (2.9)$$

其中 φ' 表示对非线性函数 φ 中的矩阵进行逐元素求导操作, \mathbf{G}^l 表示第 l 层变量 \mathbf{X}^l 的梯度。

二次方结构在迭代优化算法中较为常见, 因此在模型驱动网络中也被广泛使用, 引理2.1中的结果将被用于下一节设计IAIDNN。根据定理2.1和引理2.1, 梯度的反向传播过程总结如下。首先, 网络输出层变量 \mathbf{X}^L 的梯度 \mathbf{G}^L , 是通过损失函数 $f(\mathbf{X})$ 关于变量 \mathbf{X}^L 求梯度得到的。随后, 根据定理2.1中的矩阵形式GCR, 可计算得到每一层的梯度 $\{\mathbf{G}^l, l \in \mathcal{L}\}$ 。最后, 关于训练参数 θ^l 的梯度可以通过 \mathbf{G}^l 计算得到。

注 2.2. 相比于直接使用Tensorflow等深度学习平台训练网络, 本章提出的矩阵形式GCR具有如下优势: (1) 平台不能对复数训练参数做反向传播; (2) 存在一些平台不支持的操作,

例如：对复数矩阵求逆和行列式，在网络训练过程中不能进行梯度反向传播；(3) 基于矩阵形式GCR，可计算出闭式梯度表达式，相比较平台通过自动微分机制求得的梯度更精确，具有更好的性能和更快的收敛速度。

2.3 迭代WMMSE预编码算法

本节将简要介绍经典的迭代WMMSE预编码算法。

2.3.1 问题建模

迭代WMMSE算法是物理层预编码中用于最大化系统和速率的经典高效的算法^[86]。考虑一个MU-MIMO系统，它包含一个装配有 N_t 根发送天线的基站和 K 个用户，每一个用户装配有 $N_{r,k}$ 接收天线，其中 $k \in \mathcal{K} \triangleq \{1, 2, \dots, K\}$ 。用 $\mathbf{V}_k \in \mathbb{C}^{N_t \times d}$ 表示基站用于处理发送给用户 $k \in \mathcal{K}$ 的信号 $\mathbf{s}_k \in \mathbb{C}^{d \times 1}$ 的预编码矩阵，则经过预编码的数据矢量可表示为

$$\mathbf{x} = \sum_{k=1}^K \mathbf{V}_k \mathbf{s}_k. \quad (2.10)$$

假定发送信号矢量 \mathbf{s}_k 的均值为零以及 $\mathbb{E}[\mathbf{s}_k \mathbf{s}_k^H] = \mathbf{I}$ ，不同用户发送的信号是相互独立的。用户 k 的接收信号矢量 $\mathbf{y}_k \in \mathbb{C}^{N_{r,k} \times 1}$ 为：

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{V}_k \mathbf{s}_k + \sum_{m=1, m \neq k}^K \mathbf{H}_k \mathbf{V}_m \mathbf{s}_m + \mathbf{n}_k, \quad \forall k \in \mathcal{K}, \quad (2.11)$$

其中 $\mathbf{H}_k \in \mathbb{C}^{N_{r,k} \times N_t}$ 表示基站到用户 k 的信道矩阵， $\mathbf{n}_k \in \mathbb{C}^{N_{r,k} \times 1}$ 代表均值为零，协方差矩阵为 $\sigma_k^2 \mathbf{I}$ ，服从复高斯分布 $\mathcal{CN}(\mathbf{0}, \sigma_k^2 \mathbf{I})$ 的加性复高斯白噪声， σ_k^2 表示用户 k 的噪声方差。

旨在满足功率约束的前提下，通过优化预编码矩阵来最大化系统中用户的和速率。因此，问题被建模为

$$\max_{\{\mathbf{V}_k\}} \sum_{k=1}^K \omega_k \log \det \left(\mathbf{I} + \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \left(\sum_{m \neq k} \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H + \sigma_k^2 \mathbf{I} \right)^{-1} \right) \quad (2.12a)$$

$$\text{s.t.} \quad \sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \leq P_T, \quad (2.12b)$$

其中 ω_k 代表系统中用户 k 的优先级， P_T 表示基站的总功率，(2.12b)为功率约束。

2.3.2 迭代WMMSE预编码算法设计

2.3.2.1 MMSE问题转化

文献[86]证明了如下MMSE问题(2.13)等价于和速率最大化问题(2.12)，这两个问题的最优解 $\{\mathbf{V}_k\}$ 是相同的，

$$\min_{\{\mathbf{W}_k, \mathbf{U}_k, \mathbf{V}_k\}} \sum_{k=1}^K \omega_k (\text{Tr}(\mathbf{W}_k \mathbf{E}_{1,k}) - \log \det(\mathbf{W}_k)) \quad (2.13a)$$

$$\text{s.t.} \quad \sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \leq P_T, \quad (2.13b)$$

其中 \mathbf{U}_k 和 \mathbf{W}_k 为引入的辅助变量，分别表示用户 k 的接收编码矩阵和权重矩阵，

$$\mathbf{E}_{1,k} \triangleq (\mathbf{I} - \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_k)(\mathbf{I} - \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_k)^H + \sum_{m \neq k} \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H \mathbf{U}_k + \sigma_k^2 \mathbf{U}_k^H \mathbf{U}_k, \quad (2.14)$$

表示用户 k 的MSE矩阵。

2.3.2.2 问题的等价转化

为了便于模型驱动网络的设计和训练，提升数值稳定性，同时满足功率约束，将 $\frac{1}{P_T} \sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H)$ 这一项放入目标函数(2.12a)，并考虑如下无约束的和速率最大化问题

$$\max_{\{\mathbf{V}_k\}} \sum_{k=1}^K \omega_k \log \det \left(\mathbf{I} + \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \left(\sum_{m \neq k} \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H + \frac{\sigma_k^2}{P_T} \sum_{n=1}^K \text{Tr}(\mathbf{V}_n \mathbf{V}_n^H) \mathbf{I} \right)^{-1} \right). \quad (2.15)$$

提出如下引理：

引理 2.2. 原始问题(2.12)的最优解 \mathbf{V}_k^* 和经过转换的无约束问题(2.15)的最优解 \mathbf{V}_k^{**} 满足如下

关系： $\mathbf{V}_k^* = \alpha \mathbf{V}_k^{**}$ ，其中 $\alpha = \frac{\sqrt{P_T}}{\left(\sum_{k=1}^K \text{Tr}(\mathbf{V}_k^{**} (\mathbf{V}_k^{**})^H) \right)^{\frac{1}{2}}}$ 为缩放因子。

证明. 可以看出 \mathbf{V}_k^* 能满足约束(2.12b)。将 $\alpha \mathbf{V}_k^{**}$ 代入(2.15)和功率约束(2.12b)，得到目标函数的最大值并使得功率约束满足。因此， $\alpha \mathbf{V}_k^{**}$ 为问题(2.12)的最优解。□

因此，可通过优化无约束优化问题(2.15)得到的解，作为原问题(2.12)的解。此外，根据问题(2.12)和问题(2.13)之间的关系，问题(2.15)可以等价地转换为如下无约束MMSE问题

$$\min_{\{\mathbf{W}_k, \mathbf{U}_k, \mathbf{V}_k\}} \sum_{k=1}^K \omega_k (\text{Tr}(\mathbf{W}_k \mathbf{E}_{2,k}) - \log \det(\mathbf{W}_k)), \quad (2.16)$$

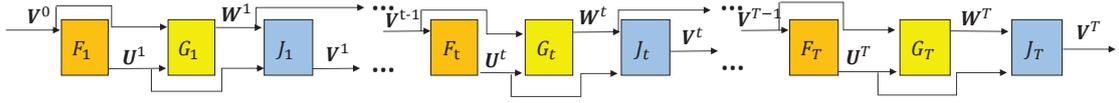


图 2.1 迭代WMMSE算法的流程框图

其中 $\mathbf{E}_{2,k}$ 为

$$\mathbf{E}_{2,k} \triangleq (\mathbf{I} - \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_k)(\mathbf{I} - \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_k)^H + \sum_{m \neq k} \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H \mathbf{U}_k + \frac{\sum_{n=1}^K \text{Tr}(\mathbf{V}_n \mathbf{V}_n^H)}{P_T} \sigma_k^2 \mathbf{U}_k^H \mathbf{U}_k, \quad (2.17)$$

其中 $\mathbf{E}_{2,k}$ 表示经过问题的等价转换之后用户 k 的MSE矩阵，而 $\mathbf{E}_{1,k}$ 则表示原始问题中用户 k 的MSE矩阵。

2.3.2.3 迭代WMMSE算法

根据求解问题(2.13)的迭代WMMSE算法^[86]，提出基于块坐标下降优化方式的迭代算法来求解问题(2.16)，它能保证收敛得到的解为原问题(2.12)的平稳点。算法的详细细节请见表2.1，其中为表述清楚起见，忽略迭代指标 t 。

表 2.1 迭代WMMSE预编码算法

算法 2.1 迭代WMMSE预编码算法

- 1: 输入：信道矩阵 $\mathbf{H}_k, \forall k$;
 - 2: 输出：预编码矩阵 $\mathbf{V}_k, \forall k$;
 - 3: 初始化： $\{\mathbf{V}_k\}$ 使得 $\sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \leq P_T$ 满足；设置迭代精度 ϵ ，最大迭代次数 I_{\max} ，以及当前迭代次数 $t = 0$;
 - 4: **while** 目标函数未收敛且 $t \leq I_{\max}$ **do**
 - 5: 更新 \mathbf{U}_k : $\mathbf{U}_k = \mathbf{A}_k^{-1} \mathbf{H}_k \mathbf{V}_k$ ，其中 $\mathbf{A}_k = \frac{\sigma_k^2}{P_T} \sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \mathbf{I} + \sum_{m=1}^K \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H, \forall k$;
 - 6: 更新 \mathbf{W}_k : $\mathbf{W}_k = \mathbf{E}_k^{-1}$ ，其中 $\mathbf{E}_k = \mathbf{I} - \mathbf{U}_k^H \mathbf{H}_k \mathbf{V}_k, \forall k$;
 - 7: 更新 \mathbf{V}_k : $\mathbf{V}_k = \omega_k \mathbf{B}^{-1} \mathbf{H}_k^H \mathbf{U}_k \mathbf{W}_k$ ，其中 $\mathbf{B} = \sum_{k=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\omega_k \mathbf{U}_k \mathbf{W}_k \mathbf{U}_k^H) \mathbf{I} + \sum_{m=1}^K \omega_m \mathbf{H}_m^H \mathbf{U}_m \mathbf{W}_m \mathbf{U}_m^H \mathbf{H}_m, \forall k$;
 - 8: $t = t + 1$;
 - 9: **end while**
 - 10: 缩放预编码矩阵 $\{\mathbf{V}_k\}$ 使得功率约束满足。
-

对比本章研究的问题(2.16)和一般形式的优化问题(2.1)，可得如下对应关系

$$\mathbf{X} \equiv \{\mathbf{W}_k, \mathbf{U}_k, \mathbf{V}_k, \forall k \in \mathcal{K}\}, \quad \mathbf{Z} \equiv \{\mathbf{H}_k, \omega_k, \sigma_k, P_T, \forall k \in \mathcal{K}\}.$$

对应于算法2.1中的第1-3行，迭代WMMSE算法可以写成更一般的迭代算法形式，

$$\mathbf{U}^t = F_t(\mathbf{V}^{t-1}), \quad (2.18a)$$

$$\mathbf{W}^t = G_t(\mathbf{U}^t, \mathbf{V}^{t-1}), \quad (2.18b)$$

$$\mathbf{V}^t = J_t(\mathbf{U}^t, \mathbf{W}^t), \quad (2.18c)$$

其中 F_t , G_t , 和 J_t 表示的 t 次的迭代函数。图2.1展示了一般形式的迭代WMMSE预编码算法流程图。

2.4 基于模型驱动网络的数字预编码设计

本节基于迭代WMMSE预编码算法，设计了用于求解数字预编码的模型驱动网络结构。

2.4.1 模型驱动网络的结构和前向传播

首先，定义两类非线性函数：（1）矩阵变量相乘；（2）矩阵逐元素的非线性操作：将矩阵 \mathbf{A} 的每个对角线元素取倒数，同时将非对角线元素置0，记作 \mathbf{A}^+ 。因为矩阵求逆操作 \mathbf{A}^{-1} 有很高的计算复杂度，提出使用如下两种低复杂度结构的组合来近似矩阵求逆操作。

- 采用含有逐元素非线性函数 \mathbf{A}^+ 和训练矩阵参数 \mathbf{X} 的结构 $\mathbf{A}^+\mathbf{X}$ ，其中引入矩阵参数 \mathbf{X} 是为了提升网络自由度，从而提升网络性能。注意到当 \mathbf{A} 为对角阵时，有 $\mathbf{A}^{-1} = \mathbf{A}^+$ 。此外，观察到在迭代WMMSE算法中，待求逆矩阵的对角元素往往远大于非对角元素。因此， \mathbf{A}^+ 为 \mathbf{A}^{-1} 较为精确的低复杂度近似。
- 根据矩阵求逆 \mathbf{A}^{-1} 在某点 \mathbf{A}_0 的一阶泰勒展开结构： $\mathbf{A}^{-1} = 2\mathbf{A}_0^{-1} - \mathbf{A}_0^{-1}\mathbf{A}\mathbf{A}_0^{-1}$ ，采用包含矩阵训练参数 \mathbf{Y} 和 \mathbf{Z} 的结构 $\mathbf{A}\mathbf{Y} + \mathbf{Z}$ 来近似 \mathbf{A}^{-1} 。

因此，采用结构 $\mathbf{A}^+\mathbf{X} + \mathbf{A}\mathbf{Y} + \mathbf{Z}$ 来近似矩阵求逆 \mathbf{A}^{-1} 。注意到 $\{\mathbf{X}_k^{u,l+1}, \mathbf{Y}_k^{u,l+1}, \mathbf{Z}_k^{u,l+1}\}$, $\{\mathbf{X}_k^{w,l+1}, \mathbf{Y}_k^{w,l+1}, \mathbf{Z}_k^{w,l+1}\}$, 和 $\{\mathbf{X}_k^{v,l+1}, \mathbf{Y}_k^{v,l+1}, \mathbf{Z}_k^{v,l+1}\}$ 分别表示引入的训练参数集合来近似变量 \mathbf{U}_k^{l+1} , \mathbf{W}_k^{l+1} , 和 \mathbf{V}_k^{l+1} 在第 $(l+1)$ 层的矩阵求逆， $\{\mathbf{O}_k^{u,l+1}, \mathbf{O}_k^{v,l+1}\}$ 为可训练的偏置矩阵参数。由此，模型驱动网络的结构可以表示为

$$\mathbf{U}_k^{l+1} = \left((\mathbf{A}_k^l)^+ \mathbf{X}_k^{u,l+1} + \mathbf{A}_k^l \mathbf{Y}_k^{u,l+1} + \mathbf{Z}_k^{u,l+1} \right) \mathbf{H}_k \mathbf{V}_k^l + \mathbf{O}_k^{u,l+1}, \quad (2.19a)$$

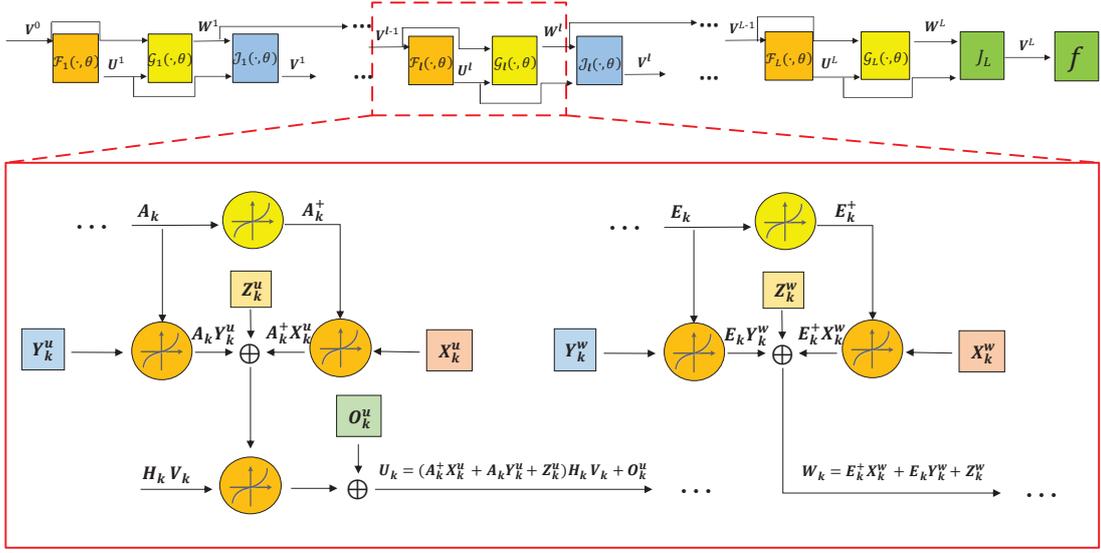


图 2.2 基于迭代WMMSE算法的模型驱动网络结构。

$$\mathbf{W}_k^{l+1} = (\mathbf{E}_k^{l+1})^+ \mathbf{X}_k^{w,l+1} + \mathbf{E}_k^{l+1} \mathbf{Y}_k^{w,l+1} + \mathbf{Z}_k^{w,l+1}, \quad (2.19b)$$

$$\mathbf{V}_k^{l+1} = \left((\mathbf{B}^{l+1})^+ \mathbf{X}_k^{v,l+1} + \mathbf{B}^{l+1} \mathbf{Y}_k^{v,l+1} + \mathbf{Z}_k^{v,l+1} \right) \omega_k \mathbf{H}_k^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} + \mathbf{O}_k^{v,l+1}, \quad (2.19c)$$

其中

$$\mathbf{A}_k^l \triangleq \frac{\sigma_k^2}{P_T} \sum_{k=1}^K \text{Tr}(\mathbf{V}_k^l (\mathbf{V}_k^l)^H) \mathbf{I} + \sum_{m=1}^K \mathbf{H}_k \mathbf{V}_m^l (\mathbf{V}_m^l)^H \mathbf{H}_k^H, \quad (2.20a)$$

$$\mathbf{B}^{l+1} \triangleq \sum_{k=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\omega_k \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} (\mathbf{U}_k^{l+1})^H) \mathbf{I} + \sum_{m=1}^K \omega_m \mathbf{H}_m^H \mathbf{U}_m^{l+1} \mathbf{W}_m^{l+1} (\mathbf{U}_m^{l+1})^H \mathbf{H}_m, \quad (2.20b)$$

$$\mathbf{E}_k^{l+1} \triangleq \mathbf{I} - (\mathbf{U}_k^{l+1})^H \mathbf{H}_k \mathbf{V}_k^l. \quad (2.20c)$$

图2.2展示了详细的模型驱动网络结构。其中红框表示每一层的详细结构，圆圈表示非线性函数，方格表示引入的训练参数。为表述清楚起见，下文略去矩阵变量的 l 上标。因为矩阵变量 \mathbf{U}_k 和 \mathbf{W}_k 的维度通常远小于 \mathbf{V}_k ，将低维度的 \mathbf{U}_k 和 \mathbf{W}_k 设为模型驱动网络的输出更有利于网络的训练和收敛。因此，使用迭代WMMSE算法中 \mathbf{V}_k 的迭代表达式作为网络最后一层，即不对该层的矩阵求逆操作做近似，即图2.2中的 J_L

$$\mathbf{V}_k = \left(\sum_{k=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\omega_k \mathbf{U}_k \mathbf{W}_k \mathbf{U}_k^H) \mathbf{I} + \sum_{m=1}^K \omega_m \mathbf{H}_m^H \mathbf{U}_m \mathbf{W}_m \mathbf{U}_m^H \mathbf{H}_m \right)^{-1} \omega_k \mathbf{H}_k^H \mathbf{U}_k \mathbf{W}_k, \forall k. \quad (2.21)$$

由于信道矩阵 \mathbf{H}_k 为随机变量，可对 \mathbf{H}_k 取期望并将目标函数(2.15)表示成

$$\max_{\{\mathbf{V}_k\}} \sum_{k=1}^K \mathbb{E}_{\mathbf{H}_k} \left\{ \omega_k \log \det \left(\mathbf{I} + \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \left(\sum_{m \neq k} \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H + \frac{\sigma_k^2}{P_T} \sum_k \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \mathbf{I} \right)^{-1} \right) \right\}. \quad (2.22)$$

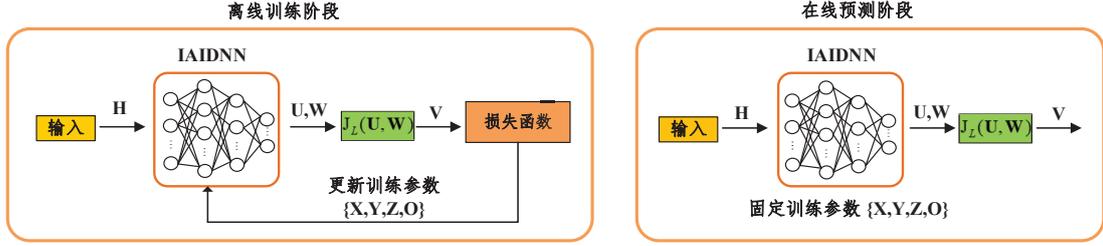


图 2.3 模型驱动网络的离线训练阶段和在线预测阶段示意图

将(2.22)设为模型驱动网络的损失函数，即图2.2中的模块 f 。此外，为了避免网络训练过程中的梯度爆炸，同时使得功率约束(2.12b)满足，将预编码矩阵 \mathbf{V}_k 在每一层的最后用 P_T 进行归一化，即 $\frac{\sqrt{P_T}}{(\sum_k \text{Tr}(\mathbf{V}_k^{l+1}(\mathbf{V}_k^{l+1})^H))^{\frac{1}{2}}} \mathbf{V}_k^{l+1}$ 。

2.4.2 矩阵形式GCR和网络的反向传播训练

首先，将(2.21)代入目标函数(2.15)，计算损失函数关于网络最后一层的变量 \mathbf{U}_k^L 和 \mathbf{W}_k^L 的梯度，记作 $\{\mathbf{G}_k^{u,L}, \mathbf{G}_k^{w,L}\}$ ，详细推导请见附录A.1.2。

在此基础上，根据定理2.1所描述的矩阵形式GCR，可以得到关于第 $(l+1)$ 层中的变量 $\{\mathbf{U}_n, \mathbf{W}_n, \mathbf{V}_n, n \in \mathcal{K}\}$ 对应梯度，到第 l 层中的变量 $\{\mathbf{G}_n^{u,l+1}, \mathbf{G}_n^{w,l+1}, \mathbf{G}_n^{v,l+1}\}$ 对应梯度 $\{\mathbf{G}_n^{u,l}, \mathbf{G}_n^{w,l}, \mathbf{G}_n^{v,l}\}$ 的递推表达式。每层变量梯度 $\{\mathbf{G}_n^{u,l}, \mathbf{G}_n^{w,l}, \mathbf{G}_n^{v,l}, \forall l \in \mathcal{L}, n \in \mathcal{K}\}$ 的详细推导请见附录A.1.3。

根据模型驱动网络结构(2.19)和 $\{\mathbf{G}_n^{u,l}, \mathbf{G}_n^{w,l}, \mathbf{G}_n^{v,l}, \forall l \in \mathcal{L}, n \in \mathcal{K}\}$ ，可得到矩阵参数的梯度

$$\begin{aligned}
\nabla_{\mathbf{X}_k^{u,l+1}} f &= \mathbf{H}_k \mathbf{V}_k^l \mathbf{G}_k^{u,l+1} (\mathbf{A}_k^l)^+, & \nabla_{\mathbf{X}_k^{w,l+1}} f &= \mathbf{G}_k^{w,l+1} (\mathbf{E}_k^{l+1})^+, & \nabla_{\mathbf{Z}_k^{w,l+1}} f &= \mathbf{G}_k^{w,l+1}, \\
\nabla_{\mathbf{Y}_k^{u,l+1}} f &= \mathbf{H}_k \mathbf{V}_k^l \mathbf{G}_k^{u,l+1} \mathbf{A}_k^l, & \nabla_{\mathbf{Y}_k^{w,l+1}} f &= \mathbf{G}_k^{w,l+1} \mathbf{E}_k^{l+1}, & \nabla_{\mathbf{Z}_k^{u,l+1}} f &= \mathbf{H}_k \mathbf{V}_k^l \mathbf{G}_k^{u,l+1}, \\
\nabla_{\mathbf{X}_k^{v,l+1}} f &= (\mathbf{H}_k)^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} \mathbf{G}_k^{v,l+1} (\mathbf{B}^{l+1})^+, & \nabla_{\mathbf{O}_k^{v,l+1}} f &= \mathbf{G}_k^{v,l+1}, & \nabla_{\mathbf{O}_k^{u,l+1}} f &= \mathbf{G}_k^{u,l+1}, \\
\nabla_{\mathbf{Y}_k^{v,l+1}} f &= (\mathbf{H}_k)^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} \mathbf{G}_k^{v,l+1} \mathbf{B}^{l+1}, & \nabla_{\mathbf{Z}_k^{v,l+1}} f &= (\mathbf{H}_k)^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} \mathbf{G}_k^{v,l+1}.
\end{aligned} \tag{2.23}$$

使用SGD来训练模型驱动网络，即 $(\mathbf{X}_k^{u,l})^{m+1} = (\mathbf{X}_k^{u,l})^m + \sigma_m \nabla_{(\mathbf{X}_k^{u,l})^m} f$ ，其中 σ_m 为学习步长， $\nabla_{(\mathbf{X}_k^{u,l})^m} f$ 表示在训练阶段，第 l 层变量 $\mathbf{X}_k^{u,l}$ 在第 m 次迭代的梯度。训练参数的梯度表达式请见公式(2.23)，为表述清楚起见，忽略脚标 m 。基于文献[110]选择步长 σ_m ，步长需要满足如下条件： σ_m 随着迭代 m 单调递减， $\sigma_m \in (0, 1]$ ， $\sigma_m \rightarrow 0$ ， $\sum_m \sigma_m \rightarrow \infty$ 以及 $\sum_m (\sigma_m)^2 < \infty$ ，比如可以选择 $\sigma_m = m^{-\alpha}$ ， $0 < \alpha < 1$ 。训练参数被随机初始化， \mathbf{V}_k^0 通过

迫零预编码初始化。模型驱动网络的训练细节请见算法2.2。此外，图2.3展示了网络的训练和预测阶段，其中 $J_L(\mathbf{U}, \mathbf{W})$ 的定义请见公式(2.21)。

表 2.2 模型驱动网络的训练过程

算法 2.2 模型驱动网络的训练过程

- 1: **输入**: 包含信道矩阵 $\mathbf{H}_k, \forall k$, 的训练数据集 \mathcal{H} ;
 - 2: **输出**: 预编码矩阵 $\mathbf{V}_k, \forall k$;
 - 3: **初始化**: $\{\mathbf{V}_k\}$ 使得 $\sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \leq P_T$ 满足; 设定网络层数 L , 每批数据量的大小 N , 网络训练收敛精度 ϵ , 网络训练最大迭代次数 I_{\max} , 初始化当前训练迭代次数 $m = 0$;
 - 4: **while** 损失函数在验证集上未收敛且 $m \leq I_{\max}$ **do**
 - 5: **前向传播**: 从训练数据集中选择一组训练样本 $\{\mathbf{H}_k, \forall k\}$, 并初始化预编码矩阵 $\{\mathbf{V}_k^0, \forall k\}$ 。根据(2.19)-(2.20) 计算 $\{\mathbf{U}_k^l, \mathbf{W}_k^l, l = 1, 2, \dots, L, \forall k\}$ 和 $\{\mathbf{V}_k^l, l = 1, 2, \dots, L - 1, \forall k\}$;
 - 6: 根据(2.21)计算网络最后一层输出的预编码矩阵 $\{\mathbf{V}_k^L, \forall k\}$ 。随后, 将 $\{\mathbf{V}_k^L, \forall k\}$ 代入损失函数并计算损失函数的值;
 - 7: **反向传播**: 根据附录A.1.2计算损失函数关于网络最后一层变量 \mathbf{U}_k^L 和 \mathbf{W}_k^L 的梯度。随后, 根据附录A.1.3中的(A.10)-(A.13)计算网络其他层变量 $\{\mathbf{U}_k^l, \mathbf{W}_k^l, \mathbf{V}_k^l, l = L - 1, \dots, 2, 1, \forall k\}$ 的梯度。最后, 根据(2.23)计算训练参数 $\{\mathbf{X}_k^{u,l}, \mathbf{Y}_k^{u,l}, \mathbf{Z}_k^{u,l}, \mathbf{O}_k^{u,l}\}, \{\mathbf{X}_k^{w,l}, \mathbf{Y}_k^{w,l}, \mathbf{Z}_k^{w,l}\}$ 和 $\{\mathbf{X}_k^{v,l}, \mathbf{Y}_k^{v,l}, \mathbf{Z}_k^{v,l}, \mathbf{O}_k^{v,l}\}$ 的梯度;
 - 8: **更新训练参数**: 重复执行 N 次步骤5-7, 并根据一批样本计算网络训练参数的梯度平均值。随后, 使用SGD更新训练参数;
 - 9: $m = m + 1$;
 - 10: **end while**
-

2.5 计算复杂度和泛化性分析

本节展示了基于CNN的黑盒网络结构, 并将其作为比较的基线算法。此外, 提供参数维度、计算复杂度和网络泛化性能的分析。

2.5.1 基于CNN的黑盒网络

根据文献[90], 引入基于CNN架构的黑盒网络, 用于作为基线算法和本章提出的模型驱动网络进行对比。

图2.4展示了基于CNN的黑盒网络结构。其输入为信道 $\mathbf{H} \triangleq [\mathbf{H}_1^T, \mathbf{H}_2^T, \dots, \mathbf{H}_K^T]^T$, 依次经过卷积层, 批归一化层, 和非线性函数。该过程重复若干次, 即输入信道矩阵需要通过若干上述网络结构。随后, 经过全连接层, 并采用拉平和随机丢弃一定数量神经元的技

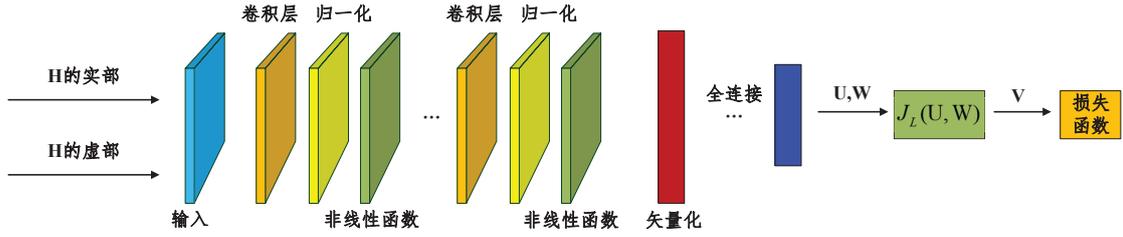


图 2.4 黑盒网络结构示意图

术。此外，采用ReLU函数作为非线性激活函数。CNN的输出是变量 \mathbf{U}_k 和 \mathbf{W}_k ，而不是 \mathbf{V}_k ，因为网络更容易学习低维度的变量 \mathbf{U}_k 和 \mathbf{W}_k ，收敛速度和效果好于让网络学习高维度变量 \mathbf{V}_k 。随后，将变量 \mathbf{U}_k 和 \mathbf{W}_k 代入迭代表达式(2.21)，从而计算得到预编码矩阵 \mathbf{V}_k 。最后，将 \mathbf{V}_k 代入损失函数，至此前向传播过程结束。基于CNN的传统黑盒网络的反向传播过程用Tensorflow等深度学习平台自动实现。采用非监督训练方式来提升传统黑盒网络的性能，训练过程可以分为如下两个阶段

- 监督训练阶段：首先，采用 $\sum_{k=1}^K (\|\mathbf{U}_k - \hat{\mathbf{U}}_k\|^2 + \|\mathbf{W}_k - \hat{\mathbf{W}}_k\|^2)$ 作为损失函数，其中 $\hat{\mathbf{U}}_k$ 和 $\hat{\mathbf{W}}_k$ 是通过迭代WMMSE算法产生的训练标签。
- 无监督训练阶段：完成监督训练之后，将(2.22)设为损失函数，进行无监督训练。

2.5.2 参数维度和计算复杂度分析

本小节讨论模型驱动网络，基于CNN的黑盒网络和迭代WMMSE算法的参数维度和计算复杂度。

2.5.2.1 参数维度

模型驱动网络的参数维度即引入的矩阵参数 $\{\mathbf{X}_k^{u,l}, \mathbf{Y}_k^{u,l}, \mathbf{Z}_k^{u,l}, \mathbf{O}_k^{u,l}\}$ ， $\{\mathbf{X}_k^{w,l}, \mathbf{Y}_k^{w,l}, \mathbf{Z}_k^{w,l}\}$ 和 $\{\mathbf{X}_k^{v,l}, \mathbf{Y}_k^{v,l}, \mathbf{Z}_k^{v,l}, \mathbf{O}_k^{v,l}\}$ 的维度。因此，模型驱动网络每层的参数维度是

$$(3N_r^2 + 3d^2 + 3N_t^2 + dN_r + dN_t)K. \quad (2.24)$$

因为最后一层没有参数 $\{\mathbf{X}_k^{v,l}, \mathbf{Y}_k^{v,l}, \mathbf{Z}_k^{v,l}, \mathbf{O}_k^{v,l}\}$ ，整个模型驱动网络的参数维度是

$$LK(3N_r^2 + 3d^2 + dN_r) + (L-1)K(3N_t^2 + dN_t), \quad (2.25)$$

其中 L 表示网络层数。基于CNN的传统黑盒网络的参数维度

$$\sum_{l=1}^{L-2} S_l^2 C_{l-1} C_l + KN_r N_t C_{L-2} C_{out}, \quad (2.26)$$

其中 S_l 和 C_l 分别表示卷积层第 l 层卷积核的大小和通道数目， C_{out} 表示全连接层的输出维度。第一项和第二项分别表示卷积层和全连接层的参数维度。本章设 $S_l = 5$, $C_l = 32, \forall l$ 和 $C_{out} = 1024$ 。

2.5.2.2 计算复杂度

经典迭代WMMSE算法的计算复杂度是：

$$\mathcal{O}(L_w(K^2 N_t N_r^2 + K^2 N_t^2 N_r + K N_t^3 + K N_r^3)), \quad (2.27)$$

其中 L_w 表示算法的迭代次数。模型驱动网络在推理阶段的计算复杂度为：

$$\mathcal{O}(L_a(K^2 N_t N_r^2 + K^2 N_t^2 N_r + K N_t^{2.37} + K N_r^{2.37})), \quad (2.28)$$

其中 L_a 表示层数。从两方面可以看出模型驱动网络的计算复杂度比迭代WMMSE算法低：

- 模型驱动网络的层数更少, i.e., $L_a < L_w$ 。
- 迭代WMMSE算法需要矩阵求逆操作，计算复杂度为 $\mathcal{O}(n^3)$ 。相比之下，模型驱动网络只包含矩阵相乘操作，计算复杂度为 $\mathcal{O}(n^{2.37})$ 。

此外，推理阶段传统黑盒网络的计算复杂度为

$$\mathcal{O}\left(\sum_{l=1}^{L-2} M_l^2 S_l^2 C_{l-1} C_l + K N_r N_t C_{L-2} C_{out} + (K^2 N_t N_r^2 + K^2 N_t^2 N_r + K N_t^3 + K N_r^3)\right), \quad (2.29)$$

其中 $M_l = (\frac{P_l - S_l + 2 * P_a}{S_l} + 1)$ 表示第 l 层的输出维度。注意到 P_l , P_a , 和 S_l 分别表示第 l 层的输入维度，填零数目和滑动步长。

在训练阶段，由于模型驱动网络采用闭式梯度(2.23)来更新参数，相比传统黑盒网络，训练更高效，收敛时间更短，尤其是在非监督训练阶段。

2.5.3 泛化性分析

2.5.3.1 泛化性

训练好一个系统参数是 (N_{t_0}, N_{r_0}, K_0) 的模型驱动网络，它可以直接迁移到具有相同系统参数 N_{t_1} 和 N_{r_1} ，但 K_1 更小的场景，即 $(N_{t_1} = N_{t_0}, N_{r_1} = N_{r_0}, K_1 < K_0)$ ，而不需要重新训练一个网络。在推理阶段，只需要将 $\{\mathbf{H}_k, k \leq K_1\}$ 和 $\{\mathbf{H}_k = \mathbf{0}, K_1 < k \leq K_0\}$ 作为输入。此外，当 $N_{t_1} < N_{t_0}$ 和 $N_{r_1} < N_{r_0}$ ，只需要将输入信道矩阵 \mathbf{H}_k 中相应的列向量和行向量设成 $\mathbf{0}$ 。

表 2.3 系统天线数为 $N_t = 8$ 和 $N_t = 16$ 的和速率性能

天线数(N_t)	8				16			
用户数(K)	1	2	3	4	2	4	6	8
WMMSE (bits/s/Hz)	13.13	22.12	27.74	31.82	25.93	43.34	53.66	58.83
IAIDNN	99.34%	99.15%	97.36%	91.35%	99.59%	99.11%	97.67%	92.13%
Black-box	93.76%	92.09%	90.27%	81.85%	93.68%	92.39%	89.32%	80.56%

表 2.4 系统天线数为 $N_t = 64$ 的和速率性能

用户数(K)	5	10	15	20	25	30
WMMSE (bits/s/Hz)	71.04	123.15	164.71	194.72	208.96	216.92
IAIDNN	99.91%	99.82%	99.76%	99.11%	98.88%	97.58%
Black-box	93.56%	92.83%	92.02%	90.38%	87.98%	82.14%

比如，对于场景($N_{t_0} = 32, N_{t_1} = 16, N_{r_0} = N_{r_1} = 2, K_0 = 10, K_1 = 5$)，可以将训练好的具有系统参数(N_{t_0}, N_{r_0}, K_0)的模型直接迁移到系统(N_{t_1}, N_{r_1}, K_1)。对于每一个样本，只需要将 $\{\mathbf{H}_k, k \leq 5\}$ 和 $\{\mathbf{H}_k = \mathbf{0}, 5 < k \leq 10\}$ 作为输入，同时对 $\{\mathbf{H}_k, k \leq 5\}$ 增加16个全零列向量。

2.5.3.2 扩展

前文假定传输功率 P_T 和噪声 σ_k 是固定的，即模型驱动网络的固定系统参数。很容易将其扩展成，和信道矩阵 $\{\mathbf{H}_k\}$ 一样，作为模型驱动网络的输入。比如，设 $P_T \sim \mathcal{N}(a_1, b_1)$ ， $\sigma_k \sim \mathcal{N}(a_2, b_2)$ ，其中 $\mathcal{N}(a, b)$ 表示均值为 a ，方差为 b 的高斯分布。

提出的模型驱动网络同样适用于存在信道估计误差求解鲁棒预编码矩阵的场景和多小区场景^[86]。可以直接将模型驱动网络拓展到这些场景，通过考虑信道估计误差的统计特性及略微调整损失函数。

2.6 仿真结果

本节通过仿真结果验证IAIDNN在性能、计算复杂度等方面的有效性。

2.6.1 仿真设置

在仿真中采用MIMO衰落信道模型，其中信道矩阵 \mathbf{H}_k 中的每个元素均服从于复高斯分布 $\mathcal{CN}(0, 1)$ 。设SNR为20 dB，并假设所有的用户均配备了 $N_r = 2$ 根接收天线。为较好地

表 2.5 系统天线数为 $N_t = 128$ 的和速率性能

用户数(K)	10	20	30	40	50	60
WMMSE (bits/s/Hz)	139.15	244.03	326.76	386.21	412.82	417.64
IAIDNN	99.87%	99.67%	99.32%	99.03%	98.68%	97.87%
Black-box	93.58%	92.76%	91.58%	89.93%	87.31%	83.89%

表 2.6 系统天线数为 $N_t = 256$ 的和速率性能

用户数(K)	20	30	40	50	60	70	80
WMMSE (bits/s/Hz)	279.56	389.31	487.56	575.32	652.59	718.41	772.07
IAIDNN	99.86%	99.73%	99.51%	99.29%	99.05%	98.79%	98.33%
Black-box	93.23%	92.97%	92.28%	91.45%	89.96%	88.03%	85.86%

训练IAIDNN，将训练批次大小设为 $N = 100$ ，IAIDNN的层数设为 $L = 7$ 。测试集的大小设为5000个信道样本，用来评估网络的性能。评估基线算法的性能时，随机化不同的初始值，执行20次迭代WMMSE算法，并将最优的结果作为其迭代算法的性能，该性能可被认为是全局最优的近似。另外，表格中IAIDNN性能的“%”表示将IAIDNN的和速率除以迭代WMMSE算法的和速率得到的百分比。与IAIDNN类似，Black-box网络中的百分比是将其性能和迭代WMMSE算法的性能相除得到的百分比。

2.6.2 和速率性能

本小节评估上述各种算法在不同用户数 K 、天线数 N_t 场景下的性能。根据表2.3、表2.4、表2.5和表2.6中的结果可知，提出的IAIDNN能够逼近迭代WMMSE算法的性能，且IAIDNN和迭代WMMSE算法的性能差距会随着用户数 K 的增加而增加。可以观察到，IAIDNN性能远超Black-box网络，且性能差距也会随着用户数 K 的增加而增加。此外，当 $K \times N_r$ 接近 N_t 时，IAIDNN的性能会更显著地超越Black-box网络，且Black-box网络的性能将会严重降低。这主要是因为，在这种场景下，最大化和速率问题会更加复杂，会有更多的局部最优解和鞍点，而Black-box网络很难处理这种复杂的情况。另外，预编码矩阵 \mathbf{V}_k 中元素的差别也会随着用户数 K 的增加而增加，即接近0或1，这也增加了网络训练的难度。

图2.5展示了训练批次大小和学习率对模型收敛性的影响。从图2.5(a)可知，尽管采用更大的训练批次会使收敛变慢，但是能够增加训练的稳定性。从图2.5(b)可知，采用较小的学习率可以使得模型达到更好的性能，而较大的学习率则可以使得模型更快收敛。另外，所提出的学习率方案实现了性能和收敛速度之间的平衡。

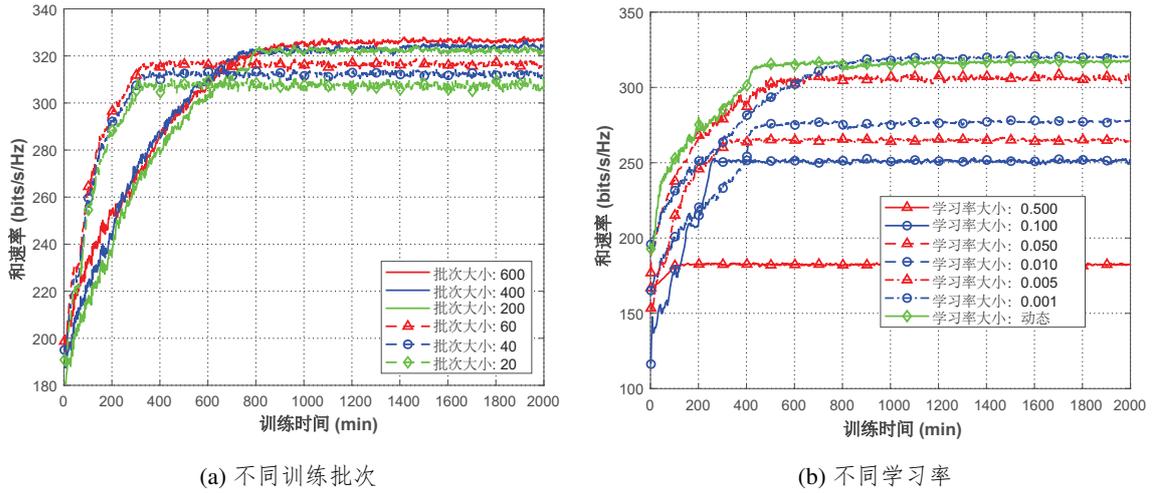


图 2.5 不同训练批次大小和学习率下系统($N_t = 128, K = 30$)的收敛性能

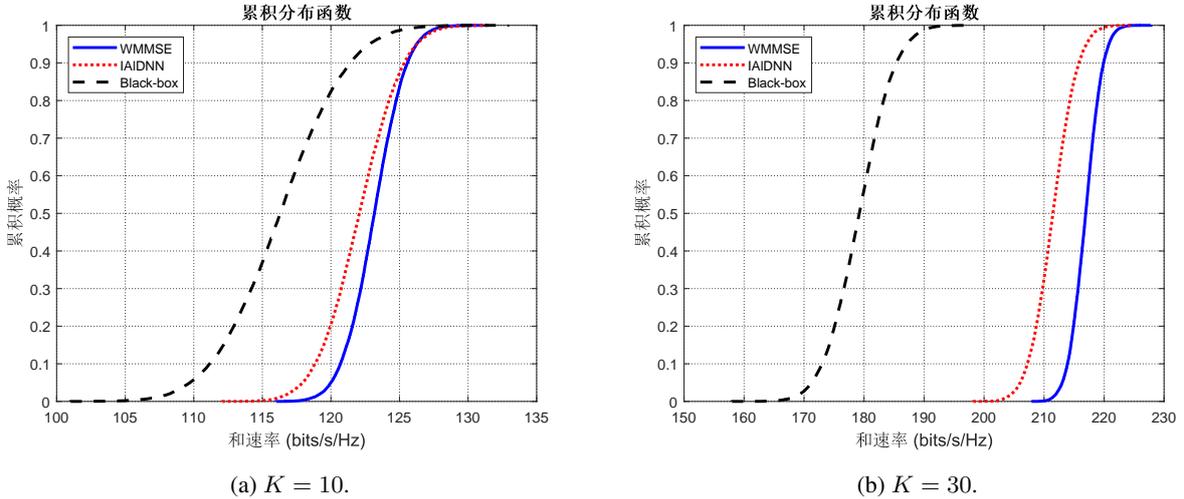


图 2.6 系统 $N_t = 64$ 的测试数据集中不同样本的和速率性能累积分布函数

图2.6展示了累积分布函数 (Cumulative Distribution Function, CDF)，其描述了不同方法在系统 $N_t = 64$ 的测试数据集大小为50000个样本 $\{\mathbf{H}, \forall k\}$ 的和速率性能累积分布。可以看到，当用户数比较小时，例如 $K = 10$ 时，IAIDNN可以达到迭代WMMSE算法99.82%的和速率性能，而此时Black-box网络仅仅能达到92.83%的性能。另外，这些方案之间的差距会随着用户数 K 增加而增加。迭代WMMSE算法实现的不同样本之间的性能方差最小，而Black-box网络的方差最大。

图2.7描述了系统 $N_t = 128$ 的测试数据集中不同样本的和速率性能分布。可以看到，IAIDNN实现的样本和速率性能分布非常接近迭代WMMSE算法，为其提供了一个很好的近似，且IAIDNN实现的和速率性能分布显著优于Black-box网络。随着用户 K 数增加时，

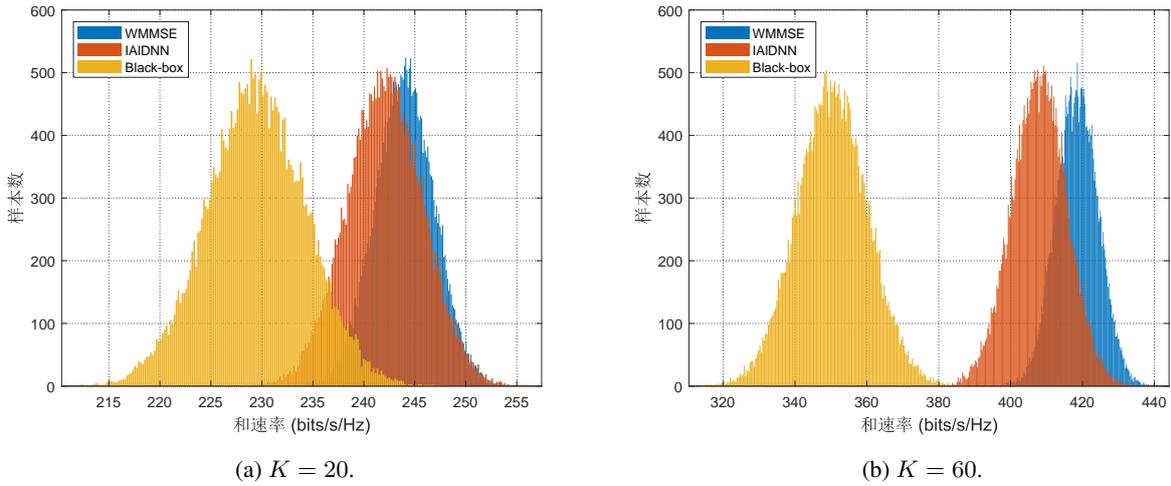


图 2.7 系统 $N_t = 128$ 的测试数据集中不同样本的和速率性能分布

表 2.7 系统($N_t = 64, K = 30$)的和速率随SNR的变化

SNR (dB)	0	5	10	15	20	25	30
WMMSE (bits/s/Hz)	45.86	85.52	118.32	164.77	216.92	266.55	317.18
IAIDNN	97.06%	97.11%	97.25%	97.32%	97.58%	97.79%	98.02%
Black-box	80.32%	80.54%	80.87%	81.32%	81.93%	82.65%	83.72%

尤其当 $K \times N_r$ 的值接近 N_t 时，IAIDNN对迭代WMMSE算法的性能分布近似变差。

表2.7展示了系统($N_t = 64, K = 30$)的和速率性能随SNR的变化情况。IAIDNN和Black-box网络能够达到的和速率性能随着SNR的增加逐渐提升。这主要是因为当SNR增加时，被求解问题的可行域扩大，网络更容易找到更优的解。

表2.8描述了系统($N_t = 128, K = 40$)的和速率性能随训练样本数的变化。相比Black-box网络，IAIDNN所需要的训练样本数显著降低，这主要是因为IAIDNN充分利用了迭代WMMSE算法的结构。由于在实际系统中信道样本是通过信道估计获得的，代价很高且很难获取，该优势在实际工业应用中具有重大意义。

根据表2.9可知，随着层数 L 的增加，IAIDNN的和速率性能先增加，然后降低。这是因为当 L 较小时，IAIDNN的自由度较小，导致其训练参数的数量较少、学习能力不理想。因此，当 L 增加时，其自由度提升、训练参数的数量增加、学习能力提高，性能就会相应提高。然而，网络训练反向传播过程中，梯度的数值误差随层数 L 的增加而增加，因为IAIDNN涉及一系列矩阵乘法运算等操作会导致误差的累计。当 L 相对较大时，例如 $L = 8$ 时，网络受到数值误差的影响导致学习能力下降，从而和速率性能会降低。此外，IAIDNN的训练时间也会随着 L 的增加而增加。因此， $L = 7$ 是层数的最佳选择，因为它在

表 2.8 系统($N_t = 128, K = 40$)的和速率随训练样本数的变化

训练样本数	5000	10000	15000	20000	25000	30000	35000	40000
Black-box	78.93%	83.14%	86.59%	88.36%	89.66%	89.93%	90.15%	90.15%
训练样本数	100	200	300	400	500	600	700	800
IAIDNN	93.03%	95.89%	97.31%	98.52%	98.94%	99.03%	99.26%	99.26%

表 2.9 系统($N_t = 64, K = 30$)的和速率随模型驱动网络层数的变化

网络层数	3	4	5	6	7	8	9	10
和速率	90.58%	93.35%	95.56%	96.61%	97.58%	97.61%	97.34%	96.93%
网络训练时间 (min)	265.63	279.35	289.02	296.97	301.61	306.16	310.91	313.56

性能和网络训练时间之间达到了较好的平衡。

图2.8描述了不同层数IAIDNN的和速率性能。在不同场景下，IAIDNN的性能都先随着层数的增加而提升，随后性能会略微下降。较少层数和较大的层数的IAIDNN的性能差距，通常是会随着 N_t 和 K 的增加而增加。具体来说，3层IAIDNN的性能在($N_t = 16, K = 4$)时可以逼近7层IAIDNN的性能，但在($N_t = 64, K = 15$)场景下，二者的差距较大。此外，由于更高维度的矩阵求逆操作需要更多的层数来近似，IAIDNN的最优层数随 N_t 和 K 的增加而增加。

图2.9展示了不同算法在非完美信道下的和速率性能。由图可知，模型的性能随着信道误差方差的增加而下降。其中，IAIDNN具有最优的性能，其次是迭代WMMSE算法，Black-box网络性能最差。这一现象表明了提出的IAIDNN能够较好地对抗信道误差。因为IAIDNN是通过最大化大批量训练样本和速率的均值得到的网络模型，所以它会比迭代WMMSE算法具有更好的鲁棒性。

2.6.3 复杂度分析

表2.10比较了各种方法的计算复杂度，采用训练和测试阶段所用的CPU运行时间作为指标。可以观察到，所有方案训练和测试阶段的CPU运行时间都会随着用户数 K 和发射天线数 N_t 的增加而增加。并且训练阶段的CPU运行时间相比测试阶段增长更快，这主要是因为训练阶段包含更多的高维矩阵乘法和求逆操作。此外，IAIDNN比Black-box网络训练时间更短，即具有更快的收敛速度。其原因在于非监督学习的损失函数(2.22)复杂，且Black-box网络计算梯度是直接采用Tensorflow平台的自动求导机制。相比之下，本章提出的基于GCR的反向传播训练方式，推导了IAIDNN的闭式梯度表达式，使得其训练更加高效。在训练阶段，IAIDNN和Black-box网络的CPU运行时间差距会随着 K 和 N_t 的

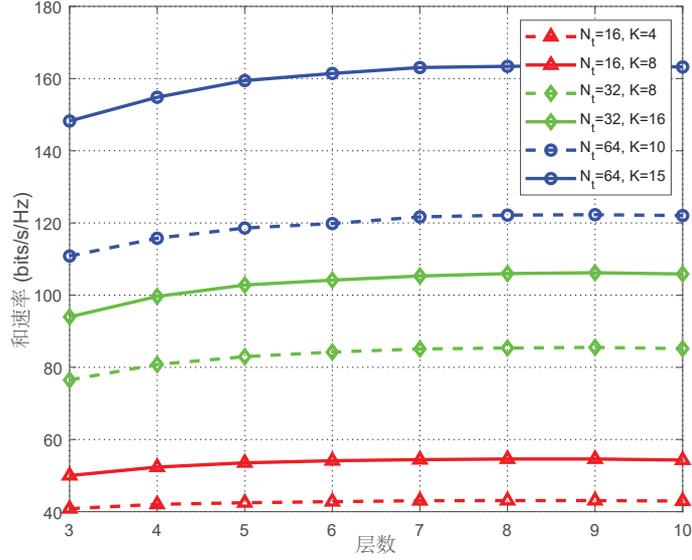


图 2.8 不同层数IAIDNN的和速率性能

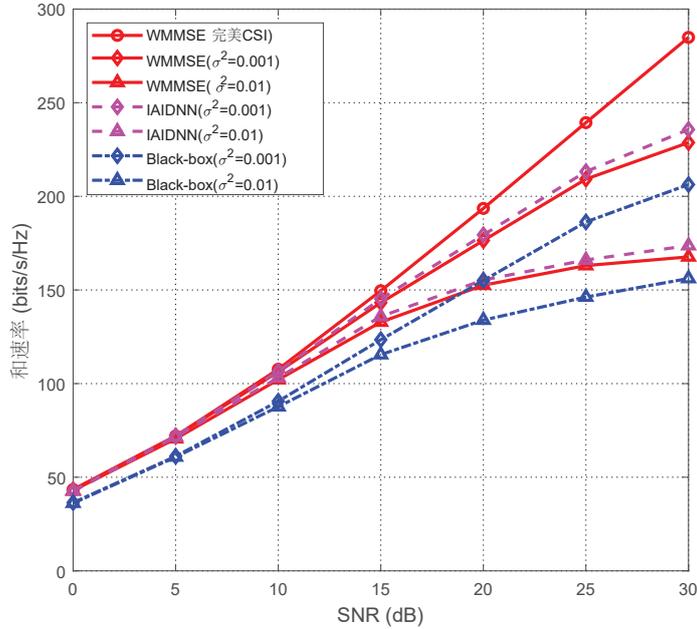


图 2.9 系统($N_t = 64, K = 20$)非完美信道下的和速率性能

增加而扩大。在测试阶段，IAIDNN比迭代WMMSE算法需要的CPU运行时间更少，这也是IAIDNN的优势所在。在大规模MIMO场景下，例如 $N_t = 128$ ，这一优势会更加显著，也使得IAIDNN可以被广泛用于实际通信系统中。

2.6.4 泛化性分析

表2.11描述了IAIDNN的泛化能力。在($N_t = 256, K = 80$)的场景下训练IAIDNN，并将该网络用于具有更少天线数 N_t 和用户数 K 的场景。通过对比表格2.4、表格2.5和表格2.6中

表 2.10 CPU运行时间

天线和用户数 (N_t, K)	训练阶段CPU运行时间(min)		预测阶段CPU运行时间(s)		
	IAIDNN	Black-box	IAIDNN	Black-box	WMMSE
(8,4)	8.71	11.12	0.01	0.01	0.08
(16,8)	21.52	31.55	0.01	0.01	0.22
(32,8)	28.65	60.18	0.02	0.03	0.51
(32,16)	70.35	130.54	0.03	0.04	1.05
(64,15)	102.23	153.66	0.04	0.05	1.46
(64,30)	301.61	467.12	0.11	0.13	4.22
(128,30)	514.56	1439.43	0.13	0.16	5.14
(128,60)	1242.23	4184.02	0.32	0.39	29.68
(256,30)	1056.59	3758.51	0.61	0.71	32.31
(256,60)	3126.12	9986.37	0.83	0.98	38.56
(256,120)	9806.85	—	2.94	3.36	291.01

表 2.11 IAIDNN的泛化能力

系统参数	$K = 70$	$K = 60$	$K = 50$	$K = 40$	$K = 30$	$K = 20$	$K = 10$
$N_t = 256$	98.07%	98.25%	98.36%	98.51%	98.84%	99.01%	99.12%
$N_t = 128$	—	95.12%	95.97%	96.67%	96.93%	97.21%	97.43%
$N_t = 64$	—	—	—	—	94.69%	96.82%	97.75%

的结果，可以看到将该大规模网络直接应用于具有相同天线数 N_t 和较小用户数 K 的场景，性能损失仅在1%左右。而如果应用于具有较小天线数 N_t 和较小用户数 K 的场景，这一损失大概在3%。由此可见IAIDNN具有较强的泛化能力。

2.6.5 IAIDNN在用户满载场景中的改进

由前述仿真结果可知，当 $K \times N_r < N_t$ 时，IAIDNN能实现较好的和速率性能，能够逼近迭代WMMSE算法。然而，在用户满载系统中，即当 $K \times N_r = N_t$ 时，IAIDNN实现的和速率性能会下降。为了解决这一问题，对IAIDNN的网络结构进行了一定的修改，来提升用户满载系统的性能。具体而言，引入了求逆操作 \mathbf{A}^{-1} ，以及新的训练参数 \mathbf{P} 来增加自由度。随后，新的结构， $\mathbf{A}^+ + \mathbf{P}\mathbf{A}^{-1}\mathbf{Y} + \mathbf{Z}$ 被用来替换原本公式(2.19)中的 $\mathbf{A}^+\mathbf{X} + \mathbf{A}\mathbf{Y} + \mathbf{Z}$ ，其中 \mathbf{P} 、 \mathbf{X} 、 \mathbf{Y} 和 \mathbf{Z} 为引入的训练参数，且训练方法和前文一致。表格2.12的结果证明了在用户满载系统中，该改进的IAIDNN能够比未改进的IAIDNN具有更好的和速率性能，并且能超越7次迭代WMMSE算法的性能。然而，改进的IAIDNN具有更多的训练参数，并且需要少量的矩阵求逆操作，这导致了计算复杂度的提高和训练时间的增加。

表 2.12 用户满载系统中改进IAIDNN的和速率性能

系统参数(N_t, K)	(8, 4)	(16, 8)	(32, 16)	(64, 32)	(128, 64)
WMMSE 7次迭代	79.19%	80.63%	82.72%	83.05%	83.52%
IAIDNN	91.35%	92.13%	92.63%	92.82%	92.95%
改进IAIDNN	95.86%	96.02%	96.85%	96.93%	97.08%
训练时间(min)	10.65	27.52	86.37	372.58	2112.31

2.7 本章小结

本章提出了一种新颖的适用于物理层通信的矩阵形式模型驱动网络框架。为了设计MU-MIMO系统中的预编码矩阵，在经典的迭代WMMSE算法基础上，根据提出的框架设计了IAIDNN。具体而言，迭代WMMSE算法被展开为类似于DNN的逐层结构，其中引入了一系列训练参数来取代高复杂性的矩阵操作。为了训练该网络，提出了矩阵形式的反向传播梯度GCR，来描述相邻两层之间梯度的递推关系。仿真结果表明，IAIDNN可以有效地逼近迭代WMMSE算法的性能，并降低了计算复杂度。因此，IAIDNN可作为迭代优化算法在实际系统中的替代。未来的工作可以将本章提出的模型驱动网络框架推广应用到其他具有挑战性的通信应用场景，例如存在信道误差的鲁棒预编码设计，多小区系统中的预编码设计，以及包含离散变量的波束选择问题等。

3 基于深度强化学习的可变层模型驱动网络：稀疏信道估计

3.1 引言

上一章提出了针对无线通信系统的模型驱动网络框架，并将其用于物理层数字预编码矩阵的设计。尽管基于迭代优化算法设计的模型驱动网络以较低的计算复杂度实现了较好的系统性能，但在设计过程中却忽略了模型驱动网络与迭代优化算法之间的主要区别^[255]。迭代优化算法可以针对不同的输入样本调整迭代次数，如图3.1(a)所示。而模型驱动网络的层数是预先确定的，计算复杂度与网络层数成正比，如图3.1(b)所示。固定层数通常会导致两个缺陷^[256]：(1) 处理“简单”的输入样本时，使用了过多的层数导致计算资源的浪费；(2) 处理“困难”的输入样本时，网络层数不够导致性能较差。此外，追求更好的性能并不是物理层收发机设计唯一的目标，一些URLLC应用要求在较短的时间内解决问题，对算法复杂度和时延有严格的要求。可变层模型驱动网络为解决该问题提供了可能性^[255-258]。然而，已有的可变层模型驱动网络架构还存在一些问题：(1) 可变层模型驱动网络往往具有不连续的损失函数，加大了训练难度^[257]；(2) 每一层的停止概率仅基于该层的输出来计算得到，这导致针对不同样本的层数选择具有一定的随机性，难以针对每个样本选择合适的层数^[258]；(3) 网络的输出是各层输出的加权和，这与迭代优化算法输出最后一层的结果不同^[257]；(4) 需要预先确定网络的最大层数，并以最大层数训练网络，这将导致训练过程出现较为严重的梯度爆炸或梯度消失问题。

本章提出了一种基于DRL的可变层模型驱动网络框架。模型驱动网络的训练参数由深度确定性策略梯度（Deep Deterministic Policy Gradient, DDPG）学习，而不是直接通过SGD更新，这避免了训练过程中的梯度消失和梯度爆炸现象。特别地，该现象随着模型驱动网络层数的增加，会严重影响梯度的反向传播过程，导致网络性能下降。基于DRL设计的可变层模型驱动网络可以很好地解决这一问题，因为在该框架下，模型驱动网络层数的增加仅仅导致DDPG的时间步长增加，并不会影响网络梯度的反向传播过程。具体而言，模型驱动网络的优化变量、训练参数和网络结构分别设计为DDPG的状态、动作和状态转移。为了实现模型驱动网络的可变层，设计了奖励函数和停止分数。采用DDPG的动

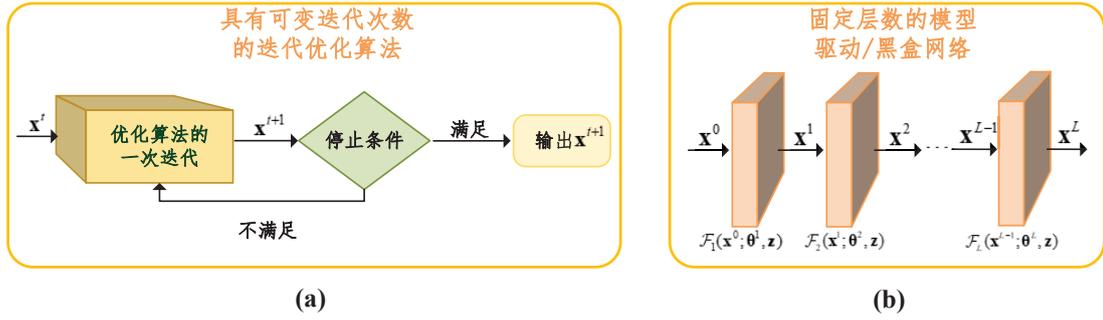


图 3.1 优化算法和网络对比：(a) 迭代次数随样本可变的优化算法；(b) 层数固定的模型驱动/黑盒网络

机主要在于：(1) 能够实现可变层模型驱动网络；(2) 可避免训练过程中的梯度消失和梯度爆炸，使得网络训练过程更稳定，收敛性能更好；(3) 可处理连续动作，而DDPG的动作（即模型驱动网络的训练参数）通常是连续的。

随后，将该框架用于解决大规模MU-MIMO系统中的信道估计问题。采用一种通用的离网模型，对配置均匀线性阵列的大规模MU-MIMO系统进行信道的稀疏表示。建模了基于离网基底的信道估计问题，并基于最大-最小优化（Majorization-Minimization, MM）的思想进行联合稀疏信道恢复和离网参数的优化求解。具体而言，提出了一种高效的基于SBL的迭代优化算法，并将该迭代算法展开为逐层结构，以更少的层数达到接近SBL算法的性能。为提高信道估计的精度，引入训练参数替代SBL算法中难以事先确定的先验分布参数以及高计算复杂度操作。

注意到不同的信道样本有不同的稀疏度，因此针对不同信道样本，SBL算法的迭代次数一般不同。本章根据基于SBL算法的模型驱动网络结构，采用所提出的基于DRL的可变层模型驱动网络框架来更好地求解该信道估计问题。具体而言，将待优化变量定义为DDPG的状态，引入的训练参数设计为DDPG的动作，并将基于SBL算法的模型驱动网络结构定义为DDPG的状态转移过程。为实现可变层，设计了用DDPG中的一个子网络来输出停止分数，用于指示网络何时停止运行并输出最终结果，且该停止分数为信道估计误差的函数，用于更精确地刻画信道估计误差。DDPG的奖励函数设计为以下两部分的加权和：(1) 归一化均方误差（Normalized Mean Squared Error, NMSE）的减小，即两次迭代之间的性能提升，并在奖励中引入惩罚，用于在DDPG不选择停止时给予该策略一定的惩罚；(2) 设计一个与停止分数相关的函数，可通过调整其中的超参数，来控制信道估计误差和模型驱动网络停止时运行的层数。

本章内容安排如下。3.2节提出了基于DRL的可变层模型驱动网络的一般框架。3.3节建模了稀疏信道估计问题，并提出了一种基于SBL的高效迭代优化算法。3.4节将基于SBL的算法展开为具有逐层结构的模型驱动网络，并提供了性能分析。3.5节提出了基于SBL算法的DDPG驱动的可变层模型驱动网络，并将该可变层架构扩展到一般的深度学习网络。仿真结果请见章节3.6。最后，3.7节对本章进行了总结。

3.2 基于深度强化学习的可变层模型驱动网络框架

本节提出基于DRL的可变层模型驱动网络框架。

3.2.1 问题建模

3.2.1.1 优化问题和迭代算法

优化问题通常具有如下形式

$$\min_{\mathbf{x}} f(\mathbf{x}; \mathbf{z}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{X}, \quad (3.1)$$

其中 $f: \mathbb{C}^m \mapsto \mathbb{R}$ 表示目标函数， $\mathbf{x} \in \mathbb{C}^m$ 是优化变量， \mathcal{X} 表示优化问题的可行域， $\mathbf{z} \in \mathbb{C}^p$ 是预先给定的问题参数。如图3.1(a)所示，用于求解优化问题(3.1)的迭代算法具有如下形式

$$\mathbf{x}^t = F_t(\mathbf{x}^{t-1}; \mathbf{z}), \quad (3.2)$$

其中 $t \in \mathcal{T} \triangleq \{1, 2, \dots, T_a\}$ 表示迭代次数， T_a 表示最大迭代次数，函数 F_t 根据参数 \mathbf{z} ，在第 t 次迭代将变量 \mathbf{x}^{t-1} 映射为 \mathbf{x}^t 。

3.2.1.2 模型驱动网络

模型驱动网络将迭代算法展开为类似于DNN的逐层结构。根据迭代表达式(3.2)，通过引入训练参数 $\boldsymbol{\theta} \in \mathbb{C}^{a \times b}$ ，图3.1(b)中描述的模型驱动网络可以表示为

$$\mathbf{x}^l = \mathcal{F}_l(\mathbf{x}^{l-1}; \boldsymbol{\theta}^l, \mathbf{z}), \quad (3.3)$$

其中 $l \in \mathcal{L} \triangleq \{1, 2, \dots, L\}$ 表示模型驱动网络的层数， L 表示模型驱动网络的最大层数， \mathcal{F}_l 是第 l 层的模型驱动网络结构， \mathbf{x}^{l-1} 和 \mathbf{x}^l 分别表示第 l 层网络的输入和输出， \mathbf{z} 是预先给定的网络参数，比如网络的输入， $\boldsymbol{\theta}^l$ 表示第 l 层的训练参数。此外，(3.1)中的目标函数 $f(\mathbf{x}; \mathbf{z})$ 可以被设计为模型驱动网络的损失函数。

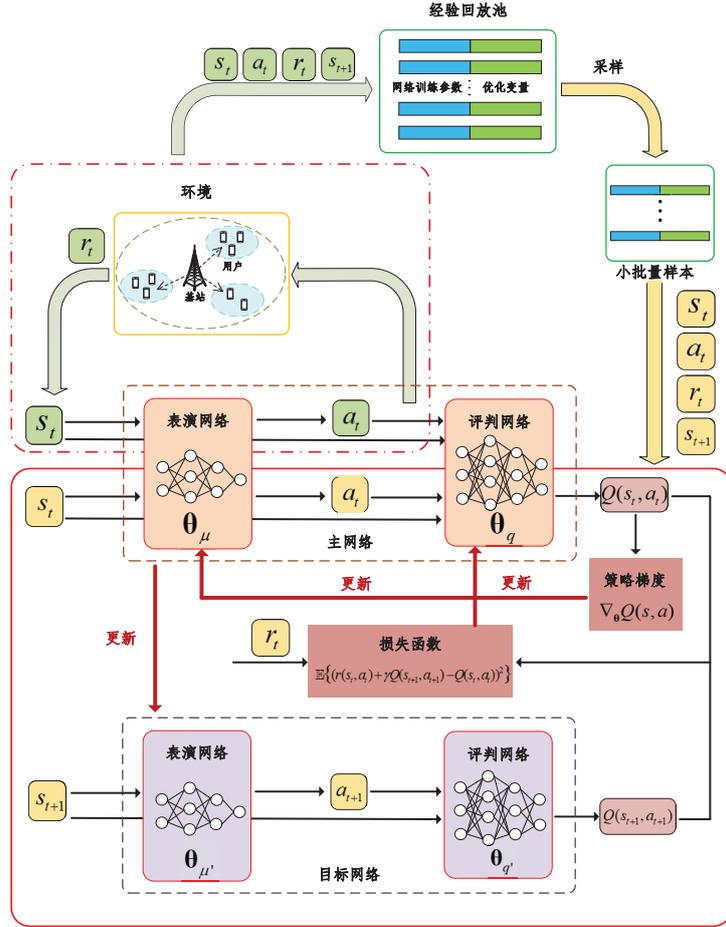


图 3.2 DDPG网络架构

3.2.2 基于确定性策略梯度的深度强化学习

3.2.2.1 MDP

MDP可以被定义为五元组 $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$ ，其中 \mathcal{S} 表示状态空间， \mathcal{A} 代表动作空间， $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ 表示奖励函数， $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ 表示状态转移函数， $\gamma \in (0, 1)$ 是折现因子。相应地， s_t ， a_t 和 r_t 分别表示第 t 时刻的状态，动作和奖励。定义 p 为转移概率，定义 π 为策略，即 $\pi : \mathcal{S} \mapsto \mathcal{A}$ 。接着，有 $a_t \sim \pi(\cdot | s_t)$ ， $s_{t+1} \sim p(\cdot | s_t, a_t)$ ， $r_t \triangleq r(s_t, a_t) \sim \mathcal{R}$ ，累计折现奖励函数 $\mathbb{E}\{\sum_{t \geq 0} \gamma^t r_t\}$ 。

3.2.2.2 确定性策略梯度

DDPG是一种结合了价值函数和策略搜索的混合DRL模型^[259-261]。融合了上述两种模型的优势，DDPG可以提升网络的收敛速度，从而更适配具有高维度的动作空间，且可以处理连续函数。如图3.2所示，DDPG包含两个基本模块^[259]：表演和评判。具体而

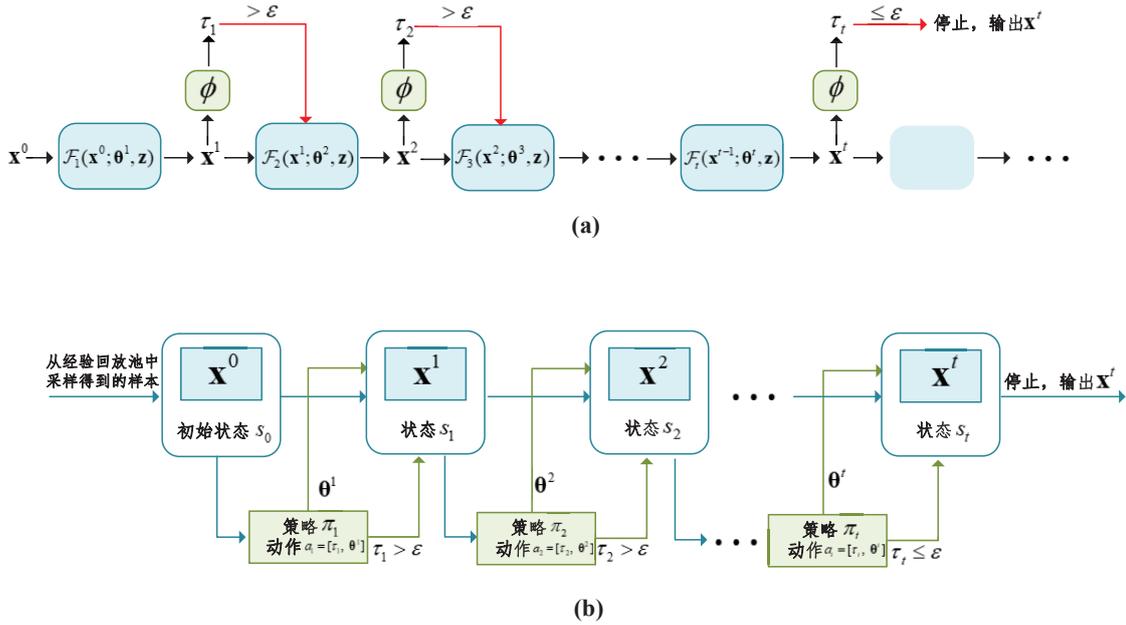


图 3.3 可变层网络框架：(a)可变层DNN框架；(b)基于DDPG的可变层模型驱动网络框架。

言，表演函数 $\pi_{\theta_{\mu}}(a|s)$ 根据当前策略将当前状态映射为一个具体动作，评判函数 $Q(s, a)$ 由深度Q学习得到，用于评估被选择的动作。针对表演和评判模块，本章分别采用经验回放池 \mathcal{D} ^[260] 和DDQN架构^[261]。具体而言，从经验回放池 \mathcal{D} 随机选取一批状态转移样本 (s_t, a_t, r_t, s_{t+1}) 用于网络训练。评判网络的训练参数 θ_q 采用SGD更新，表演网络的策略采用采样策略梯度更新^[259]。目标表演网络的训练参数 θ_{μ}^l 和目标评判网络的训练参数 θ_{q^l} 是从主表演网络 θ_{μ} 和主评判网络 θ_q 每间隔一段时间拷贝得到的。

3.2.3 可变层模型驱动网络框架

图3.3(a)描述了可变层模型驱动网络框架^[255]，其中 ϕ 是输出为 τ_t 的函数，用于指示在第 t 层是否停止。在此基础上，本章提出了基于DRL的可变层模型驱动网络架构¹。具体而言，一层模型驱动网络建模成MDP的一次状态转移，如图3.3(b)所示。深度展开网络的训练即优化每一层的训练参数 $\theta^l, \forall l$ ，这些训练参数被设计为DDPG的动作空间。将MDP建模如下。为避免混淆，分别采用 l 和 t 表示模型驱动网络的层数和DDPG的状态转移时刻。

- 智能体：基站观察到当前状态 s_t ，根据策略 π 选择一个动作 a_t ，和环境进行交互。接着，环境反馈奖励，基站根据奖励相应地调整策略 π 。旨在学习最优策略 π ，得到最优网络训练参数，最大化累计折现奖励 $\mathbb{E}\{\sum_{t \geq 0} \gamma^t r_t\}$ 。

¹采用DDPG网络结构，后文简称DDPG。

- 状态空间： \mathcal{S} 表示优化变量空间，包含变量初始值 \mathbf{x}^0 和所有中间变量 $\mathbf{x}^l, \forall l$ ，即模型驱动网络每一层中间层的输出变量。第 t 个状态转移时刻的状态 s_t ，包括对应第 l 层的模型驱动网络的输出，即 $s_t \triangleq \mathbf{x}^l$ 。
- 动作空间： \mathcal{A} 包括停止指示变量 τ 和模型驱动网络每一层的训练参数 $\boldsymbol{\theta}^l, \forall l$ 。第 t 个状态转移时刻对应的动作包括第 t 层对应的停止指示变量 τ_t 和第 t 层模型驱动网络的训练参数，即 $a_t \triangleq [\tau_t, \boldsymbol{\theta}^t]$ 。停止指示变量 $\tau_t \in [0, 1]$ 的作用是在当前层是否停止模型驱动网络的运行并输出结果。当 $\tau_t > \varepsilon$ ，则继续运行模型驱动网络的下一层，也即继续执行下一个DDPG的状态转移，其中 ε 是一个给定的超参数。否则，模型驱动网络的运行将会停止，并输出当前结果作为最终结果。
- 状态转移：状态转移函数 $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ 根据选择的动作 a_t ，将当前状态 s_t 映射为下一个状态 s_{t+1} 。状态转移过程 $s_{t+1} = p(s_t, a_t)$ 由一层或多层模型驱动网络组成，即(3.3)中的 $\mathbf{x}^{l+1} = \mathcal{F}_{l+1}(\mathbf{x}^l; \boldsymbol{\theta}^{l+1}, \mathbf{z})$ 。
- 奖励函数：在每次状态转移之后，环境根据奖励函数 $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ 反馈奖励 r_t ，其被设计为相比上一个状态对应的系统性能，当前状态转移导致的系统性能提升，即 $r_t = f(\mathbf{x}^{t-1}; \mathbf{z}) - f(\mathbf{x}^t; \mathbf{z}) - \eta$ ，其中 $f(\mathbf{x}^t; \mathbf{z})$ 表示(3.1)中的第 t 次迭代对应的目标函数。如果当前策略导致较高的性能提升，智能体将会得到更高的奖励。此外， η 是一个给定常数，表示如果当前 t 时刻没有停止，智能体将会受到一定的惩罚。如果当前DDPG的状态转移导致的系统性能提升没能超过惩罚项 η ，则智能体将获得负值奖励。通过该设计，使得学习到的策略能够随着奖励值的下降，较早地停止模型驱动网络的运行，从而实现针对不同输入样本的可变层模型驱动网络。

3.3 基于稀疏贝叶斯学习的信道估计

本节先介绍如何使用离网基底来解决方向角的失配问题。随后，建模了包含方向角失配的信道估计问题，并采用离网基底求解。此外，建模了离网参数的分布，并设计迭代SBL算法进行求解。

3.3.1 大规模MIMO信道的离网基底

本章考虑平坦衰落信道。从基站到第 k 个用户的下行信道矢量包含 N_c 簇，每簇包含 N_s 条传播路径^[164]

$$\mathbf{h}_k = \sum_{i=1}^{N_c} \sum_{j=1}^{N_s} \xi_{ij} \mathbf{a}(\phi_{ij}), \quad (3.4)$$

其中 $\xi_{ij} \sim \mathcal{CN}(0, \sigma_\alpha^2)$ 表示第 i 个簇中第 j 条路径的复增益， ϕ_{ij} 表示相应的离开角（Angle-of-Departure, AoD）²， $\mathbf{a}(\phi_{ij})$ 表示导向矢量。对于一个有 N 条天线的均匀线性阵列，导向矢量为

$$\mathbf{a}(\phi) = \frac{1}{\sqrt{N}} [1, e^{-j2\pi \frac{d}{\lambda} \sin(\phi)}, \dots, e^{-j2\pi \frac{d}{\lambda} (N-1) \sin(\phi)}]^T, \quad (3.5)$$

其中 d 和 λ 分别表示相邻天线之间的距离和相邻子载波之间的间隔。

基站装配有一个均匀线性阵列，传输一串长度为 T 的导频符号，用 $\mathbf{X} \in \mathbb{C}^{T \times N}$ 表示，发送给每个用户用于信道估计。第 k 个用户接收到的导频信号 $\mathbf{y}_k \in \mathbb{C}^{T \times 1}$ 为

$$\mathbf{y}_k = \mathbf{X} \mathbf{h}_k + \mathbf{n}_k, \quad (3.6)$$

其中 $\mathbf{n}_k \in \mathbb{C}^{T \times 1} \sim \mathcal{CN}(\mathbf{0}, \sigma^2)$ 表示加性高斯白噪声（Additive White Gaussian Noise, AWGN），且 $\text{Tr}(\mathbf{X} \mathbf{X}^H) = PTN$ 。因为基站端的天线数量 N 较大，很难使用传统算法，比如最小二乘等，来精确估计和恢复信道 \mathbf{h}_k 。因此，提出采用SBL算法，通过发送少量有限的导频精确估计信道。

为表述清楚起见，后文略去用户的脚标 k ，真实的AoD表示为 $\{\phi_j, j = 1, 2, \dots, J\}$ ，其中 $J = N_c N_s$ 。令 $\{\hat{\phi}_j\}_{j=1}^{\hat{J}}$ 为固定采样的网格点，均匀覆盖了角度域 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ ，其中 \hat{J} 表示网格点的数量。当网格足够精细，所有真实的AoD，即 $\{\phi_j, j = 1, 2, \dots, J\}$ 刚好落在网格点上，可以采用如下模型对 \mathbf{h} 进行建模

$$\mathbf{h} = \mathbf{A} \mathbf{w}, \quad (3.7)$$

其中 $\mathbf{A} = [\mathbf{a}(\hat{\phi}_1), \mathbf{a}(\hat{\phi}_2), \dots, \mathbf{a}(\hat{\phi}_{\hat{J}})] \in \mathbb{C}^{N \times \hat{J}}$ ， $\mathbf{a}(\phi)$ 表示导向矢量， $\mathbf{w} \in \mathbb{C}^{\hat{J} \times 1}$ 为一个稀疏矢量，它的非零元素对应真实的方向角 $\{\phi_j, j = 1, 2, \dots, J\}$ 。如果 \mathbf{w} 的第 \hat{j} 个元素非零，对应真实的方向角 ϕ_j ，则 $\phi_j = \hat{\phi}_{\hat{j}}$ 。

然而，假定真实的方向角刚好落在预先设定的网格点上是不切实际的。为解决方向角失配的问题，采用离网模型。具体而言，当 $\phi_j \notin \{\hat{\phi}_j\}_{j=1}^{\hat{J}}$ ，以及 $\hat{\phi}_{n_j}, n_j \in \{1, 2, \dots, \hat{J}\}$ 是距离 ϕ_j 最近的网格点，可以将 ϕ_j 表述为

$$\phi_j = \hat{\phi}_{n_j} + \beta_{n_j}, \quad (3.8)$$

²方向角包含到达角和离开角，本章仅考虑离开角，可直接拓展到包含到达角的场景。

其中 β_{n_j} 表示离网距离参数。根据(3.8)，可得 $\mathbf{a}(\phi_j) = \mathbf{a}(\hat{\phi}_{n_j} + \beta_{n_j})$ 。随后， \mathbf{h} 可以被表示为

$$\mathbf{h} = \mathbf{A}(\boldsymbol{\beta})\mathbf{w}, \quad (3.9)$$

其中 $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_j]^T$, $\mathbf{A}(\boldsymbol{\beta}) = [\mathbf{a}(\hat{\phi}_1 + \beta_1), \mathbf{a}(\hat{\phi}_2 + \beta_2), \dots, \mathbf{a}(\hat{\phi}_j + \beta_j)]$ ，以及

$$\beta_{n_j} = \begin{cases} \phi_j - \hat{\phi}_{n_j}, & l = 1, 2, \dots, L, \\ 0, & \text{其他情况.} \end{cases} \quad (3.10)$$

引入离网基底，模型显著降低了由方向角失配问题带来的影响，因为总是存在 β_{n_j} 使得(3.8)满足。随后，用户接收到的导频信号可以写成

$$\mathbf{y} = \mathbf{X}\mathbf{A}(\boldsymbol{\beta})\mathbf{w} + \mathbf{n} = \boldsymbol{\Phi}(\boldsymbol{\beta})\mathbf{w} + \mathbf{n}, \quad (3.11)$$

其中 $\boldsymbol{\Phi}(\boldsymbol{\beta}) \triangleq \mathbf{X}\mathbf{A}(\boldsymbol{\beta})$ 。因此，该信道估计问题可以建模为

$$\min_{\mathbf{w}} \|\mathbf{w}\|_0, \quad \text{s.t.} \|\mathbf{y} - \mathbf{X}\mathbf{A}(\boldsymbol{\beta})\mathbf{w}\|_2 < \varsigma, \quad (3.12)$$

其中 ς 是一个依赖于 $\|\mathbf{n}\|_2$ 的常数。由于系数矢量 $\boldsymbol{\beta}$ 是未知的，已有的 l_1 范数最小化算法，并不能用于直接求解问题(3.12)。接下来，建模离网参数的分布，并提出基于SBL的迭代算法来联合优化待恢复的稀疏信道和离网参数。

3.3.2 稀疏贝叶斯学习框架

受到文献[164]的启发，参数的分布建模如下。基于圆对称复高斯噪声的假设，可得

$$p(\mathbf{y}|\mathbf{w}, \alpha, \boldsymbol{\beta}) = \mathcal{CN}(\mathbf{y}|\boldsymbol{\Phi}(\boldsymbol{\beta})\mathbf{w}, \alpha^{-1}\mathbf{I}), \quad (3.13)$$

其中 $\alpha = \sigma_n^{-2}$ 表示噪声精度。因为 α 通常是未知的，将其建模为Gamma先验分布，即 $p(\alpha) = \Gamma(\alpha; 1 + a_1, b_1)$ ，其中借鉴文献[165]，设定 $a_1, b_1 \rightarrow 0$ ，以后的较广的先验分布，避免由于先验分布过窄导致的估计误差。随后，假定 $\boldsymbol{\beta}$ 的每个元素为没有先验信息的IID均匀先验分布

$$p(\boldsymbol{\beta}) = \prod_{j=1}^J U(\beta_j; -\frac{\pi}{2}, \frac{\pi}{2}). \quad (3.14)$$

根据文献[165]中的稀疏贝叶斯模型，假定矢量 \mathbf{w} 的每个元素的先验分布为参数 γ_i 的高斯分布。令 $\boldsymbol{\gamma} = [\gamma_1, \gamma_2, \dots, \gamma_j]^T$ ，可以得到

$$p(\mathbf{w}|\boldsymbol{\gamma}) = \mathcal{CN}(\mathbf{w}|\mathbf{0}, \text{diag}(\boldsymbol{\gamma}^{-1})). \quad (3.15)$$

随后，将 γ 中的每个元素建模为IID的Gamma分布

$$p(\gamma) = \prod_{j=1}^J \Gamma(\gamma_j; 1 + a_2, b_2). \quad (3.16)$$

因此，上述两阶段多层次的先验分布可以描述为

$$p(\mathbf{w}) = \int_0^\infty p(\mathbf{w}|\gamma)p(\gamma)d\gamma \propto \prod_{i=1}^J (b_2 + |w_i|^2)^{-(a_2 + \frac{3}{2})}, \quad (3.17)$$

由于该分布具有较长的拖尾，促使变量 \mathbf{w} 趋于稀疏化，且由于 b_2 的值往往较小，其在0点具有较高的尖峰^[165]。

注意到(3.15)中的 γ_j 指示了变量 \mathbf{w} 的支撑集。比如，当 γ_j 的值较大， \mathbf{w} 的第 j 个元素往往容易趋于0。反之， \mathbf{w} 的第 j 个元素的值往往会比较大。因此，当获得精确的矢量 γ 和离网距离参数 β ，被估计的信道可以表示为

$$\hat{\mathbf{h}} = \mathbf{A}_\Omega(\beta)(\Phi_\Omega(\beta))^\dagger \mathbf{y}, \quad (3.18)$$

其中 $\Omega \triangleq \text{supp}(\mathbf{w})$ 。因此，旨在寻找最优的变量 β 和 γ 。因为噪声精度 α 是未知的，通过最大化后验概率 $p(\alpha, \gamma, \beta|\mathbf{y})$ 来寻找变量 α^* ， γ^* 和 β^* 的最优值，即

$$(\alpha^*, \gamma^*, \beta^*) = \arg \max_{\alpha, \gamma, \beta} \ln p(\mathbf{y}, \alpha, \gamma, \beta). \quad (3.19)$$

由于目标函数(3.19)是高维且非凸的，且梯度下降算法收敛速度很慢、容易陷入性能较差的局部最优解，所以直接使用梯度下降算法求解并不实际。此外，原始目标函数(3.19)的梯度没有闭式解。为了解决这些问题，设计了基于SBL的迭代算法，该算法可以保证以较快的速度收敛到问题(3.19)的平稳点^[164]。

3.3.3 基于稀疏贝叶斯学习的迭代算法

3.3.3.1 逐块MM算法框架

著名的MM算法可以高效求解非凸问题并加快收敛速度^[262]，其旨在通过迭代的方式为原始目标函数 $\ln p(\mathbf{y}, \alpha, \gamma, \beta)$ 在每次迭代过程中建立一个连续替代函数，交替地优化变量 α ， γ 和 β ，从而最大化替代函数

$$\alpha^{(t+1)} = \arg \max_{\alpha} \mathcal{G}(\alpha, \gamma^{(t)}, \beta^{(t)} | \alpha^{(t)}, \gamma^{(t)}, \beta^{(t)}), \quad (3.20a)$$

$$\gamma^{(t+1)} = \arg \max_{\gamma} \mathcal{G}(\alpha^{(t+1)}, \gamma, \beta^{(t)} | \alpha^{(t+1)}, \gamma^{(t)}, \beta^{(t)}), \quad (3.20b)$$

$$\boldsymbol{\beta}^{(t+1)} = \arg \max_{\boldsymbol{\beta}} \mathcal{G}(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\beta}^{(t)}), \quad (3.20c)$$

其中 $(\cdot)^{(t)}$ 表示第 t 次迭代， $\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})$ 表示在当前迭代点 $(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})$ 建立的替代函数，需要满足以下要求：（1）替代函数为函数 $\ln p(\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})$ 的下界；（2）替代函数在当前点 $(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})$ ，该替代函数的值和关于变量 $\boldsymbol{\alpha}$ ， $\boldsymbol{\gamma}$ 和 $\boldsymbol{\beta}$ 的梯度值等于 $\ln p(\mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})$ 的函数值和梯度值。变量的更新迭代表达式(3.20a)-(3.20c)可以确保逐块MM算法的收敛性^[262]。

3.3.3.2 替代函数

受到期望最大化（Expectation-Maximization, EM）算法思想的启发^[262]，在当前迭代点 $(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})$ 建立替代函数

$$\mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta} | \boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)}) = \int p(\mathbf{w} | \mathbf{y}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)}) \ln \frac{p(\mathbf{w}, \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})}{p(\mathbf{w} | \mathbf{y}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})} d\mathbf{w}. \quad (3.21)$$

根据(3.13)和(3.15), $p(\mathbf{w} | \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})$ 为复高斯分布

$$p(\mathbf{w} | \mathbf{y}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) = \mathcal{CN}(\mathbf{w} | \boldsymbol{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}), \boldsymbol{\Sigma}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})), \quad (3.22)$$

其中

$$\begin{aligned} \boldsymbol{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) &= \boldsymbol{\alpha} \boldsymbol{\Sigma}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) \boldsymbol{\Phi}^H(\boldsymbol{\beta}) \mathbf{y}, \\ \boldsymbol{\Sigma}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) &= (\boldsymbol{\alpha} \boldsymbol{\Phi}^H(\boldsymbol{\beta}) \boldsymbol{\Phi}(\boldsymbol{\beta}) + \text{diag}(\boldsymbol{\gamma}))^{-1}. \end{aligned} \quad (3.23)$$

接下来，推导关于变量 $\boldsymbol{\alpha}$ 、 $\boldsymbol{\gamma}$ 和 $\boldsymbol{\beta}$ 的迭代优化闭式解。

3.3.3.3 求解变量 $\boldsymbol{\alpha}$

关于变量 $\boldsymbol{\alpha}$ 的优化问题(3.20a)有如下闭式解

$$\boldsymbol{\alpha}^{(t+1)} = \frac{T + a}{b + \eta(\boldsymbol{\alpha}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})}, \quad (3.24)$$

其中

$$\eta(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) = \text{Tr}(\boldsymbol{\Phi}(\boldsymbol{\beta}) \boldsymbol{\Sigma}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) \boldsymbol{\Phi}^H(\boldsymbol{\beta})) + \|\mathbf{y} - \boldsymbol{\Phi}(\boldsymbol{\beta}) \boldsymbol{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta})\|^2. \quad (3.25)$$

3.3.3.4 求解变量 $\boldsymbol{\gamma}$

关于变量 $\boldsymbol{\gamma}$ 的优化问题(3.20b)有如下闭式解

$$\gamma_j^{(t+1)} = \frac{a + 1}{b + [\boldsymbol{\Lambda}(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\beta}^{(t)})]_{jj}}, \forall j, \quad (3.26)$$

其中

$$\boldsymbol{\Lambda}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) = \boldsymbol{\Sigma}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) + \boldsymbol{\mu}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}) \boldsymbol{\mu}^H(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}). \quad (3.27)$$

3.3.3.5 求解变量 β

由于优化问题(3.20c)是非凸的，难以直接找到全局最优解。因此，采用针对目标函数的梯度更新，得到变量 β 的一步更新迭代表达式。目标函数(3.20c)关于变量 β 的梯度表达式为

$$\Xi_{\beta}^{(t)} = [\Xi^{(t)}(\beta_1), \Xi^{(t)}(\beta_2), \dots, \Xi^{(t)}(\beta_j)]^T, \quad (3.28)$$

其中

$$\Xi^{(t)}(\beta_j) = 2\Re \left\{ (\mathbf{a}'(\hat{\phi}_j + \beta_j))^H \mathbf{X}^H \mathbf{X} (\mathbf{a}(\hat{\phi}_j + \beta_j)) \right\} \cdot c_1^{(t)} + 2\Re \left\{ (\mathbf{a}'(\hat{\phi}_j + \beta_j))^H \mathbf{X}^H \mathbf{c}_2^{(t)} \right\}, \quad (3.29)$$

以及

$$c_1^{(t)} \triangleq -\alpha^{(t+1)}(\chi_{jj}^{(t)} + |\mu_j^{(t)}|^2), \quad (3.30a)$$

$$\mathbf{c}_2^{(t)} \triangleq \alpha^{(t+1)} \left(\mu_j^{(t)} \mathbf{y}_{-j}^{(t)} - \mathbf{X} \sum_{i \neq j} \chi_{ij}^{(t)} \mathbf{a}(\hat{\phi}_i + \beta_i) \right), \quad (3.30b)$$

$$\mathbf{y}_{-j}^{(t)} \triangleq \mathbf{y} - \mathbf{X} \cdot \sum_{i \neq j} (\mu_i^{(t)} \cdot \mathbf{a}(\hat{\phi}_i + \beta_i)), \quad (3.30c)$$

$$\mathbf{a}'(\hat{\phi}_i + \beta_i) \triangleq d\mathbf{a}(\hat{\phi}_i + \beta_i)/d\beta_i, \quad (3.30d)$$

$\mu_j^{(t)}$ 和 $\chi_{ij}^{(t)}$ 分别表示 $\boldsymbol{\mu}(\alpha^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\beta}^{(t)})$ 的第 j 个元素和 $\boldsymbol{\Sigma}(\alpha^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\beta}^{(t)})$ 的第 (i, j) 个元素。

关于公式(3.24)-(3.29)的推导可以根据基本的数理统计和微积分知识得到，具体细节可参考文献[164]。关于变量 β 问题非凸，很难获得变量 β 的最优解，因此只需要寻找到其次优解，保证每次迭代目标函数的值递增即可。因此，采用梯度下降算法更新变量 β

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \Delta_{\beta} \cdot \Xi_{\beta}^{(t)}, \quad (3.31)$$

其中 Δ_{β} 表示步长。关于该基于SBL的迭代算法的执行步骤和具体细节请见算法表格3.1。

3.4 模型驱动信道估计网络

本节将上述基于SBL的迭代算法展开为类似于DNN的逐层结构并提供理论性能分析。

3.4.1 基于稀疏贝叶斯学习算法的模型驱动网络

如图3.4所示，将基于SBL的迭代算法展开为类似于DNN的逐层结构，其中 $\mathcal{F}_l(\cdot, \Theta)$ ， $\mathcal{U}_l(\cdot, \Theta)$ 和 $\mathcal{W}_l(\cdot, \Theta)$ 分别表示第 l 层输出 α ， $\boldsymbol{\gamma}$ 和 $\boldsymbol{\beta}$ 的模块。为表述清楚起见，下文略去每层的

表 3.1 基于SBL的迭代优化算法

算法 3.1 基于SBL的迭代优化算法

- 1: **输入**: 导频矩阵 \mathbf{X} , 噪声方差 σ^2 , 传输功率 P , 导频长度 T , 天线数 N , 网格点数 \hat{J} , 以及需要达到的信道估计精度 δ ;
- 2: **输出**: 恢复得到的信道 $\hat{\mathbf{h}}^t$;
- 3: **初始化**: 变量 α , γ 和 β , 设当前迭代次数 $t = 1$;
- 4: **while** $\|\hat{\mathbf{h}}^t - \hat{\mathbf{h}}^{t-1}\|^2 > \delta$ **do**
- 5: 根据公式(3.24)更新变量 α ;
- 6: 根据公式(3.26)更新变量 γ ;
- 7: 根据公式(3.31)更新变量 β ;
- 8: 根据公式(3.18)恢复信道 $\hat{\mathbf{h}}^t$;
- 9: $t = t + 1$.
- 10: **end while**

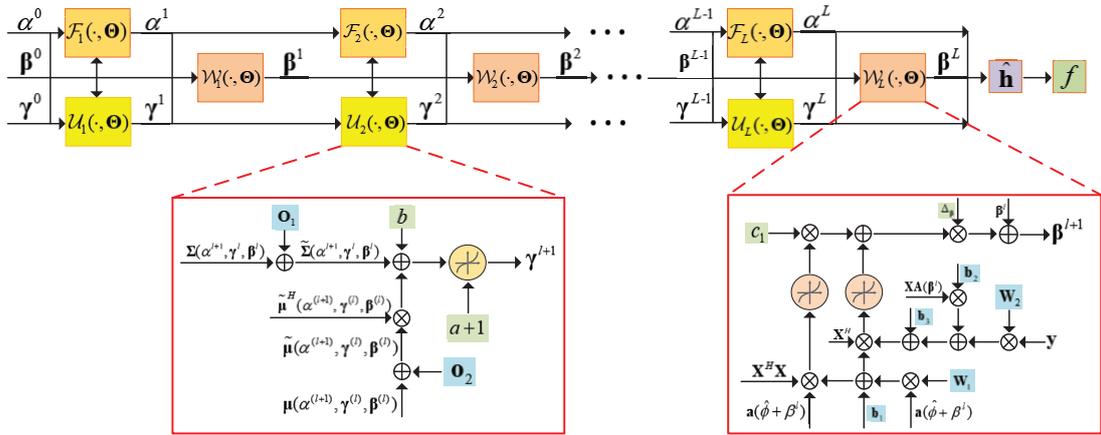


图 3.4 基于SBL算法的用于信道估计的模型驱动网络结构

脚标 l 。训练参数 $\Theta = \{\Theta_1, \Theta_2\}$ 可被分为两类：(1) 先验分布中的超参数和迭代算法中的步长参数，这些超参数往往较难取值，可通过训练得到，即 $\Theta_1 = \{a, b, c_1, \Delta_\beta\}$ ；(2) 引入的用于替换高计算复杂度操作的训练参数，同时可用于提升模型驱动网络的自由度和性能，即 $\Theta_2 = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{O}_1, \mathbf{o}_2, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ 。模型驱动网络结构及表达式如下式(3.32)所示。

$$\alpha^{(l+1)} = (T + a) \left(b + \text{Tr} \left(\Phi(\beta^{(l)}) \tilde{\Sigma}(\alpha^{(l)}, \gamma^{(l)}, \beta^{(l)}) \Phi^H(\beta^{(l)}) \right) + \|\mathbf{y} - \Phi(\beta^{(l)}) \tilde{\mu}(\alpha^{(l)}, \gamma^{(l)}, \beta^{(l)})\|_2^2 \right)^{-1}, \quad (3.32a)$$

$$\gamma_j^{(l+1)} = (a+1) \left(b + [\tilde{\Sigma}(\alpha^{(l+1)}, \gamma^{(l)}, \beta^{(l)}) \right. \quad (3.32b)$$

$$\left. + \tilde{\boldsymbol{\mu}}(\alpha^{(l+1)}, \gamma^{(l)}, \beta^{(l)}) \tilde{\boldsymbol{\mu}}^H(\alpha^{(l+1)}, \gamma^{(l)}, \beta^{(l)}) \right]_{jj}^{-1}, \forall j, \quad (3.32c)$$

$$\beta_j^{(l+1)} = \beta_j^{(l)} + \Delta_{\beta_j}^{(l)} \cdot \left\{ \Re \left((\mathbf{W}_1 \mathbf{a}(\hat{\phi}_j + \beta_j) + \mathbf{b}_1)^H \mathbf{X}^H \mathbf{X} (\mathbf{a}(\hat{\phi}_j + \beta_j)) \right) c_1^{(l)} \right. \\ \left. + \Re \left((\mathbf{W}_1 \mathbf{a}(\hat{\phi}_j + \beta_j) + \mathbf{b}_1)^H \mathbf{X}^H (\mathbf{W}_2 \mathbf{y} + \mathbf{X} \mathbf{A}(\beta) \mathbf{b}_2 + \mathbf{b}_3) \right) \right\}, \forall j. \quad (3.32d)$$

从如下几个方面引入训练参数 Θ_2 :

- 因为 $\Sigma(\alpha, \gamma, \beta)$ 和 $\boldsymbol{\mu}(\alpha, \gamma, \beta)$ 往往是难以精确估计得到的，所以并不能直接精确描述概率分布 $p(\mathbf{w}|\mathbf{y}, \alpha, \gamma, \beta)$ 的统计特性。为了提升精度，引入训练参数 \mathbf{O}_1 和 \mathbf{o}_2 ，即 $\tilde{\Sigma}(\alpha, \gamma, \beta) = \Sigma(\alpha, \gamma, \beta) + \mathbf{O}_1$ 和 $\tilde{\boldsymbol{\mu}}(\alpha, \gamma, \beta) = \boldsymbol{\mu}(\alpha, \gamma, \beta) + \mathbf{o}_2$ 。
- 注意到 $\mathbf{a}'(\hat{\phi}_i + \beta_i) = d\mathbf{a}(\hat{\phi}_i + \beta_i)/d\beta_i$ ，矢量 $\mathbf{a}(\hat{\phi}_i + \beta_i)$ 的元素形式为 $e^{-j2\pi\sin(\hat{\phi}_i + \beta_i)}$ ，一阶导数的形式为 $-j2\pi\cos(\hat{\phi}_i + \beta_i)e^{-j2\pi\sin(\hat{\phi}_i + \beta_i)}$ 。因此， $\mathbf{a}'(\hat{\phi}_i + \beta_i)$ 可以被写成 $\Phi \mathbf{a}(\hat{\phi}_i + \beta_i)$ ，其中 Φ 由 $-j2\pi\cos(\hat{\phi}_i + \beta_i)$ 组成。将 $\mathbf{a}'(\hat{\phi}_i + \beta_i)$ 替换为 $\mathbf{W}_1 \mathbf{a}(\hat{\phi}_i + \beta_i) + \mathbf{b}_1$ ，其中 \mathbf{W}_1 和 \mathbf{b}_1 为引入的训练参数。
- 回顾 \mathbf{c}_2 的表达式(3.30b) 以及 $\mathbf{A}(\beta) = [\mathbf{a}(\hat{\phi}_1 + \beta_1), \mathbf{a}(\hat{\phi}_2 + \beta_2), \dots, \mathbf{a}(\hat{\phi}_j + \beta_j)]$ 。注意到 $\mathbf{X} \sum_{i \neq j} \chi_{ij}^{(l)} \mathbf{a}(\hat{\phi}_i + \beta_i)$ 可被记作 $\mathbf{X} \mathbf{A}(\beta) \boldsymbol{\chi}$ ，其中 $\boldsymbol{\chi}$ 由 χ_{ij} 组成。因此，采用结构 $\mathbf{W}_2 \mathbf{y} + \mathbf{X} \mathbf{A}(\beta) \mathbf{b}_2 + \mathbf{b}_3$ 来替换 \mathbf{c}_2 ，其中 \mathbf{W}_2 ， \mathbf{b}_2 和 \mathbf{b}_3 为引入的训练参数。
- 导频信号矩阵 \mathbf{X} 被视为训练参数，因为训练得到的导频信号矩阵 \mathbf{X} 能更好地适配当前的信道统计特性，从而以更短的导频长度实现更高的信道估计精度。

得到优化变量 α ， γ 和 β 的值之后，用公式(3.18)重构信道。此外，采用如下 NMSE 作为模型驱动网络的损失函数

$$\frac{1}{|\mathcal{M}|} \sum_{m=1}^{|\mathcal{M}|} \frac{\|\hat{\mathbf{h}}_m - \mathbf{h}_m\|^2}{\|\mathbf{h}_m\|^2}, \quad (3.33)$$

其中 $|\mathcal{M}|$ 为训练数据集中样本的数量， $\hat{\mathbf{h}}_m$ 为第 m 个真实信道样本 \mathbf{h}_m 的估计。此外，在下一小节，提供该网络性能的理论分析。

3.4.2 模型驱动网络的性能分析

通常，由于不同模型驱动网络引入的训练参数不同、结构不同、基于的迭代算法不同等，提供一般化的模型驱动网络性能分析比较困难。本节提出如下定理，用于分析本章提出的模型驱动网络的性能：

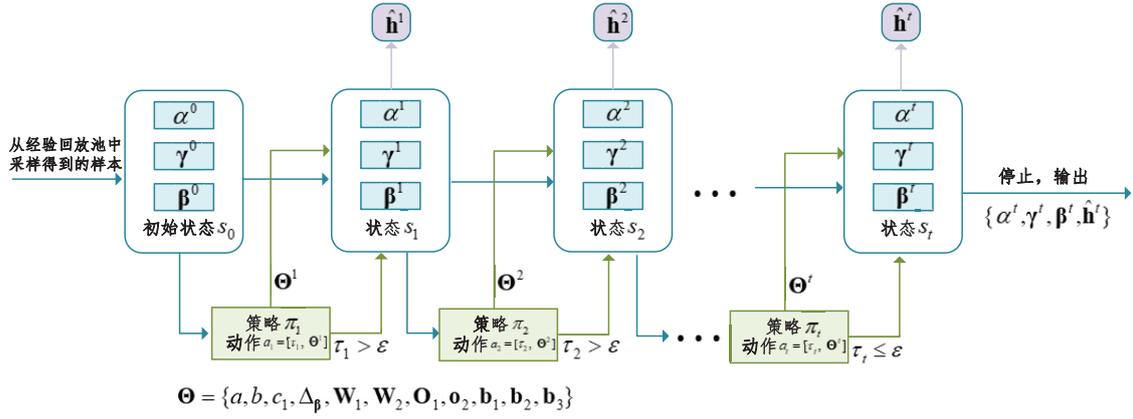


图 3.5 基于DDPG的用于展开迭代SBL算法的可变层模型驱动网络结构

定理 3.1. 存在训练参数确保一层该模型驱动网络实现的性能可以逼近原迭代算法两次迭代的性能。

证明. 见附录A.2.1。 □

该证明可直接拓展到证明如下结论：存在训练参数确保一层该模型驱动网络实现的性能可以逼近原迭代算法多次迭代的性能。因此，提出的模型驱动网络可以用远少于原迭代算法的层数实现逼近原迭代算法的性能。此外，需要的网络层数随着输入信道样本的变化而变化，本章提出的具有自适应层数的模型驱动网络设计可用于找到针对每个信道样本最优的层数，从而针对每个样本设计最优层数，且以更少的层数逼近原迭代算法的性能。

3.5 基于深度强化学习的模型驱动信道估计网络

本节提出基于DDPG的可变层模型驱动信道估计网络，并设计可用于动态控制信道估计误差的停止分数。

3.5.1 基于DDPG的可变层模型驱动信道估计网络

如图3.5所示，基于DDPG的可变层模型驱动框架可用于求解该问题。相应的MDP建模如下。

- **状态空间：**由该问题包含的所有优化变量 $\{\alpha, \gamma, \beta\}$ 组成，其中包括模型驱动网络的初始化变量 $\{\alpha^0, \gamma^0, \beta^0\}$ 和所有模型驱动网络的中间层输出变量 $\{\alpha^l, \gamma^l, \beta^l, \forall l\}$ 。具体而言，第 t 时刻的状态为 $s_t \triangleq [\alpha^t, \gamma^t, \beta^t]$ ，即对应某层模型驱动网络的输出变量。

- 动作空间：由每层的停止指示变量 τ 和网络训练参数 $\{\Theta^l, \forall l\}$ 组成。具体而言， t 时刻的动作定义为 $a_t \triangleq [\tau_t, \Theta^t]$ 。停止指示变量 $\tau_t \in [0, 1]$ 的作用是决定是否在当前层停止运行模型驱动网络，并将该层的输出结果作为最终结果，且 τ_t 是通过DDPG直接学出来的。当 $\tau_t > \varepsilon$ ，基于DDPG的模型驱动网络将会继续运行，其中 ε 是给定的超参数。否则，基于DDPG的模型驱动网络将在当前层停止运行，并将该层的输出变量（即DDPG的当前状态）作为最终结果。
- 状态转移：根据当前状态 $s_t = [\alpha^t, \gamma^t, \beta^t]$ 和选择的动作 $a_t = [\tau_t, \Theta^t]$ ，当 $\tau_t > \varepsilon$ 时，状态转移过程由一层或多层模型驱动网络(3.32)组成。比如，将状态转移过程定义为由一层模型驱动网络组成，状态转移过程写作 $[\alpha^l, \gamma^l, \beta^l] \rightarrow [\alpha^{l+1}, \gamma^{l+1}, \beta^{l+1}]$ 。当 $\tau_t \leq \varepsilon$ ，基于DDPG的模型驱动网络将停止运行，并将当前状态视为最终的结果输出。
- 奖励函数：奖励函数设计为网络的上一层和当前层的输出变量对应的NMSE性能的增量

$$r_t = \frac{\|\hat{\mathbf{h}}^{t-1} - \mathbf{h}\|^2}{\|\mathbf{h}\|^2} - \frac{\|\hat{\mathbf{h}}^t - \mathbf{h}\|^2}{\|\mathbf{h}\|^2} - \eta, \quad (3.34)$$

其中 $\hat{\mathbf{h}}^t$ 为真实信道 \mathbf{h} 在DDPG网络第 t 时刻（即对应模型驱动网络第 l 层）的估计。如果当前网络的策略能获得两层之间较高的性能提升，则表明当前网络参数训练得较好，会收到较高的奖励值。此外， η 是常数，表示如果网络在当前 t 时刻不停止，将给予惩罚。如果当前DDPG的状态转移导致的系统性能提升没能超过惩罚项 η ，则智能体将收到负值奖励。通过该设计，使得学习到的策略能够随着奖励值的下降，较早地停止模型驱动网络的运行，从而实现针对不同输入样本的可变层模型驱动网络。

结合DDPG的性能分析^[259-261]和章节3.4.2中模型驱动网络的性能分析，可以得出结论：本章节提出的基于DDPG的模型驱动信道估计网络可以实现针对每个信道样本的可变层，且可以更少的层数实现逼近基于SBL的迭代算法的性能。

3.5.2 基于DDPG的可变层模型驱动网络的改进

上述针对停止指示变量 τ_t 的设计是一种一般化的方法。对于信道估计问题，上述方法并不能精确动态地控制信道估计误差 $\|\mathbf{h} - \hat{\mathbf{h}}^t\|^2$ ，且超参数 ε 很难确定。为了解决这些问题，本章节针对信道估计问题，提出一种更为精确有效的方法设计停止指示变量 τ_t 。

3.5.2.1 停止分数

首先，引入停止分数，即在DDPG网络第 t 时刻（对应模型驱动网络某一层） $L_t \in [0, 1]$ 用于指示何时停止。希望停止分数和信道重建误差 $\|\mathbf{h} - \hat{\mathbf{h}}^t\|^2$ 可以建立联系，从而可以动态精确地通过控制层数，来控制信道估计误差。然而，网络并不知道真实的信道 \mathbf{h} 。为了得到信道估计误差，考虑使用 $\|\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t\|^2$ ，该公式被广泛应用于多个算法的停止准则^[255]。本章节采用 $\|\mathbf{Q}(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)\|^2$ 作为第 t 层信道估计误差的近似，其中 \mathbf{Q} 是针对估计误差的线性映射： $(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t) = \mathbf{X}(\mathbf{h} - \hat{\mathbf{h}}^t)$ 。由于难以直接得到精确的映射矩阵 \mathbf{Q} ，将 \mathbf{Q} 设为DDPG网络的训练参数，和DDPG网络一起训练，从而使得 \mathbf{Q} 更适配当前信道的统计特性。因此，停止分数可以设计为

$$L_t = \sigma(p_1 \|\mathbf{Q}(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)\|^2 + p_2), \quad (3.35)$$

其中 $p_1 > 0$ ， p_2 和 \mathbf{Q} 是训练参数， $\sigma(\cdot)$ 为Sigmoid函数，输出的值在0到1之间， $p_1 > 0$ 确保停止分数 L_t 随着信道估计误差 $\|\mathbf{Q}(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)\|^2$ 的降低而减小。根据(3.35)进行拓展，采用一个 r 层FNN学习停止分数

$$L_t = \sigma_r(\mathbf{q}_r^T (\cdots \sigma_2(\mathbf{Q}_2 \sigma_1(\mathbf{Q}_1(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t) + \mathbf{p}_1) + \mathbf{p}_2) \cdots) + p_r), \quad (3.36)$$

其中 $\mathbf{p}_i, \mathbf{Q}_i, \forall i = 1, 2, \dots, r$ 为训练参数，引入矢量参数 \mathbf{q}_r^T 和标量参数 p_r 用于确保该网络的输出为一个标量。该网络的输入和输出分别为信道估计误差 $(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)$ 和停止分数 L_t 。

如果针对该网络有一个精心设计的合理的损失函数，网络参数可以被更好地训练，从而更好地适配信道估计误差的统计特性，更精确地估计 $\|\mathbf{h} - \hat{\mathbf{h}}^t\|^2$ 。由此，设计下述可微分的损失函数

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_t \frac{\|\mathbf{h} - \hat{\mathbf{h}}^t\|^2}{L_t} + \rho L_t, \quad (3.37)$$

其中 $\rho \geq 0$ 为正则参数， $\boldsymbol{\theta}$ 表示该DNN的训练参数， \mathbf{h} 为真实的信道，在训练阶段假设已知。不同于迭代算法和现有的模型驱动网络，它们中间层的输出结果对损失函数没有显式的贡献。而这里的求和是从第一层一直加和到当前第 t 层，即中间层的输出结果对损失函数有贡献和影响。具体而言，每一层网络输出的估计信道 $\hat{\mathbf{h}}^t$ 和停止分数 L_t 都对损失函数(3.37)有影响。令损失函数(3.37)关于 L_t 的导数为0，得到最优的停止分数为

$$L_t = \frac{\|\mathbf{h} - \hat{\mathbf{h}}^t\|}{\sqrt{\rho}}. \quad (3.38)$$

由此可见，一个训练良好的网络如果能产生最优的停止分数，则该停止分数正比于信道估计误差。网络层数取决于停止分数第一次小于 ϵ 对应的网络层数，即

$$T_s = \min\{t : L_t \leq \epsilon\}, \quad (3.39)$$

其中 ϵ 为一个预先给定的较小的常数。改变常数 ϵ 的值通常会导致可变层模型驱动网络的层数变化，以及对应信道估计误差的变化。注意到 L_t 和信道估计误差以及正则参数 ρ 相关。此外，(3.37)中的导数在 $L_t = \frac{\|\mathbf{h} - \hat{\mathbf{h}}^t\|}{\sqrt{\rho}}$ 等于0。因此，可以通过调整超参数 ϵ 和 ρ 实现不同的信道估计误差。比如，如果希望当信道估计误差为 $\|\mathbf{h} - \hat{\mathbf{h}}^t\|^2 = 0.01$ 时停止，可以设置超参数为 $\rho = 1$ 和 $\epsilon = 0.1$ 。

3.5.2.2 包含停止分数的可变层模型驱动网络

本小节采用章节3.5.2.1提出的停止时间 L_t 作为 τ_t ，章节3.5.1中设计的MDP需要做如下修改

- 已知网络(3.36)的输入是信道估计误差 $(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)$ ，且DDPG的表演网络的输入为状态 s_t 。因此， $(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)$ 需要被设计为状态的一部分，即 $s_t \triangleq [\alpha^t, \gamma^t, \beta^t, (\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)]$ 。具体而言， $(\mathbf{y} - \mathbf{X}\hat{\mathbf{h}}^t)$ 将和 $[\alpha^t, \gamma^t, \beta^t]$ 一起输入DDPG的表演网络。
- 已知网络(3.36)的输出是停止分数 L_t ，且 L_t 是动作的一部分，即 $a_t \triangleq [L_t, \Theta^t]$ ，其中动作 a_t 为DDPG的表演网络的输出。因此，网络(3.36)可以被视为DDPG的表演网络的子网络。该网络输出 L_t ，用于指示网络当前是否停止运行并输出最终结果。
- 奖励函数将被修改为(3.34)中的奖励函数和损失函数(3.37)的加权和。具体而言，(3.34)确保了信道估计误差逐层显著递减，而(3.37)通过调整超参数 ϵ 和 ρ ，用于控制信道估计误差和可变层模型驱动网络针对每个信道样本的层数。

3.5.3 基于DDPG的可变层黑盒网络拓展

基于DDPG的可变层模型驱动网络可以被拓展为基于DDPG的可变层黑盒网络。接下来，以具有全连接结构的黑盒网络作为例子进行阐述，其他结构比如具有卷积层和残差层的可变层黑盒网络，设计方法类似。全连接层可以表示为

$$\mathbf{y}^{l+1} = \varphi(\mathbf{W}^l \mathbf{y}^l + \mathbf{b}^l), \quad (3.40)$$

其中 \mathbf{y}^l 表示第 l 层的输出， φ 表示非线性函数，比如Sigmoid函数， \mathbf{W}^l 和 \mathbf{b}^l 分别表示第 l 层网络的训练权重参数和偏置参数。其MDP建模如下，其中奖励函数的设计同章节3.5.1。

- 状态空间：第 t 时刻的状态设计为 $s_t \triangleq \mathbf{y}^t$ ，即第 $(t-1)$ 层黑盒网络的输出。

- 动作空间：第 t 时刻的动作设计为 $a_t \triangleq [\tau_t, \mathbf{W}^t, \mathbf{b}^t]$ 。其中 $\tau_t \in [0, 1]$ 的作用是决定在当前层是否停止黑盒网络的运行，将当前层的结果作为最终结果输出。如果 $\tau_t > \varepsilon$ ，则基于DDPG的可变层黑盒网络继续运行下一层。
- 状态转移：根据观测到的状态 $s_t = \mathbf{y}^t$ 和选择的动作 $a_t = [\tau_t, \mathbf{W}^t, \mathbf{b}^t]$ ，当 $\tau_t > \varepsilon$ 时，状态转移由(3.40)所表示的一层或多层黑盒网络组成。以每次状态转移包含一层黑盒网络为例，状态转移过程被定义为 $\mathbf{y}^t \rightarrow \mathbf{y}^{t+1}$ 。

对于信道估计问题，采用具有全连接结构(3.40)的黑盒网络学习变量 α 、 γ 和 β 。因此，DDPG的状态定义为 $s_t = [\alpha, \gamma, \beta]$ 。基于DDPG的可变层黑盒网络输出 $[\alpha, \gamma, \beta]$ ，估计的信道可以根据公式(3.18)获得。此外，可以采用停止分数 L_t 作为 τ_t ，MDP可以参照章节3.5.2.2进行相应修改。

3.6 仿真结果

本节通过数值仿真来验证基于DDPG的可变层模型驱动网络的有效性。

3.6.1 仿真设置

考虑如下场景：基站配备包含 $N = 128$ 根天线的线性阵列，发送导频符号的长度为 $T = 60$ ，信噪比为 $\text{SNR} = 20\text{dB}$ 。采用3GPP空域信道模型来产生城市小区信道样本。不同的信道路径数 J 会导致信道的稀疏度不同，因此需要不同层数的网络（迭代次数不同的优化算法）进行求解。为提升样本的多样性，采用的训练数据集中的信道样本包含6到20条路径数。比较以下几种算法的仿真结果：

- DDPG Unfold/Adapt：提出的基于SBL算法和DDPG的可变层模型驱动网络；
- Unfold/Fix：基于SBL算法的模型驱动网络，具有离网基底和固定层数；
- DDPG Black-box/Adapt：基于DDPG的可变层黑盒网络；
- Black-box/Fix：固定层数的黑盒网络^[167]；
- SBL Off-grid：具有离网基底的SBL算法^[164]；
- Standard SBL：传统的标准SBL算法，采用公式(3.7)中的标准字典 \mathbf{A} ^[163]；
- Two-stage CS：传统的两阶段压缩感知算法^[159]。

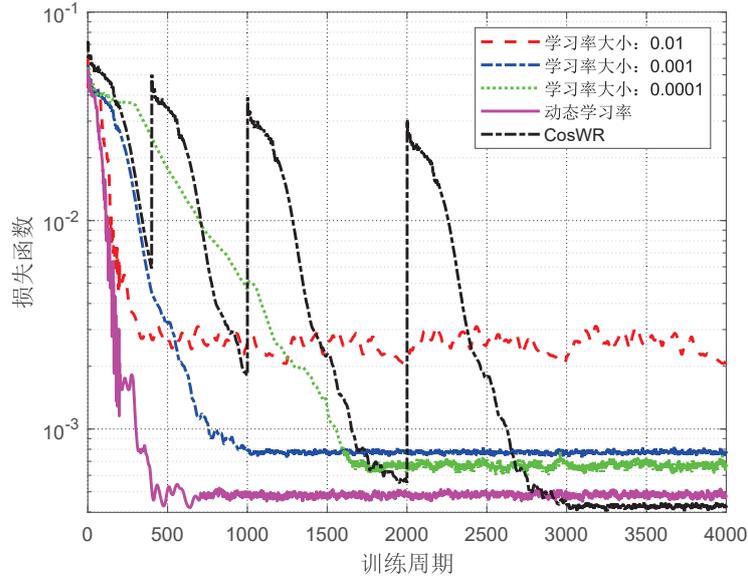


图 3.6 不同学习率下网络的收敛性

在仿真中，所有的迭代优化算法，即SBL Off-grid、Standard SBL和Two-stage CS，均执行到收敛。为Unfold/Fix和Black-box/Fix选择了最优的网络层数 L_{opt} ，即 L_{opt} 层网络能取得较好的性能，且在 L_{opt} 层之后再继续增加层数，性能不会继续提升。为了比较的公平性，通过调整(3.39)中的超参数 ϵ 来确保DDPG Unfold/Adapt的平均层数和Unfold/Fix相同。

3.6.2 NMSE性能

图3.6展示了不同学习率下网络的NMSE收敛性。由图可见，使用更小的学习率可以得到更好的性能，但收敛速度会减慢。另外，采用动态学习率，即学习率逐渐从 10^{-2} 下降到 10^{-4} ，可以实现良好的性能且收敛速度也较快。余弦退火和热重启学习率^[263]，即CosWR，实现了最佳性能，但与动态学习率相比，收敛速度相对较慢。另外，由于采用了热重启学习率，会导致收敛曲线的周期性振荡。

图3.7描述了基于DDPG的可变层模型驱动网络和各种基线算法在不同SNR下的NMSE性能。可以看出，所有算法实现的NMSE都会随SNR增加而降低。所提出的DDPG Unfold/Adapt优于其他基线算法，且性能由好到差依次为Unfold/Fix、DDPG Black-box/Adapt、Black-box/Fix、Standard SBL和Two-stage CS，且性能差距随SNR的增加而扩大。高效的原始迭代算法SBL Off-grid实现了最佳性能，主要因为引入离网基底并考虑了离网参数的优化，能保证找到原问题的局部最优解。但其需要的迭代次数较多，计算复杂度较高。此外，本章提出的DDPG Unfold/Adapt实现了接近SBL Off-grid的性能，且需要的层数更少，计算复杂度更低。因此，提出的基于DDPG的可变层模型驱动网络是求解信道估计问题的

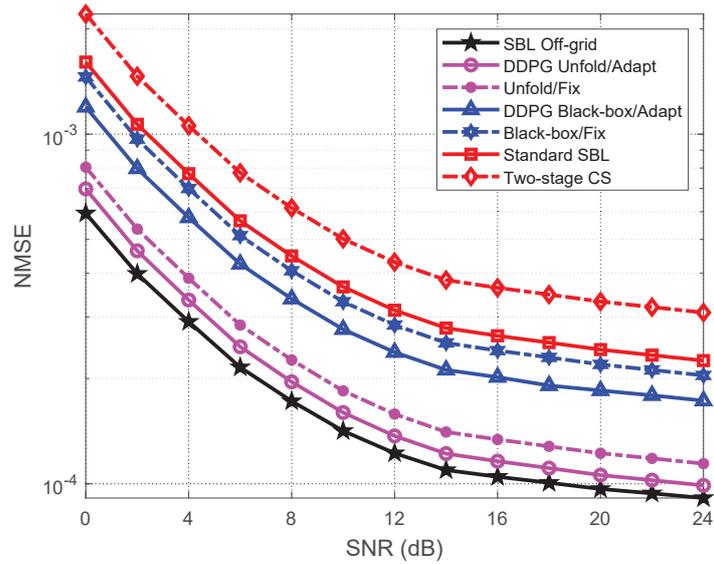


图 3.7 NMSE性能随SNR的变化

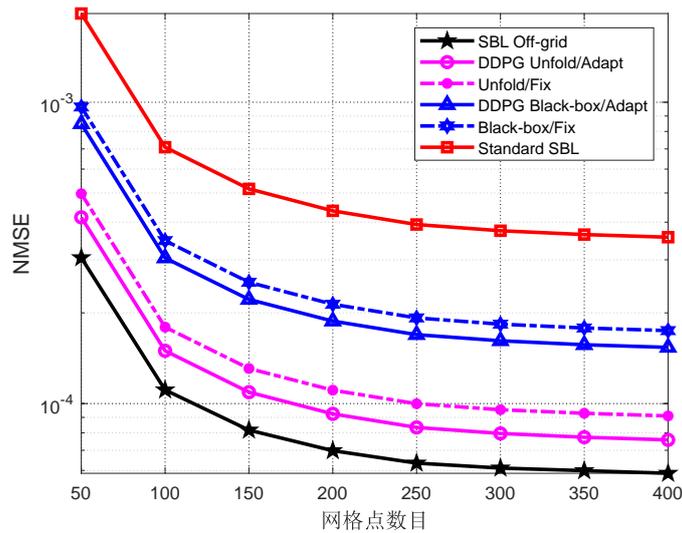


图 3.8 NMSE性能随网格点数目 \hat{J} 的变化

有效框架。

图3.8展示了NMSE性能随网格点数目 \hat{J} 的变化。由于更为密集的网格点使得对AoD的估计更为精确，所有算法实现的NMSE都会随着 \hat{J} 的增加而降低。此外，提出的DDPG Unfold/Adapt实现的性能逼近SBL Off-grid，且优于其他基线算法。特别地，因为随着 \hat{J} 增加，网格会变密，SBL Off-grid和DDPG Unfold/Adapt可以获得离网参数和AoD更精确的估计，因此和其他算法的性能差距会随 \hat{J} 增加而扩大。此外，可以看到，SBL Off-grid实现了比Standard SBL更好的性能，这主要是因为：（1）Standard SBL得到的解并不完全稀疏，并且由于没有离网基底和离网参数导致的方向失配和能量泄漏，使得其有明显的性能损

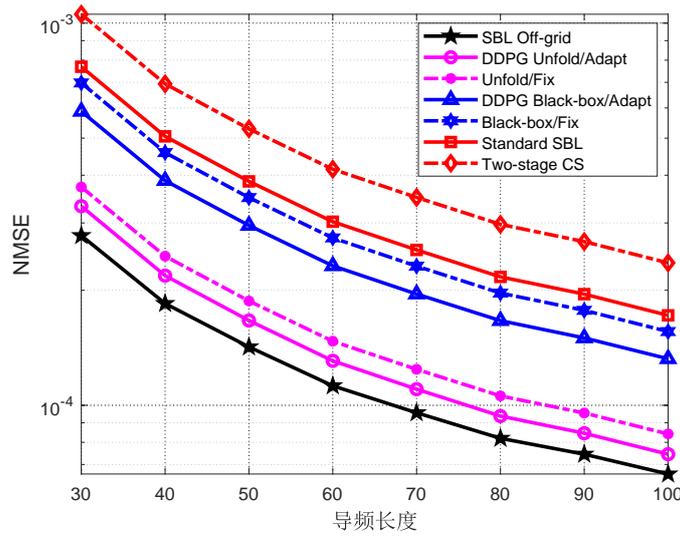


图 3.9 NMSE性能随导频长度 T 的变化

失；(2) SBL Off-grid算法显著提高了信道表征的稀疏性和准确性，并且可以很大程度上消除方向失配。

图3.9描述了NMSE性能随导频长度 T 的变化。可以看出，所有算法实现的NMSE随着 T 的增加而降低。由于Standard SBL和Two-stage CS忽略了离网参数，所以信道估计最不准确，性能最差。此外，DDPG Black-box/Adapt和Black-box/Fix都实现了更好的NMSE性能，但性能增益并不显著。尽管这两种黑盒方案考虑了离网参数，但它们并未采用迭代算法的结构。相比之下，本章所提出的DDPG Unfold/Adapt展开了带有离网参数的SBL算法，并实现了可变层模型驱动网络。它实现的性能超越了具有相同导频长度下的基线算法，并能够逼近SBL Off-grid的性能。

3.6.3 网络层数

图3.10展示了不同信道路径数和SNR下模型驱动网络的最优平均层数。其中，最优平均层数是通过平均测试数据集中所有样本达到最优性能时的层数得到的。虚线表示具有固定层数的模型驱动网络的最优平均层数，而实线则表示在达到几乎相同的NMSE性能前提下，基于DDPG的可变层模型驱动网络的最优平均层数。可以看出，提出的基于DDPG的可变层模型驱动网络在实现和固定层模型驱动网络性能一致的前提下，需要的最优平均层数减少了约30%。此外，所需的最优平均层数随着信道路径数的增加而增加，因为较多的信道路径数通常意味着信道更高的稀疏度，需要的网络层数也随之增加。另外，随着SNR的增加，所需的最优平均层数则会减少。这主要是因为SNR越大，信道估计越准

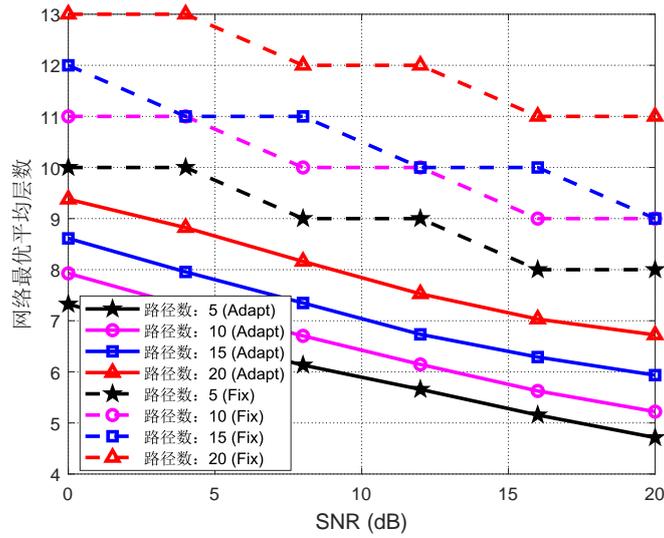


图 3.10 不同信道路径数和SNR下模型驱动网络的最优平均层数

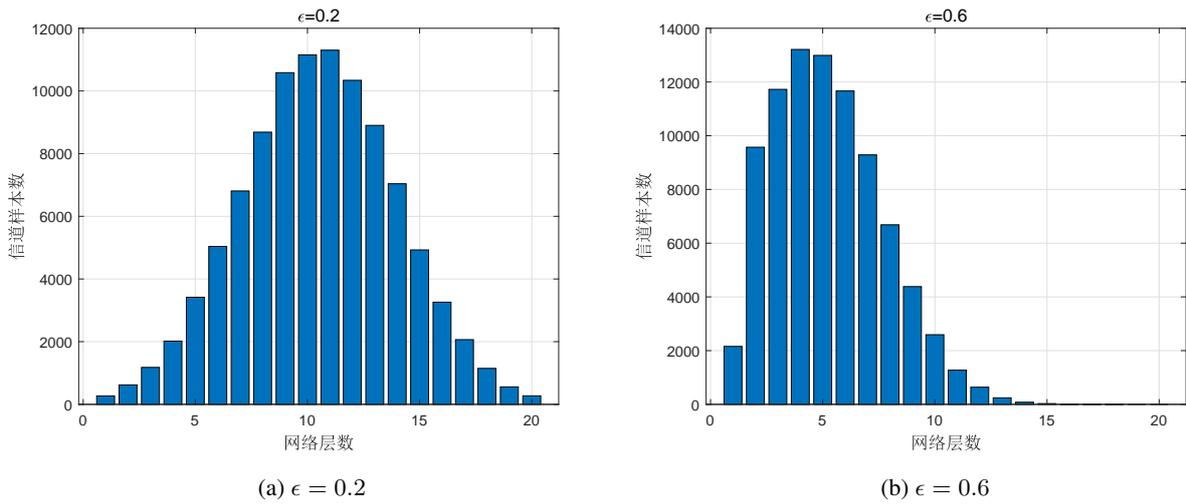


图 3.11 不同停止常数 ϵ 下的网络层数分布

确，需要的层数越少。

图3.11描述了公式(3.39)中定义的不同停止分数 ϵ 下，测试数据集中不同信道样本达到性能最优值，所需要网络层数的分布。这验证了不同稀疏度的信道样本所需要网络层数不同，因此可变层网络架构对于求解此类问题有实际的意义。仿真结果表明，越小的 ϵ 对应需要更多的网络层数。这是因为越小的 ϵ 对信道估计精度的要求越高，因此需要更多的网络层数来达到该精度。

图3.12展示了NMSE性能随网络层数 L 的变化，其中 L 表示DDPG Unfold/Adapt和DDPG Black-box/Adapt在测试数据集上达到最优性能时所有测试样本的平均层数。运行至

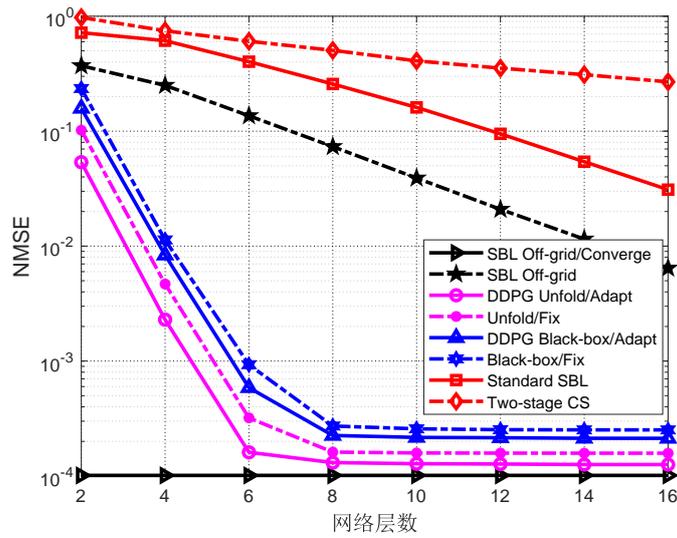


图 3.12 NMSE性能随网络层数 L 的变化

收敛的SBL Off-grid/Converge可以作为性能下界。由图可知，所有方案的NMSE都随着层数增加而降低，达到一定层数之后，性能变化趋于平缓达到饱和。对于深度学习算法，即DDPG Unfold/Adapt、Unfold/Fixed、DDPG Black-box/Adapt和Black-box/Fix，其NMSE在 L 较小时快速下降，而在 $L \geq 10$ 之后不再下降趋于平缓，因此 $L = 10$ 为最优平均层数。相比之下，迭代优化算法实现的NMSE性能，即Standard SBL、Two-stage CS和SBL Off-grid，其NMSE随 L 增加而缓慢下降，因为这些迭代优化算法需要大量的迭代次数才能收敛到一个较好的结果。因此，当层数 L 较小时，深度学习算法显著优于迭代优化算法。另外，本章提出的DDPG Unfold/Adapt超越其他深度学习方法，并且能够逼近迭代优化算法SBL Off-grid/Converge的性能下界，且需要的层数大大减少。算法DDPG Unfold/Adapt和Unfold/Fix的最优层数分别为 $L = 6$ 和 $L = 8$ 。因此，提出的DDPG Unfold/Adapt能够以比Unfold/Fix少约25%的层数实现与其相同的性能。由此可以得出结论：本章提出的基于DDPG的可变层模型驱动网络架构能够在计算复杂度和性能之间实现很好的平衡。

图3.13描述了不同网络架构下，NMSE性能随网络层数的变化，其中比较了四种不同的网络架构：（1）一个具有固定层数 L 的模型驱动网络（One Unfolding NN）；（2）两个具有不同固定层数的模型驱动网络（Two Unfolding NNs）；（3）基于DDPG的可变层模型驱动网络（DDPG Unfold/Adapt）；（4）基于DDPG的可变层黑盒网络（DDPG Black-box/Adapt）。同时保证每种方案在一个测试数据集上的平均层数相同。通常，信道路径数 J 的不同会导致信道的稀疏度不同，而具有更高稀疏度的信道样本往往需要更多的层数来实现较好的NMSE性能。因此，第二种方案采用一个固定层数为 $L + 1$ 层的模型驱动网络

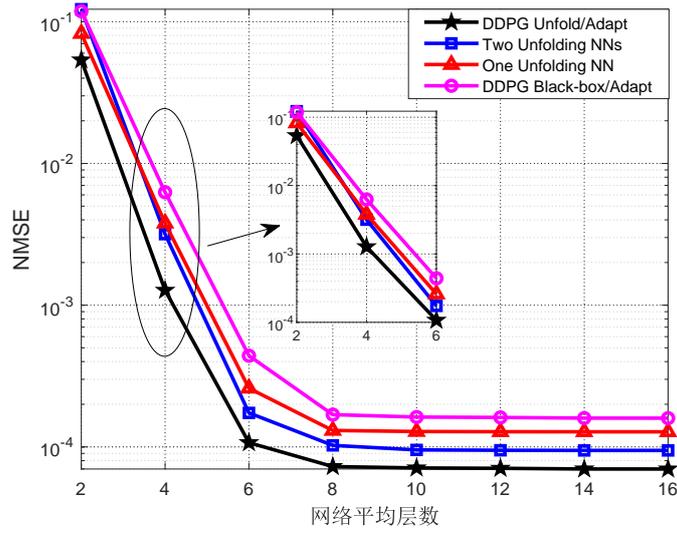


图 3.13 NMSE性能随网络层数的变化：不同网络架构的比较

来求解信道路径数为 $13 \leq J \leq 20$ 的场景，并采用另一个固定层数为 $L - 1$ 层的模型驱动网络来求解信道路径数为 $6 \leq J \leq 13$ 的场景。由仿真结果可见，当层数较小 $L < 4$ 时，方案二的性能比方案一的性能差。而当 $L \geq 4$ 时，方案二的性能超越了方案一的性能。此外，方案二在 $L = 6$ 时的性能可以逼近方案一在 $L = 8$ 时的性能。这表明了针对不同样本采用具有动态层数网络架构的有效性。另外，所有方案实现的 NMSE 随 L 的增加而先降低，后趋于平缓，且本章所提出的 DDPG Unfold/Adapt（方案三）实现的 NMSE 性能显著优于其他方案。

3.6.4 泛化能力

图3.14(a)展示了网络在网格点数 \hat{J} 和导频长度 T 不同的场景下的泛化能力。在 $\hat{J} = 400$, $T = 100$, $\text{SNR} = 20\text{dB}$, 以及 $N = 128$ 的系统参数下训练 DDPG Unfold/Adapt 和 DDPG Black-box/Adapt。图中“无系统参数失配”指训练数据集和测试数据集的系统参数相同，而“有系统参数失配”则指训练数据集和测试数据集的系统参数不同，测试训练集采用更小的 \hat{J} 和 T 。可以在较大系统参数下训练一个 DDPG Unfold/Adapt 或者 DDPG Black-box/Adapt，并通过填零的方式将其直接应用于一个具有较小系统参数的场景。具体而言，采用导频 $\mathbf{X}_1 \in \mathbb{C}^{T_1 \times N}$ 和字典 $\mathbf{A}_1(\boldsymbol{\beta}) = [\mathbf{a}_1(\hat{\phi}_1 + \beta_1), \mathbf{a}_1(\hat{\phi}_2 + \beta_2), \dots, \mathbf{a}_1(\hat{\phi}_{\hat{J}_1} + \beta_{\hat{J}_1})] \in \mathbb{C}^{N \times \hat{J}_1}$ 来训练一个 DDPG Unfold/Adapt，并可将其直接用于一个具有更小 \hat{J}_2 和 T_2 的系统中。对 $\mathbf{X}_2 \in \mathbb{C}^{T_2 \times N}$ 和 $\mathbf{A}_2(\boldsymbol{\beta}) \in \mathbb{C}^{N \times \hat{J}_2}$ 进行填零操作，填充后的结果可以分别表示为 $[\mathbf{X}_2; \mathbf{0}^{(T_1 - T_2) \times N}]$ 和 $[\mathbf{a}_2(\hat{\phi}_1 + \beta_1), \mathbf{a}_2(\hat{\phi}_2 + \beta_2), \dots, \mathbf{a}_2(\hat{\phi}_{\hat{J}_2} + \beta_{\hat{J}_2}), \mathbf{0}^{N \times (\hat{J}_1 - \hat{J}_2)}]$ ，从而能够具有和 \mathbf{X}_1 、 $\mathbf{A}_1(\boldsymbol{\beta})$ 相同的维度。由图可知，尽管训练数据和测试数据存在系统参数 \hat{J} 和 T 不匹

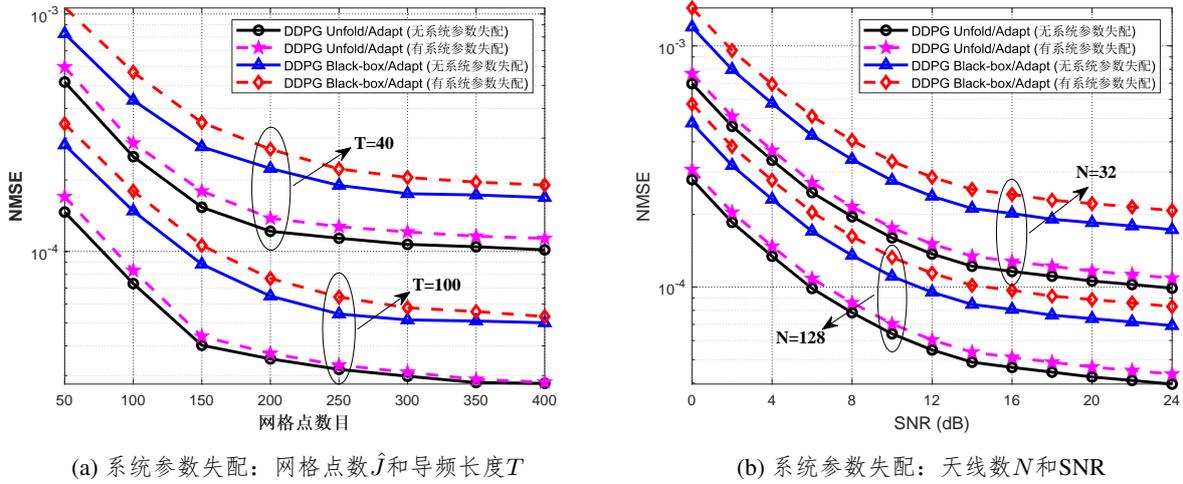


图 3.14 网络在不同系统参数失配场景下的泛化能力

配，但由此导致的网络性能损失很小。这一现象证明了该方案具有良好的泛化能力，即在不同的系统参数 \hat{J} 和 T 下都能实现较好的性能。此外，由系统参数失配导致的性能损失随着 \hat{J} 和 L 增加而降低。这主要是因为当训练数据集和测试数据集的系统参数失配程度减小，其导致的性能损失也会减小。

图3.14(b)展示了网络在天线数 N 和 SNR 不同的场景下的泛化能力。为了提升网络对 SNR 的泛化能力，采用如下系统配置下的数据集来训练网络： $\hat{J} = 400$ ， $T = 100$ ， $\text{SNR} = 0, 2, \dots, 24 \text{ dB}$ ，以及 $N = 128$ 。由图可知，由于训练数据集和测试数据集的系统参数 N 和 SNR 的失配，带来的 DDPG Unfold/Adapt 的性能损失很小。此外，由于系统参数失配导致的 DDPG Unfold/Adapt 的性能损失小于 DDPG Black-box/Adapt。这证明了本章所提出的可变层模型驱动网络架构 DDPG Unfold/Adapt 相比黑盒网络架构 DDPG Black-box/Adapt 具有更好的泛化能力。

3.7 本章小结

本章提出了一种基于 DDPG 的可变层模型驱动网络架构，其针对不同的输入样本实现了自适应调整所需的网络层数，它的训练参数通过 DDPG 学习得到。该框架可被用于解决大规模 MIMO 系统中的信道估计问题。具体而言，本章建模了基于离网基底的大规模 MIMO 系统中的信道估计问题，并提出了一种基于 SBL 的迭代优化算法求解该问题。随后，将该算法展开为类似于 DNN 的逐层结构，引入一组训练参数降低其复杂度，同时提升性能，并给出了该模型驱动网络的性能分析。在此网络结构的基础上，结合基于 DDPG 的

可变层模型驱动网络框架更高效地解决了该信道估计问题。为了实现模型驱动网络的可变层结构，设计了停止分数来指示网络何时停止并输出结果。而该停止分数被设计为信道估计误差的函数，用来刻画信道估计精度。此外，本章对该提出的框架进行了扩展，可被用于实现具有全连接层、卷积层结构等普适性结构的DNN的可变层架构。仿真结果表明，提出的基于DDPG的可变层模型驱动网络架构在层数大大减少的情况下，可以实现优于传统迭代优化算法的性能。

4 基于联合强化学习和模型驱动网络的波束选择和预编码设计

4.1 引言

前两章提出了针对无线通信系统中物理层收发机的模型驱动网络框架，并设计了可变量模型驱动网络，以较低的计算复杂度并实现了较好的性能。但对于更为复杂的问题，比如带有离散变量、连续变量和耦合约束的MINLP问题或者NP难问题，一般的深度学习方法难以求解，需要设计更为精巧的网络结构和训练方式才能高效求解并实现满意的性能。这主要是因为具有离散变量的问题求解剪复杂度，且离散变量的存在使得一般的黑盒网络和模型驱动网络在训练过程中，梯度难以直接通过反向传播算法逐层传递，加上耦合约束难以满足，从而为网络设计和训练带来了极大的挑战。

为解决物理层收发机中存在的更为复杂的MINLP问题和NP难问题，本章提出了一种新颖的基于联合强化学习和模型驱动网络的框架，分别用于求解问题中的离散变量和连续变量，并通过网络结构的设计使得约束满足。同时，提出了一个系统中多个子网络的联合训练方式，使得物理层收发机中处理不同变量和模块的网络可以端到端联合训练，从而实现更优的系统性能。具体而言，在物理层收发机中，配置DLA的大规模MU-MIMO系统中的联合波束选择和预编码设计问题是一个具有离散变量、连续变量和耦合约束的MINLP问题，难以通过低复杂度算法找到性能令人满意的解。本章提出了一个联合优化波束选择和数字预编码矩阵的框架。首先，将波束选择问题建模为MDP，并设计DDQN来求解该问题。目标是通过选择最优的波束，使被服务用户的和速率最大化，同时确保波束选择矩阵相关的约束条件得到满足。基站被设计为一个智能体，并设计了相应的状态、动作和奖励函数。此外，充分利用波束空间信道的稀疏性来降低波束空间的维度以加快网络收敛，并将波束的选择建模为DRL网络的一个动作。

在给定波束选择矩阵的情况下，数字预编码设计问题是一个经典的带有功率约束的和速率最大化问题，可使用迭代WMMSE算法^[86]求解。该算法可以实现较好的和速率性能，但由于存在高维矩阵求逆操作以及需要较大的迭代次数，其计算复杂度较高。为了解决这些问题，设计一种模型驱动网络，将迭代WMMSE算法被展开为逐层的网络结构。该网络

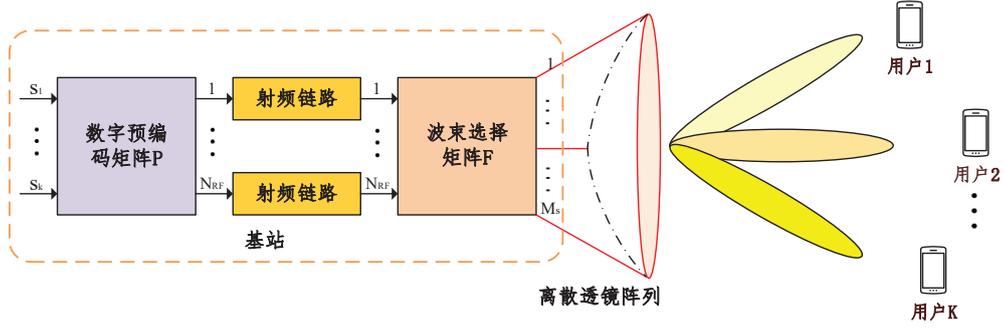


图 4.1 带有DLA的毫米波MU-MIMO系统

可以使用更少的层数来逼近迭代WMMSE算法的性能，并且避免了高维矩阵求逆操作，从而降低了计算复杂度。此外，引入训练参数，以提高网络实现的和速率性能。为了高效地联合训练上述强化学习网络和模型驱动网络，提出了一种联合训练方法，这与现有的网络联合设计方案^[264]不同。另外，分析了所提出的联合网络架构的计算复杂度，并与一些基线算法进行对比。

本章内容安排如下。4.2节介绍了带有DLA的MU-MIMO系统模型，并对联合波束选择和预编码问题进行了数学建模。4.3节描述了联合网络设计框架。用于设计波束选择矩阵的DRL网络在4.4节中介绍。4.5节针对数字预编码矩阵设计了模型驱动网络，并分析了其计算复杂度。4.6节给出了仿真实验的结果。最后，4.7节对本章进行了总结。

4.2 系统模型和问题建模

本节介绍了联合波束选择和数字预编码的系统模型，并对该问题进行了数学建模。

4.2.1 系统模型

如图4.1所示，本章研究下行毫米波MU-MIMO系统，它由一个装配有DLA的基站组成。基站装配有 M_s 根发送天线和 $N_{RF} < M_s$ 条射频链路。基站同时服务 K 个用户，每一个用户装配单天线。为了确保 K 个用户的空间复用增益，需要满足如下条件： $N_{RF} \geq K$ ¹。在基站端经过预编码的发送数据矢量表示为

$$\mathbf{x} = \mathbf{P}\mathbf{s} = \sum_{k=1}^K \mathbf{p}_k s_k, \quad (4.1)$$

其中 $\mathbf{s} = [s_1, s_2, \dots, s_K]^T$, s_k 表示传输给用户 $k \in \mathcal{K} \triangleq \{1, 2, \dots, K\}$ 的信号，其均值为零，协方差矩阵为 $\mathbb{E}[\mathbf{s}\mathbf{s}^H] = \mathbf{I}$ ，且 $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K] \in \mathbb{C}^{N_{RF} \times K}$ 表示预编码矩阵， \mathbf{p}_k 为针对用

¹现已有符号级预编码技术可在 N_{RF} 略小于 K 的情况下实现全用户复用，本章为描述方便起见假定 $N_{RF} \geq K$ 。

户 k 的数字预编码矢量。用户接收到的信号矢量 $\mathbf{y} \in \mathbb{C}^{K \times 1}$ 可以表示为

$$\mathbf{y} = \mathbf{H}^H \mathbf{F} \mathbf{P} \mathbf{s} + \mathbf{n}, \quad (4.2)$$

其中 $\mathbf{H} \in \mathbb{C}^{M_s \times K}$ 表示波束空间信道矩阵， $\mathbf{F} \in \mathbb{C}^{M_s \times N_{RF}}$ 为波束选择矩阵，其元素 $f_{ij}, (i, j) \in \mathcal{T}$ 为0或1， $\mathcal{T} \triangleq \{(i, j) | i = 1, 2, \dots, M_s, j = 1, 2, \dots, N_{RF}\}$ 。此外， $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_K)$ 为 $K \times 1$ 维的AWGN矢量，其中 σ^2 表示噪声方差。

4.2.2 波束空间信道模型

波束空间信道矩阵 \mathbf{H} 可以通过物理空间MIMO信道进行建模

$$\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K] = [\mathbf{U} \mathbf{g}_1, \mathbf{U} \mathbf{g}_2, \dots, \mathbf{U} \mathbf{g}_K], \quad (4.3)$$

其中 $\mathbf{U} \in \mathbb{C}^{M_s \times M_s}$ 是离散傅里叶变换(DFT)矩阵， $\mathbf{g}_k \in \mathbb{C}^{M_s \times 1}$ 为基站和用户 k 之间的空间域信道矢量。DFT矩阵 \mathbf{U} 包含 M_s 个正交的波束^[124]

$$\mathbf{U} = [\mathbf{a}(\varphi_1), \mathbf{a}(\varphi_2), \dots, \mathbf{a}(\varphi_{M_s})]^H, \quad (4.4)$$

其中 $\varphi_m = \frac{1}{M_s} (m - \frac{M_s + 1}{2})$, $m = 1, 2, \dots, M_s$ 为经过标准化的空间方向， $\mathbf{a}(\varphi_m) = \frac{1}{\sqrt{M_s}} [e^{-j2\pi\varphi_m i}]_{i \in \mathcal{I}}$ 表示相应的 $M_s \times 1$ 维阵列导向矢量， $\mathcal{I} \triangleq \{n - \frac{M_s - 1}{2} | n = 0, 1, \dots, M_s - 1\}$ 为阵元的标号。注意到 \mathbf{U} 矩阵的列之间是相互正交的，即 $\mathbf{U}^H \mathbf{U} = \mathbf{I}$ 。

本章采用著名的Saleh-Valenzuela信道模型^[129]

$$\mathbf{g}_k = \rho_k^{(0)} \mathbf{a}(\phi_k^{(0)}) + \sum_{l=1}^L \rho_k^{(l)} \mathbf{a}(\phi_k^{(l)}), \quad (4.5)$$

其中 $\rho_k^{(0)} \mathbf{a}(\phi_k^{(0)})$ 和 $\rho_k^{(l)} \mathbf{a}(\phi_k^{(l)})$ 分别表示基站和用户 k 之间的直射径 (Line-of-Sight, LoS)，以及第 l 条非直射径 (Non Line-of-Sight, NLoS) 对应的信道矢量。此外， $\rho_k^{(0)}$ 和 $\rho_k^{(l)}$ 分别表示LoS和NLoS信道路径的增益， $\phi_k^{(0)}$ 和 $\phi_k^{(l)}$ 表示空间方向。为方便起见，考虑二维信道模型，即只有AoD的情况。在(4.5)中，NLoS信道路径的数目 L 远小于发送天线数 M_s ，因为环境中散射体的数目是有限的，因此信道 \mathbf{H} 具有稀疏性。在信道模型中， \mathbf{H} 包含 M_s 个波束， \mathbf{H} 的每一行代表一个波束矢量。

注 4.1. 用户多天线 (每个用户装配 N_r 根天线) 和用户单天线场景的区别: (1) 多天线用户接收到的信号为 N_r 维的矢量，而不是标量; (2) 多天线用户需要设计接收端的数字编码矩阵; (3) 多天线用户的信道 \mathbf{H} 有AoA和AoD且维度为 $M_s \times KN_r$ 。但这并不会影响基站端

的强化学习网络和模型驱动网络的设计。此外，多天线用户接收端的数字编码矩阵可以通过迭代WMMSE算法得到。因此，可以通过设计基于多用户MIMO信道的模型驱动网络，将提出的联合网络设计框架拓展到用户多天线的场景。

4.2.3 问题建模

关注波束选择矩阵 \mathbf{F} 和数字预编码矩阵 \mathbf{P} 的联合设计问题，使得系统下行传输的和速率最大化。系统和速率最大化的问题可以数学建模为

$$\max_{\{\mathbf{F}, \mathbf{P}\}} \sum_{k=1}^K \log \left(1 + \frac{|\mathbf{h}_k^H \mathbf{F} \mathbf{p}_k|^2}{\sum_{i \neq k} |\mathbf{h}_k^H \mathbf{F} \mathbf{p}_i|^2 + \sigma^2} \right) \quad (4.6a)$$

$$\text{s.t.} \quad \text{Tr}(\mathbf{P}^H \mathbf{F}^T \mathbf{F} \mathbf{P}) \leq P_s, \quad (4.6b)$$

$$\sum_{i=1}^{M_s} f_{ij} = 1, \forall j, \quad (4.6c)$$

$$\sum_{j=1}^{N_{RF}} f_{ij} \leq 1, \forall i, \quad (4.6d)$$

$$f_{ij} \in \{0, 1\}, \forall (i, j) \in \mathcal{T}, \quad (4.6e)$$

其中 P_s 为基站的传输功率。注意到约束(4.6b)可以被化简为 $\text{Tr}(\mathbf{P}^H \mathbf{P}) \leq P_s$ ，因为 $\mathbf{U}^H \mathbf{U} = \mathbf{I}_{M_s}$ ，以及 $\mathbf{F}^T \mathbf{F} = \mathbf{I}_{N_{RF}}$ 。约束(4.6c)确保每一条射频链路对应一个波束，约束(4.6d)保证每个波束至多对应一条射频链路。这些约束保证了可以从 M_s 个波束中选择 N_{RF} 个波束来服务 K 个用户。问题(4.6)为MINLP，是非凸的且包含离散变量 \mathbf{F} ，因此很难求解。因此，提出高效的网络联合设计框架对其进行求解。

4.3 网络的联合设计框架

本节提出网络的联合设计框架来求解通信系统中极具挑战的MINLP问题。具体而言，DRL网络和模型驱动网络可分别被用于优化离散和连续的变量。相应地，可用该框架求解联合波束选择和数字预编码设计的问题。如图4.2所示，提出的框架包括两部分：联合网络设计和信号流程图，下面进行详细阐述。

4.3.1 网络联合设计

为了求解MINLP问题(4.6)，提出网络的联合设计框架，其中包括基于DRL网络的波束选择矩阵 \mathbf{F} 的设计和基于模型驱动网络的数字预编码矩阵 \mathbf{P} 的设计。首先，信道矩阵 \mathbf{H} 被

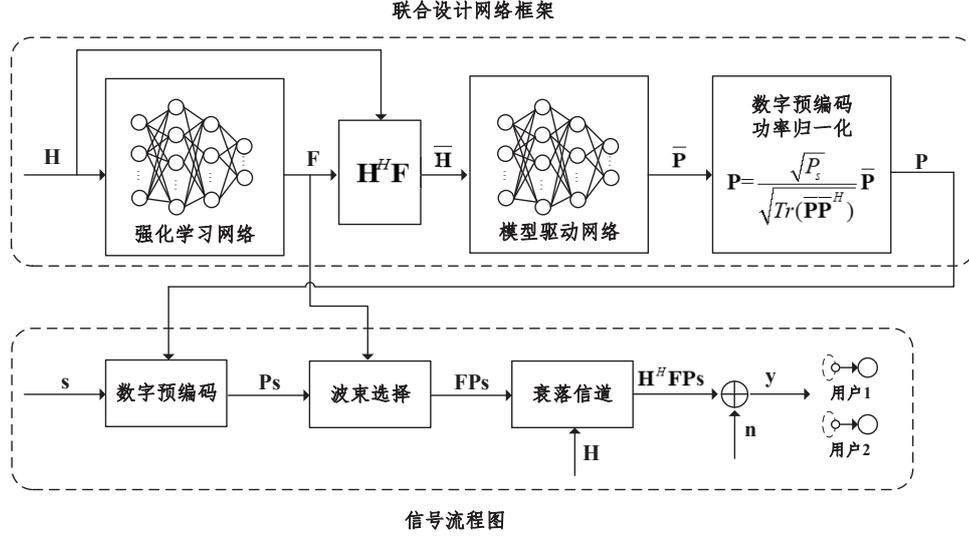


图 4.2 联合波束选择和数字预编码设计的网络框架

转换成维度为 $2 \times M_s \times K$ 的实值矩阵，实部和虚部分开存储。接着，将信道矩阵 \mathbf{H} 输入到具有 DDQN 结构的 DRL 网络。该网络的目标是选择最优的波束并最大化被服务用户的和速率，同时使得约束(4.6c)-(4.6e)满足。DRL 网络输出相应的波束选择矩阵 \mathbf{F} ，由此得到等效信道矩阵 $\bar{\mathbf{H}}^H = \mathbf{H}^H \mathbf{F} \in \mathbb{C}^{K \times N_{RF}}$ ，它的维度比原始信道矩阵 \mathbf{H} 更低。

给定等效信道矩阵 $\bar{\mathbf{H}}$ ，通过设计基于迭代 WMMSE 算法的模型驱动网络，求解关于数字预编码矩阵 $\bar{\mathbf{P}}$ 的子问题。具体而言，迭代 WMMSE 算法被展开成类似于 DNN 的逐层结构，同时引入训练参数。模型驱动网络的输入是 $\bar{\mathbf{H}}$ ，输出是未经过归一化的数字预编码矩阵 $\bar{\mathbf{P}}$ 。根据待求解问题的结构，可以看到最优解为使得功率约束(4.6b)取等号的解。为了使得约束(4.6b)满足，使用如下归一化层将 $\bar{\mathbf{P}}$ 映射为 \mathbf{P}

$$\mathbf{P} = \frac{\sqrt{P_s}}{\sqrt{\text{Tr}(\bar{\mathbf{P}}\bar{\mathbf{P}}^H)}} \bar{\mathbf{P}}. \quad (4.7)$$

最后， \mathbf{H} ， \mathbf{F} 和 \mathbf{P} 带入损失函数

$$\mathcal{L}_1(\boldsymbol{\theta}) = -\frac{1}{N_s} \sum_{n=1}^{N_s} f(\mathbf{H}_n, \mathbf{F}_n, \mathbf{P}_n; \boldsymbol{\theta}), \quad (4.8)$$

其中 N_s 表示训练数据集的大小， $\boldsymbol{\theta}$ 表示网络的训练参数， $f(\mathbf{H}_n, \mathbf{F}_n, \mathbf{P}_n; \boldsymbol{\theta})$ 表示和速率(4.6a)是通过输入第 n 个信道样本计算得到的。在训练阶段，输入 \mathbf{H} ，DRL 网络和模型驱动网络分别产生波束选择矩阵 \mathbf{F} 和数字预编码矩阵 \mathbf{P} 。随后，使用梯度反向传播算法联合训练两个网络，并使用 SGD 更新训练参数。此外，将模型驱动网络的输出带入目标函数(4.8)，该计算得到的值为 DRL 网络奖励函数的一部分，可以帮助更高效地训练 DRL 网络。

4.3.2 信号流程图

在推理阶段，信号流程图模拟了从发送数据符号 \mathbf{s} 到接收符号 \mathbf{y} 的整个过程。其中，发送信号经历了数字预编码 \mathbf{P} 、波束选择 \mathbf{F} 、无线衰落信道 \mathbf{H} 和高斯白噪声 \mathbf{n} 。信道 \mathbf{H} 输入网络并输出波束选择矩阵 \mathbf{F} 和 \mathbf{P} 。之后数字预编码模块和波束选择模块被网络输出的矩阵 \mathbf{P} 和 \mathbf{F} 分别替代。

4.4 基于深度强化学习网络的波束选择

本节先介绍MDP和DRL的基本概念，再介绍基于DRL网络的波束选择设计。

4.4.1 马尔科夫决策过程和价值函数

4.4.1.1 马尔科夫决策过程

马尔科夫决策过程由五元组 $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma)$ 组成，其中 \mathcal{S} 表示状态空间， \mathcal{A} 表示动作空间， $\mathcal{R} : \mathcal{S} \rightarrow \mathbb{R}$ 为奖励函数， $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ 代表状态转移概率， $\gamma \in (0, 1)$ 表示折现因子。此外， s_t, a_t, r_t 分别表示 t 时刻的状态、动作和奖励函数。其中 $a_t \sim \pi(\cdot|s_t)$ ， $s_{t+1} \sim p(\cdot|s_t, a_t)$ ， $r_t \triangleq r(s_t, a_t) \sim \mathcal{R}$ ，累计折现奖励 $\sum_{t \geq 0} \gamma^t r_t$ ， p 为转移概率，以及 π 表示策略 $\pi : \mathcal{S} \mapsto \mathcal{A}$ 。

4.4.1.2 价值函数

状态 s 的价值函数 $V^\pi(s)$ 定义为根据策略 π 得到的累积奖励

$$V^\pi(s) = \mathbb{E} \left\{ \sum_{t \geq 0} \gamma^t r_t | s_0 = s, \pi \right\}, \quad (4.9)$$

其中期望 $\mathbb{E}\{\cdot\}$ 表示对一批样本求的平均值。类似地，状态-价值函数 $Q^\pi(s, a)$ 表示在状态 s 和策略 π 下选取动作 a 得到的奖励

$$Q^\pi(s, a) = \mathbb{E} \left\{ \sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi \right\}. \quad (4.10)$$

最优策略 π^* 通过贝尔曼方程计算获得

$$Q^{\pi^*}(s_t, a_t) = \max_{\pi} \mathbb{E} \left\{ r_{t+1} + \gamma \max_a Q^\pi(s_{t+1}, a) | s_t, a_t \right\}. \quad (4.11)$$

DRL用于估计动作-价值函数。DRL的目标为最小化损失函数

$$L(\boldsymbol{\theta}) = \mathbb{E} \left\{ \left(y_t - Q(s_{t+1}, a; \boldsymbol{\theta}) \right)^2 \right\}, \quad (4.12)$$

其中 $\boldsymbol{\theta}$ 表示网络的训练参数。本章采用DDQN结构来学习目标值 y_t ，避免对目标值过高的估计同时使得网络更稳定，数学表达为

$$y_t = \mathbb{E} \left\{ r_{t+1} + \gamma Q \left(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \boldsymbol{\theta}); \boldsymbol{\theta}' \right) \right\}, \quad (4.13)$$

其中 $\boldsymbol{\theta}'$ 和 $\boldsymbol{\theta}$ 分别表示两个网络的学习参数，分别用于选择和评估动作。梯度可以表示为

$$\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \mathbb{E} \left\{ r_{t+1} + \gamma Q \left(s_{t+1}, \arg \max_a Q(s_{t+1}, a; \boldsymbol{\theta}); \boldsymbol{\theta}' \right) - Q(s_{t+1}, a; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} Q(s_{t+1}, a; \boldsymbol{\theta}) \right\}. \quad (4.14)$$

关于DRL中策略梯度的详细介绍请见附录A.3.1。

4.4.2 基于深度强化学习的波束选择

首先，将波束选择的过程建模为MDP，其中基站被设为智能体。本章精细设计了状态、动作、状态转移和奖励函数。求解该问题的主要难点和创新点在于：（1）该波束选择问题是NP难问题，含有离散变量和耦合约束，其中约束(4.6c)-(4.6e)必须保证被满足。（2）目标是通过精心地设计奖励函数，选择最优的波束，并最大化用户的和速率(4.6a)。因此，需充分利用波束空间信道的性质，并考虑被选择波束的能量，用户公平性和SINR。（3）因为天线数 M_s 非常大，导致状态空间非常大，严重影响了网络的收敛性能。因此，需通过降低维度来加快网络的收敛速度，同时尽可能不影响系统的和速率性能。（4）DRL网络需要和模型驱动网络进行联合训练，这部分将会在下一章节进行详细阐述。

4.4.2.1 波束选择的马尔科夫决策过程建模

波束选择的MDP建模如下。

- 智能体: 基站观察到当前状态 s_t ，基于策略 π ，选择一个动作 a_t ，和环境进行交互。根据从环境得到的反馈（奖励函数），基站调整策略 π 。目标是寻找最优策略 π ，最大化累计折现奖励函数的期望 $\mathbb{E} \left\{ \sum_{t \geq 0} \gamma^t r_t \right\}$ 。
- 状态空间: 将波束空间信道矩阵 $\mathbf{H} \in \mathbb{C}^{M_s \times K}$ 转换成维度为 $2 \times M_s \times K$ 的实值矩阵，其中实部和虚部分开存储。状态空间 $\mathcal{S} \in \mathbb{C}^{3 \times M_s \times K}$ 包括两部分: (1)维度为 $2 \times M_s \times K$ 的信道矩阵 \mathbf{H} ; (2)维度为 $1 \times M_s \times K$ 的指示矩阵，表示该波束是否已被选择。指示矩阵中的所有元素都初始化为1，表示一开始所有波束都未被选择。

- 动作空间: 动作空间被设计为可选择的波束, 即 $\mathcal{A} = \{1, 2, \dots, M_s\}$, 在每个时刻 t 只有一个波束被选择。马尔科夫决策过程在一个周期有 N_{RF} 个时刻, 即 $t \in \{1, 2, \dots, N_{RF}\}$ 。通过该设计, 约束(4.6c)和(4.6e)可以被满足。
- 状态转移: 给定当前的状态 s_t 和选择的动作 $a_t = i$, 状态将会转移到 s_{t+1} 。状态中的指示矩阵第 i 行将被设成 $\mathbf{0}$, 信道矩阵 \mathbf{H} 保持不变。

4.4.2.2 奖励函数

首先需要确保约束(4.6d)满足, 即每个波束最多被选择一次。如果一个波束被重复选择, 智能体将会被给予惩罚 ρ 。波束被重复选择, 仅当指示矩阵为 $\mathbf{0}$ 的位置对应的波束又被选中。如果被选择的波束之前没被选中过, 奖励函数将由如下几部分构成, 不同的部分从不同方面对被选择的波束进行评估。

- 首先, 奖励函数设计为在每个时刻被选择波束矢量的 l_2 范数, 它表征了被选择波束的能量。由于波束空间信道矩阵 \mathbf{H} 有稀疏的结构, 这部分奖励函数可以避免选择能量较低的波束。
- 其次, 在最后一个时刻 $t = N_{RF}$, N_{RF} 个波束都已选完。由此, 可得到波束选择矩阵 \mathbf{F} 。由于旨在最大化用户和速率(4.6a), 在最后一个时刻 $t = N_{RF}$ 的另一部分奖励函数可以被设计为

$$\sum_{k=1}^K \log \left(1 + \frac{|\mathbf{h}_k^H \mathbf{F} \mathbf{p}_k|^2}{\sum_{i \neq k} |\mathbf{h}_k^H \mathbf{F} \mathbf{p}_i|^2 + \sigma^2} \right), \quad (4.15)$$

其中数字预编码 \mathbf{p}_k 是通过模型驱动网络计算得到的, 该部分将在下一章节介绍。

- 再者, 因为奖励函数(4.15)仅能用于最后一个时刻 $t = N_{RF}$, 在其他时刻 $t < N_{RF}$ 增加如下表达式作为奖励函数的一部分。这是 SINR 的近似^[132], 评估了不同用户之间的干扰情况。

$$\sum_{k=1}^K \frac{|h_{jk}^t|^2}{\sum_{i \neq k} |h_{ji}^t|^2 + \sigma^2}, \quad (4.16)$$

其中 j 为时刻 t 被选择波束的编号, h_{ji}^t 表示矩阵 \mathbf{H} 第 j 行第 i 列的元素。

- 此外, 为了避免没有波束对准某些用户, 这些用户的速率会非常小甚至接近 0。在时刻 $t \geq \frac{N_{RF}}{2}$, 将每个用户波束的平均能量作为奖励函数的一部分, 确保用户公平性。

$$\sum_{k=1}^K \frac{\|\tilde{\mathbf{h}}_k^t\|^2 - \|\tilde{\mathbf{h}}_k^{t-1}\|^2}{\|\tilde{\mathbf{h}}_k^t\|^2 + \varepsilon} (\text{sgn}(\delta - \|\tilde{\mathbf{h}}_k^{t-1}\|^2) + 1). \quad (4.17)$$

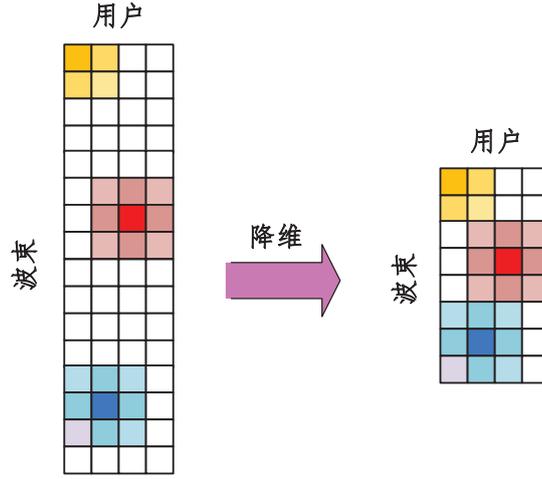


图 4.3 波束空间信道的降维过程，其中颜色越深表示波束具有越高的能量

其中 $\tilde{\mathbf{H}}^t \in \mathbb{C}^{t \times K}$ 表示由 t 个被选择波束组成的信道矩阵，它包含在 t 时刻之前被选择的所有波束， $\tilde{\mathbf{h}}_k^t \in \mathbb{C}^{t \times 1}$ 表示矩阵 $\tilde{\mathbf{H}}^t$ 的第 k 列， ε 往往取一个比较小的值以避免数值误差， δ 是一个给定的阈值。 $\|\tilde{\mathbf{h}}_k^t\|^2 - \|\tilde{\mathbf{h}}_k^{t-1}\|^2$ 这一项代表了在时刻 t 被选择的波束在用户 k 的能量。这里引入符号函数 $\text{sgn}(\cdot)$ ，(4.17) 为一个针对用户 k 的额外奖励，当没有被选择的波束对准用户 k ，用户 k 的波束能量很小，即 $\|\tilde{\mathbf{h}}_k^{t-1}\|^2 < \delta$ 。此时，为用户 k 选择一个能够对准的波束，可以极大提升其速率，且有助于用户公平性。

注意到上述四部分奖励函数，即波束能量和(4.15)-(4.17)，将被加权并求和写成一个公式，以提升DRL网络的性能。

4.4.2.3 提升网络性能的技巧

- 降维: 由于天线数 M_s 往往很大，波束空间信道矩阵 $\mathbf{H} \in \mathbb{C}^{M_s \times K}$ 的维度一般会很大。选择 $\bar{M}_s (\bar{M}_s \ll M_s)$ 个能量最大的波束来构建低维的波束空间信道矩阵，以起到降维的作用。相应地，状态和动作空间的维度分别被减小为 $3 \times \bar{M}_s \times K$ 和 \bar{M}_s 。当状态和动作空间维度较低时，DRL网络更稳定且具有更好的收敛性能。由于信道矩阵 \mathbf{H} 具有稀疏结构，通过选择合适的 \bar{M}_s ，适当的降维并不会导致系统性能的下降。此外，超参数 \bar{M}_s 需要被仔细地选择，因为如果 \bar{M}_s 过大会导致信道矩阵仍然具有较高的维度，起不到降维的作用，而如果 \bar{M}_s 过小，将会删去一些较好的波束从而影响系统性能。图4.3以 $K = N_{RF} = 4, M_s = 16, \bar{M}_s = 8$ 为例展示了降维的过程。具有较低能量的波束被剔除，波束空间信道矩阵最终降维到包含 $\bar{M}_s = 8$ 个波束。
- 带有优先级的经验回放池^[218]: 为了克服网络训练的不稳定性，并降低样本之间的相

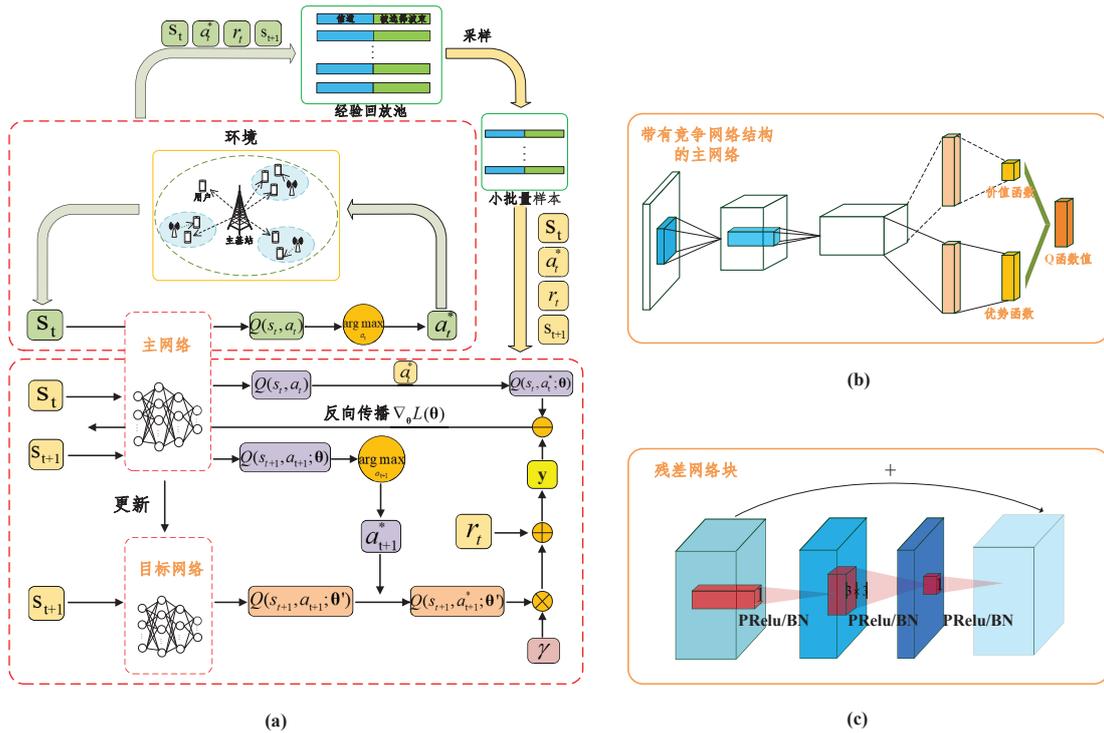


图 4.4 DRL网络结构：(a)基于DRL网络的波束选择设计；(b)竞争网络结构；(c)残差网络块结构

关性，引入经验回放池 \mathcal{D} 用于存储训练样本 (s_t, a_t, r_t, s_{t+1}) 。如图4.4所示，这些样本是通过在策略 π 下，智能体基于当前状态选择动作，并与环境交互得到的。一小批存储在 \mathcal{D} 中的样本，将会根据优先级，被采样用于网络训练。训练过程中，性能较差的样本，将会被给予更高的优先级，希望被更频繁地采样用于网络训练。

- ϵ -贪婪策略^[217]: 使用的策略：以概率 ϵ 随机选择一个动作 a_t ，以概率 $1 - \epsilon$ 选择动作 $a_t = \max_a Q^{\pi^*}(s_t, a; \theta)$ ，其中 ϵ 随时间递减。该策略会将更多的动作(可选择的波束)引入训练样本中，增强网络的探索性。
- 带噪声的网络: 为了提升网络的探索能力，对网络中的全连接层的训练参数增加噪声。

4.4.3 深度强化学习网络结构

4.4.3.1 DDQN

如果用同一个网络来选择动作和评估动作的价值，Q函数往往会被过高地估计。为了避免动作的价值被高估，采用DDQN的结构，其中有两个网络分别用于选择动作和评估动作的价值，分别被称作主网络和目标网络。图4.4(a)描述了带有DDQN的DRL网络

架构。具体而言，下一个时刻的状态 s_{t+1} 被主网络和目标网络分别用于选择动作和评估Q值 $Q(s_{t+1}, a_{t+1}; \theta')$ 。随后，目标值 y 通过折现因子 γ 和奖励函数 r_t 计算得到。最后，误差通过将目标值 y 和最优值 $Q(s_t, a_t^*; \theta)$ 做差得到，该误差最终用于反向传播更新训练参数 θ 。

4.4.3.2 竞争网络结构

Q函数描述了在状态 s 执行某个动作 a 所能获得的价值。采用竞争网络结构来估计价值函数 $V(s)$ 和优势函数 $A(s, a) = Q(s, a) - V(s)$ ，其中 $A(s, a)$ 描述的是动作 a 相比较其他动作的优势。因此，如图4.4(b)所示，DDQN的最后一层被分成了两个子网络分别用于估计 $V(s)$ 和 $A(s, a)$ ，限制 $\sum_a A(s, a) = 0$ 。其中， $V(s)$ 可以被解释为是所有备选动作在状态 s 价值的平均值， $A(s, a)$ 评估的是动作 a 相对于平均值 $V(s)$ 的价值。根据 $V(s)$ 和 $A(s, a)$ ，动作价值函数 $Q(s, a)$ 可以被更准确地估计。

4.4.3.3 网络架构

针对主网络和目标网络，采用“移动网络”架构，包含若干级联的残差块和一个全连接层。每一个残差块由包含 3×3 和 1×1 卷积模板的卷积层，“PReLU”函数，和批量归一化层组成，如图4.4(c)所示。和全连接网络相比，该网络的训练参数更少，收敛速度更快，性能更稳定。根据上述设计，针对波束选择的DRL网络的训练过程总结在算法4.1中。

4.5 基于模型驱动网络的数字预编码设计

本节提出基于模型驱动网络的数字预编码设计，并分析了计算复杂度。

4.5.1 迭代WMMSE算法

基于被选择的 N_{RF} 个波束，波束选择矩阵 \mathbf{F} 可以被构建，从而得到等效信道矩阵 $\bar{\mathbf{H}}^H = \mathbf{H}^H \mathbf{F}$ 。在给定矩阵 \mathbf{F} 的前提下，和速率最大化问题(4.6)可以被表示成如下形式

$$\max_{\{\mathbf{p}_k\}} \sum_{k=1}^K \log \left(1 + \frac{|\bar{\mathbf{h}}_k^H \mathbf{p}_k|^2}{\sum_{i \neq k} |\bar{\mathbf{h}}_k^H \mathbf{p}_i|^2 + \sigma^2} \right) \quad (4.18a)$$

$$\text{s.t.} \quad \sum_{k=1}^K \text{Tr}(\mathbf{p}_k \mathbf{p}_k^H) \leq P_s, \quad (4.18b)$$

其中 $\bar{\mathbf{H}} \triangleq [\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \dots, \bar{\mathbf{h}}_K] \in \mathbb{C}^{N_{RF} \times K}$ 和 $\bar{\mathbf{h}}_k$ 表示用户 k 的等效信道矢量，即 $\bar{\mathbf{h}}_k^H = \mathbf{h}_k^H \mathbf{F}$ 。文献[86]证明了下述MMSE问题(4.19)等价于和速率最大化问题(4.18)，即这两个问题的最优

表 4.1 DRL网络的训练过程

算法 4.1 DRL网络的训练过程

-
- 1: **输入:** 包含信道样本的训练数据集, 样本批次大小 B , 学习率 η , 经验回放周期 Q , 经验回放池容量大小 D , 以及周期数 J ;
 - 2: **输出:** 用于选择波束的DRL网络及其训练参数 θ ;
 - 3: **初始化:** 经验回放池 $\mathcal{D} = \emptyset$, 随机初始化动作价值函数 $Q^\pi(s, a)$ 及网络训练参数 θ ;
 - 4: **for** $j = 1 : J$ **do**
 - 5: 观察到初始状态 s_0 , 并选择动作 $a_0 \sim \pi_\theta(s_0)$;
 - 6: **for** $t = 1 : T$ **do**
 - 7: 以概率 ϵ 随机选择一个动作 a_t , 以概率 $1 - \epsilon$ 选择动作 $a_t = \max_a Q^{\pi^*}(s_t, a; \theta)$;
 - 8: 执行动作 a_t , 根据章节4.4.2.2, 与环境交互得到奖励函数 r_t ;
 - 9: 根据状态转移, 得到下一个状态 s_{t+1} , 将该状态转移对应的四元组 (s_t, a_t, r_t, s_{t+1}) 存储至 \mathcal{D} ;
 - 10: **end for**
 - 11: **if** $j = 0 \bmod Q$ **then**
 - 12: 从经验回放池 \mathcal{D} 中, 随机采样批次大小为 B 的状态转移四元组 (s_t, a_t, r_t, s_{t+1}) ;
 - 13: 计算损失函数(4.12)的值, 并根据公式(4.14), 采用SGD更新网络训练参数 θ ;
 - 14: 每间隔一段时间, 将训练参数 θ 拷贝到目标网络。
 - 15: **end if**
 - 16: **end for**
-

解是相同的。

$$\min_{\{\mathbf{p}_k, w_k, u_k\}} \sum_{k=1}^K w_k e_k - \log w_k \quad (4.19a)$$

$$\text{s.t.} \quad \sum_{k=1}^K \text{Tr}(\mathbf{p}_k \mathbf{p}_k^H) \leq P_s, \quad (4.19b)$$

其中 u_k 和 w_k 为引入的辅助变量, 以及

$$e_k \triangleq |u_k \bar{\mathbf{h}}_k^H \mathbf{p}_k|^2 - 2\Re(u_k \bar{\mathbf{h}}_k^H \mathbf{p}_k) + 1 + \sigma^2 |u_k|^2 + \sum_{i \neq k} |u_k \bar{\mathbf{h}}_k^H \mathbf{p}_i|^2.$$

文献[86]提出迭代WMMSE算法求解该问题, 该算法交替优化这三个变量 $\{u_k, w_k, \mathbf{p}_k\}$, 优化某一变量时另两个变量固定。迭代的闭式表达式为

$$u_k = \left(\sum_{i=1}^K \bar{\mathbf{h}}_k^H \mathbf{p}_i \mathbf{p}_i^H \bar{\mathbf{h}}_k + \sigma^2 \right)^{-1} \mathbf{p}_k^H \bar{\mathbf{h}}_k, \forall k, \quad (4.20a)$$

$$w_k = (1 - u_k \bar{\mathbf{h}}_k^H \mathbf{p}_k)^{-1}, \forall k, \quad (4.20b)$$

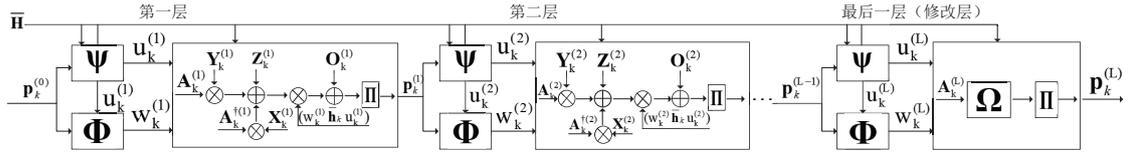


图 4.5 用于数字预编码设计的模型驱动网络结构

$$\mathbf{p}_k = \left(\sum_{k=1}^K w_k \bar{\mathbf{h}}_k u_k^* u_k \bar{\mathbf{h}}_k^H + \lambda \mathbf{I} \right)^{-1} w_k \bar{\mathbf{h}}_k u_k, \forall k, \quad (4.20c)$$

其中 λ 表示拉格朗日乘子。迭代 WMMSE 算法的步骤是迭代地执行 (4.20a), (4.20b) 和 (4.20c), 直至 (4.19a) 收敛。根据迭代 WMMSE 算法的表达式 (4.20a)-(4.20c), 接下来的章节设计新颖的模型驱动网络。

4.5.2 基于模型驱动网络的数字预编码设计

本节设计模型驱动网络用于数字预编码, 其中迭代 WMMSE 算法被展开为类似于 DNN 的逐层结构, 如图 4.5 所示。因为表达式 (4.20c) 中有高维度的矩阵求逆操作且通常需要较多的迭代次数, 所以迭代 WMMSE 算法的计算复杂度很高。本节设计的模型驱动网络可以使用较少的迭代次数并避免矩阵求逆来降低计算复杂度。此外, 通过引入训练参数来提升系统的和速率性能, 并且避免迭代 WMMSE 算法中拉格朗日乘子的二分搜索。

首先, 定义新颖的非线性函数, 将矩阵 \mathbf{A} 的对角线元素取倒数, 并将非对角线元素置成 0, 记作 \mathbf{A}^\dagger 。以 3×3 的矩阵为例,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \mathbf{A}^\dagger = \begin{bmatrix} \frac{1}{a_{11}} & 0 & 0 \\ 0 & \frac{1}{a_{22}} & 0 \\ 0 & 0 & \frac{1}{a_{33}} \end{bmatrix}. \quad (4.21)$$

注意到当 \mathbf{A} 为对角阵时, $\mathbf{A}^{-1} = \mathbf{A}^\dagger$ 。在迭代 WMMSE 算法中, 待求逆矩阵的对角线元素显著大于非对角线元素。因此, \mathbf{A}^\dagger 为 \mathbf{A}^{-1} 较好的一个近似。矩阵求逆操作 \mathbf{A}^{-1} 由如下两部分近似: (1) $\mathbf{A}^\dagger \mathbf{X}$: 含有逐元素非线性函数 \mathbf{A}^\dagger 和训练参数 \mathbf{X} ; (2) 回顾矩阵求逆 \mathbf{A}^{-1} 在 \mathbf{A}_0 点的一阶泰勒展开: $\mathbf{A}^{-1} = 2\mathbf{A}_0^{-1} - \mathbf{A}_0^{-1} \mathbf{A} \mathbf{A}_0^{-1}$, 参照该矩阵形式的泰勒展开结构, 采用网络结构 $\mathbf{A}\mathbf{Y} + \mathbf{Z}$, 其中包含训练参数 \mathbf{Y} 和 \mathbf{Z} 。

图 4.5 描述了用于设计数字预编码矩阵的模型驱动网络结构, 其中 Ψ , Φ 和 Ω 分别表示迭代过程 (4.20a), (4.20b) 和 (4.20c)。将矩阵求逆操作 \mathbf{A}_k^{-1} 替换成 $\mathbf{A}_k^\dagger \mathbf{X}_k + \mathbf{A}_k \mathbf{Y}_k + \mathbf{Z}_k$, 其中 $\mathbf{A}_k \triangleq \left(\sum_{k=1}^K w_k \bar{\mathbf{h}}_k u_k^* u_k \bar{\mathbf{h}}_k^H + \lambda_k \mathbf{I} \right)$ 。为了避免拉格朗日乘子的二分搜索, 针对变量 \mathbf{p}_k 的迭

代, 引入 λ_k 作为训练参数。因此, 在模型驱动网络中, \mathbf{p}_k 的迭代表达式(4.20c)写成

$$\mathbf{p}_k^l = \left((\mathbf{A}_k^l)^\dagger \mathbf{X}_k^l + \mathbf{A}_k^l \mathbf{Y}_k^l + \mathbf{Z}_k^l \right) w_k^l \bar{\mathbf{h}}_k u_k^l + \mathbf{O}_k^l, \forall k, \quad (4.22)$$

其中 $(\cdot)^{(l)}$ 表示第 l 层, $\Theta \triangleq \{\mathbf{X}_k^l, \mathbf{Y}_k^l, \mathbf{Z}_k^l, \mathbf{O}_k^l, \lambda_k^l\}$ 为引入的训练参数。为了使得功率约束(4.18b)满足, 采用如下投影算子

$$\prod\{\mathbf{p}_k\} = \begin{cases} \mathbf{p}_k, & \sum_{k=1}^K \text{Tr}(\mathbf{p}_k \mathbf{p}_k^H) \leq P_s, \\ \frac{\sqrt{P_s}}{\|\mathbf{P}\|} \mathbf{p}_k, & \text{其他情况,} \end{cases} \quad (4.23)$$

其中 $\mathbf{P} \triangleq [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K]$ 。为了提升性能, 在模型驱动网络的最后一层采用 \mathbf{p}_k 的迭代表达式(4.20c), 即在最后一层不对矩阵求逆操作做近似。

模型驱动网络的训练过程总结如下。首先, 从训练数据集中选择一批样本 $\bar{\mathbf{H}}^H$ 并输入模型驱动网络。模型驱动网络输出数字预编码矩阵 \mathbf{P} 并对其进行功率归一化, 使得功率约束满足。接着, $\{\bar{\mathbf{H}}^H, \mathbf{P}\}$ 输入(4.18a)计算和速率。随后, 计算目标函数(4.18a)对于训练参数 $\{\mathbf{X}_k^l, \mathbf{Y}_k^l, \mathbf{Z}_k^l, \mathbf{O}_k^l, \lambda_k^l\}$ 的梯度。最后, 计算网络各层训练参数的梯度, 并使用SGD更新参数。

算法4.2提出了一种新颖的联合训练两个网络的训练方法, 该方案和现有的联合训练方案不同, 因为训练参数的梯度并不能直接从模型驱动网络反向传递到模型驱动网络。提出的训练方式从以下两个方面对网络进行联合训练: (1) DRL网络的奖励函数(4.15)的计算需要基于数字预编码矩阵 \mathbf{P} , 且 \mathbf{P} 是由模型驱动网络计算的; (2) 模型驱动网络的训练样本 $\bar{\mathbf{H}}$ 是由DRL网络产生的。

注 4.2. 提出的联合训练网络框架相对已有的对两个模块分开设计框架的优点主要在于: (1) 对波束选择和数字预编码矩阵分开设计的方案有一定的性能损失, 因为该方案并不直接优化波束选择矩阵来最大化和速率, 而是依赖于启发式算法最大化其他指标, 比如波束能量; (2) 提出的联合设计网络框架具有更低的计算复杂度, 并且在对抗信道估计误差上具有更强的鲁棒性。

4.5.3 计算复杂度分析

首先, 分析提出的联合训练网络的计算复杂度, 由两部分组成: DRL网络的计算复杂度和模型驱动网络的计算复杂度。DRL网络的计算复杂度为 $\mathcal{O}_d \triangleq \mathcal{O}\left(\sum_{l=1}^{L-1} Q_l^2 S_l^2 C_{l-1} C_l + \dots\right)$

表 4.2 DRL网络和模型驱动网络的联合训练过程

算法 4.2 DRL网络和模型驱动网络的联合训练过程

-
- 1: **输入:** 算法4.1中DRL网络第1行的输入; 输入模型驱动网络的参数, 即网络层数 L , 批次大小 B , 收敛精度 ϵ ;
 - 2: **输出:** 用于选择波束的DRL网络和用于设计数字预编码矩阵的模型驱动网络及其参数;
 - 3: **初始化:** 采用算法4.1中的第3行对DRL网络进行参数初始化; 随机初始化模型驱动网络的训练参数 $\{\mathbf{X}_k^l, \mathbf{Y}_k^l, \mathbf{Z}_k^l, \mathbf{O}_k^l, \lambda_k^l\}$;
 - 4: **while** DRL网络的损失函数或模型驱动网络的和速率目标函数未收敛 **do**
 - 5: **for** $j = 1 : J$ **do**
 - 6: 执行算法4.1的第5-10行一个循环周期, 生成四元组 (s_t, a_t, r_t, s_{t+1}) 和 $\bar{\mathbf{H}}^H$;
 - 7: 将 $\bar{\mathbf{H}}^H$ 输入模型驱动网络执行一次前向传播得到 \mathbf{P} , 根据 $\bar{\mathbf{H}}^H$ 和 \mathbf{P} 计算奖励函数(4.15);
 - 8: 将四元组 (s_t, a_t, r_t, s_{t+1}) 和 $\bar{\mathbf{H}}^H$ 存入经验回放池作为训练集, 用于训练DRL网络和模型驱动网络;
 - 9: **end for**
 - 10: **if** $j = 0 \bmod Q$ **then**
 - 11: **训练DRL网络:** 执行算法4.1的第12-14行;
 - 12: **训练模型驱动网络:** 从训练数据集中选择一个批次的样本 $\bar{\mathbf{H}}^H$, 将其输入模型驱动网络, 执行前向传播; 计算和速率目标函数(4.18a)对模型驱动网络训练参数的梯度, 执行反向传播并采用SGD来更新模型驱动网络的训练参数。
 - 13: **end if**
 - 14: **end while**
-

$KN_{RF}C_{L-1}C_{out}$), 其中 L 表示网络层数, S_l 表示第 l 层的卷积核的大小, C_l 是第 l 层的通道数, C_{out} 代表全连接层输出的维度, Q_l 表示第 l 层的输出维度。注意到 Q_l 依赖于输入维度、填零数和步长。本章取 $S_l = 5$ 和 $C_l = 10$ 。模型驱动网络的计算复杂度为 $\mathcal{O}(I_n(K^2N_{RF}^2 + KN_{RF}^{2.37}))$, 其中 I_n 表示层数。相比迭代WMMSE算法的计算复杂度, 即 $\mathcal{O}(I_m(K^2N_{RF}^2 + KN_{RF}^3))$, 模型驱动网络的计算复杂度更低, 原因如下: (1) 模型驱动网络的层数远小于迭代WMMSE算法, 即 $I_n < I_m$; (2) 迭代WMMSE算法包含高维度矩阵求逆操作, 其计算复杂度为 $\mathcal{O}(N_{RF}^3)$, 而模型驱动网络只包含矩阵乘法, 其计算复杂度为 $\mathcal{O}(N_{RF}^{2.37})$ 。表4.3给出了如下部分求解波束选择和数字预编码问题算法的计算复杂度, 且其性能对比将在下一小节给出:

- **DDQN-Unfold:** 本章提出的联合网络设计, 采用具有DDQN结构的DRL网络用于波束选择, 同时采用模型驱动网络用于数字预编码矩阵的设计。

表 4.3 不同算法的计算复杂度比较

算法	波束选择	数字预编码	整体 ($M_s \gg N_{RF} \geq K$)
DDQN-Unfold	\mathcal{O}_d	$\mathcal{O}(I_n(K^2 N_{RF}^2 + K N_{RF}^{2.37}))$	$\mathcal{O}_d + \mathcal{O}(I_n(K^2 N_{RF}^2 + K N_{RF}^{2.37}))$
PDD	$\mathcal{O}(I_{p_1} I_{p_2} M_s^2 N_{RF} K)$	$\mathcal{O}(I_{p_1} I_{p_2} M_s^3)$	$\mathcal{O}(I_{p_1} I_{p_2} (M_s^2 N_{RF} K + M_s^3))$
MM-WMMSE	$\mathcal{O}(M_s \log M_s)$	$\mathcal{O}(I_m(K^2 N_{RF}^2 + K N_{RF}^3))$	$\mathcal{O}(M_s \log M_s + I_m K N_{RF}^3)$
IA-ZF	$\mathcal{O}(N_{RF} M_s)$	$\mathcal{O}(N_{RF}^3)$	$\mathcal{O}(N_{RF} M_s + N_{RF}^3)$
MS-ZF	$\mathcal{O}(N_{RF} K M_s^2)$	$\mathcal{O}(N_{RF}^3)$	$\mathcal{O}(N_{RF} K M_s^2)$
FD-ZF	–	$\mathcal{O}(M_s^3)$	$\mathcal{O}(M_s^3)$
FD-WMMSE	–	$\mathcal{O}(I_w K M_s^3)$	$\mathcal{O}(I_w K M_s^3)$

- **DQN-Unfold**: 采用具有DQN结构的DRL网络用于波束选择，即一个网络同时用于选择动作和评估动作价值，同时采用模型驱动网络用于数字预编码矩阵设计。
- **B-DNN**: 两个具有全连接结构的黑盒网络分别被用于优化波束选择矩阵和数字预编码矩阵。波束选择问题可以被建模为多分类问题^[133]，其中波束选择网络的输入是波束空间信道矩阵，输出每个波束被选择的概率，选择概率最高的 N_{RF} 个波束作为被选择的波束。随后，数字预编码网络的输入是被选择的波束，输出是数字预编码矩阵。此外，损失函数为用户的和速率，问题(4.6)的约束被惩罚到损失函数中。具体而言，等式约束 $f(x) = 0$ 被惩罚为 $\rho|f(x)|$ ，不等式约束 $g(x) \leq 0$ 被惩罚为 $\varrho \text{ReLU}(g(x))$ ，其中 $\text{ReLU}(x) = \max(0, x)$ ， ρ 和 ϱ 是惩罚因子。
- **PDD**: 基于PDD迭代优化算法的波束选择和数字预编码设计^[127]。
- **MM-WMMSE**: 基于最大幅度的波束选择方案和迭代WMMSE算法的数字预编码设计。
- **IA-ZF**: 基于干扰消除的波束选择方案，迫零数字预编码设计^[126]。
- **MS-ZF**: 基于最大化SINR的波束选择方案，迫零数字预编码设计^[129]。
- **FD-ZF**: 纯数字系统（不包含波束选择模块，射频链路数等于天线数），迫零数字预编码设计。
- **FD-WMMSE**: 纯数字系统，基于迭代WMMSE算法的纯数字预编码设计^[86]。

关于干扰消除（Interference-Aware, IA）波束选择算法和最大化SINR（Maximization of SINR, MS）波束选择算法的详细介绍请见附录A.3.2，将上述两种启发式的波束选择算法作为基线算法。在表4.3中， I_{p_1} 和 I_{p_2} 分别表示PDD算法的内层和外层迭代次数，

I_m 和 I_w 分别表示MM-WMMSE算法和FD-WMMSE算法的迭代次数。注意到DQN-Unfold算法的计算复杂度和提出的DDQN-Unfold是相同的。根据波束选择和数字预编码的计算复杂度分析,表4.3中的计算复杂度是基于条件 $M_s \gg N_{RF} \geq K$,通信系统通常都满足该条件。由表可知,本章提出的DDQN-Unfold算法计算复杂度和 M_s 无关,从而显著降低了计算复杂度。从表4.3中可以看到,FD-ZF和FD-WMMSE有很高的计算复杂度。此外,这类纯数字系统要求射频链路数等于天线数,这导致了极大的硬件开销,使其难以应用于实际系统。相比之下提出的DDQN-Unfold算法计算复杂度较低,同时具有逼近纯数字系统的性能,详细性能分析和结果请见下一小节。此外,相比PDD和MM-WMMSE算法,IA-ZF和MS-ZF算法有更低的计算复杂度,但它们的性能较差。因此,本章提出的DDQN-Unfold算法以较低的计算复杂度实现了较好的系统性能。

4.6 仿真结果

本节通过数值仿真来评估上述算法的性能。

4.6.1 仿真设置

通过测试数据集来评估所提出的联合网络设计的和速率性能。系统配置如下:基站配备DLA,包含 $M_s = 256$ 根天线的线性阵列,以及 $N_{RF} = 20$ 条射频链路,共服务 $K = 18$ 个用户,发送功率和噪声功率分别被设置为30 dBm和-10 dBm。信道模型的参数根据参考文献[128]进行设置:(1)信道包含1条LoS径和3条NLoS径;(2) $\rho_k^{(0)} \sim \mathcal{CN}(0, 1)$ 并且 $\rho_k^{(l)} \sim \mathcal{CN}(0, 0.1)$, $l = 1, 2, 3$;(3) $\phi_k^{(0)}$ 和 $\phi_k^{(l)}$ 均是从 $[-1, 1]$ 范围内通过随机采样产生;(4) $\rho_k^{(0)}$, $\rho_k^{(l)}$, $\phi_k^{(0)}$,和 $\phi_k^{(l)}$ 之间相互独立。DRL网络的训练参数设置为:衰减率 $\eta = 0.9$,缓存容量 $D = 16000$,训练批次大小 $B = 40$,数据重新播放的周期 $Q = 10$,选择重复波束的惩罚 $\rho = -50$ 。模型驱动网络的层数为 $L = 6$ 。训练和测试的样本数分别设置为5000和1000。

本节的仿真实验在“Python 3.6”、“Pytorch 1.5.0”的环境下进行。由于该平台不能直接处理复数矩阵操作,将复数矩阵 $\mathbf{A} \in \mathbb{C}^{a \times b}$ 转换成维度为 $2 \times a \times b$ 的实数张量,其中实部 $\Re\{\mathbf{A}\}$ 和虚部 $\Im\{\mathbf{A}\}$ 分开存储。随后,两个复数矩阵 \mathbf{A} 和 \mathbf{B} 相乘得到的复数矩阵,其实部和虚部分别为 $(\Re\{\mathbf{A}\}\Re\{\mathbf{B}\} - \Im\{\mathbf{A}\}\Im\{\mathbf{B}\})$ 和 $(\Re\{\mathbf{A}\}\Im\{\mathbf{B}\} + \Im\{\mathbf{A}\}\Re\{\mathbf{B}\})$ 。对于复数矩阵和行列式操作,根据下式使用“Numpy”工具包计算该操作的梯度闭式表达式,重写并覆盖了“Pytorch 1.5.0”环境中的自动微分代码

$$d \log \det(\mathbf{X}) = \text{Tr}(\mathbf{X}^{-1}d\mathbf{X}), \quad d\text{Tr}(\mathbf{X}^{-1}) = -\text{Tr}(\mathbf{X}^{-1}(d\mathbf{X})\mathbf{X}^{-1}), \quad (4.24)$$

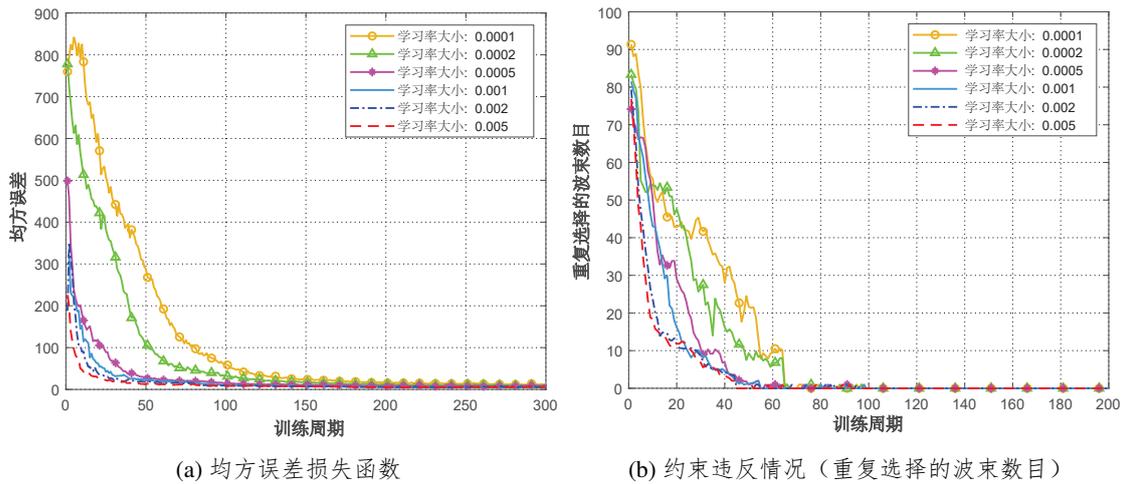


图 4.6 联合设计网络的收敛性能

该做法提供了更高的梯度精度，并加快了网络的训练。

4.6.2 联合设计网络的收敛性

首先展示联合设计网络的收敛性能。图4.6(a)描述了采用不同大小的学习率进行训练时，均方误差（Mean Square Error, MSE）损失函数的收敛性能。可以观察到，采用较小的学习率可以实现更好的MSE性能，而较大的学习速率可以带来更快的收敛速度。图4.6(b)描述了网络训练过程中的约束违反情况，即包含1000个被选定波束的50个波束空间信道样本中，被重复选择波束的数量。当被重复选择波束的数量等于0时，系统满足约束(4.6d)，且越大的被重复选择波束数量意味着约束(4.6d)被更严重地违反。可以看到，该约束在训练开始时被严重违反，但在训练了大约100轮周期后得到了满足，其中更大的学习率会带来更快的收敛速度，使得约束被更快地满足。

图4.7展示了采用不同大小的学习率训练时，联合设计网络的和速率性能。与图4.6(a)中的MSE相似，较小的学习率实现了更好的和速率性能且具有更稳定的收敛性能，而较大的学习速率能够带来更快的收敛速度。

4.6.3 和速率性能

图4.8描述了当用户数 $K = 16$ 时，在不同射频链数 N_{RF} 下各算法实现的和速率性能。由图可知，所有算法实现的和速率性能均随着射频链数的增加而提升。提出的联合设计网络DDQN-Unfold和纯数字预编码方案FD-ZF之间的性能差距会随着 N_{RF} 的增加而降低。另外，在射频链数量较大时，比如 $N_{RF} = 30$ ，DDQN-Unfold的和速率性能逼近FD-ZF。随

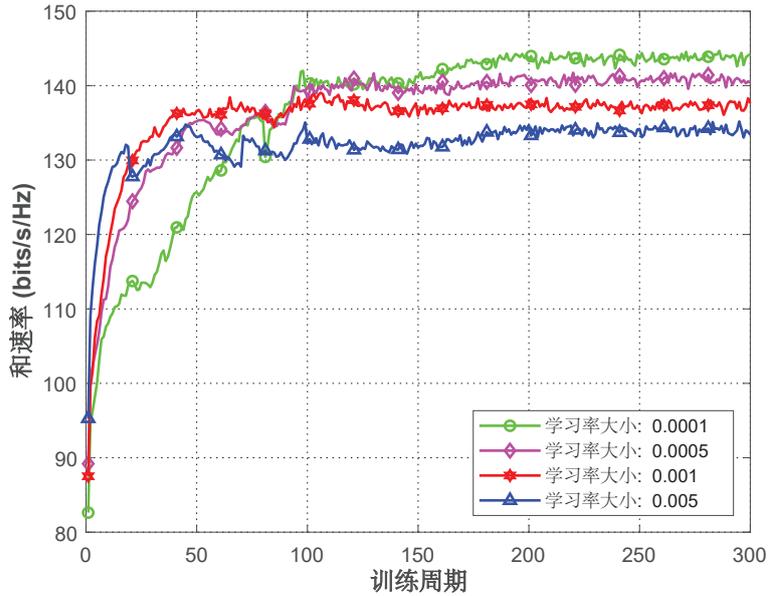


图 4.7 联合设计网络的和速率收敛性能

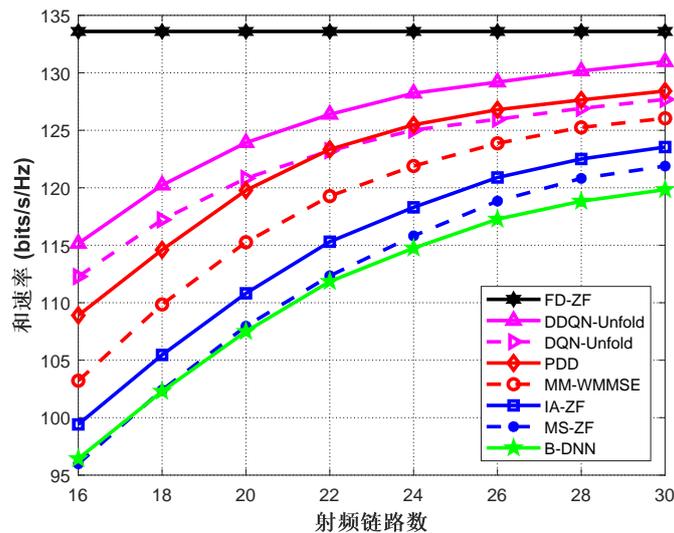


图 4.8 系统和速率性能随射频链路数 N_{RF} 的变化

随着 N_{RF} 的增加，和速率性能会逐渐接近其瓶颈，并且增速减缓，但难以完全达到FD-ZF实现的性能上界。主要原因在于该性能上界是由全数字预编码架构实现的，而本章采用的离散透镜阵列架构在硬件复杂度和功耗上远低于全数字预编码架构，包含的功率约束和硬件约束更为严格，故在性能上会略有损失。从仿真结果可知，FD-WMMSE的和速率性能最好，随后依次是FD-ZF、DDQN-Unfold、DQN-Unfold、PDD、MM-WMMSE、IA-ZF、MS-ZF、B-DNN。采用DDQN的联合网络设计DDQN-Unfold优于采用DQN的联合网络设计DQN-Unfold，这是因为DQN倾向于高估某些动作的Q值，从而选择实际性能不够好的波束。PDD和MM-WMMSE优于IA-ZF和MS-ZF，主要是因为前两种方案将选

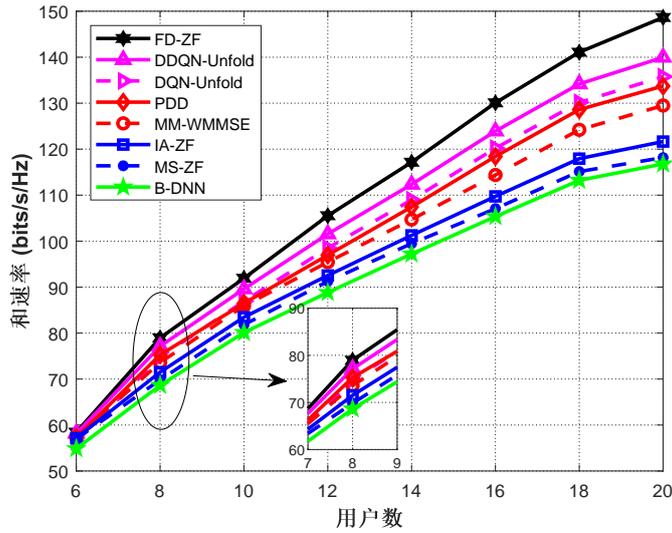


图 4.9 系统和速率性能随用户数 K 的变化

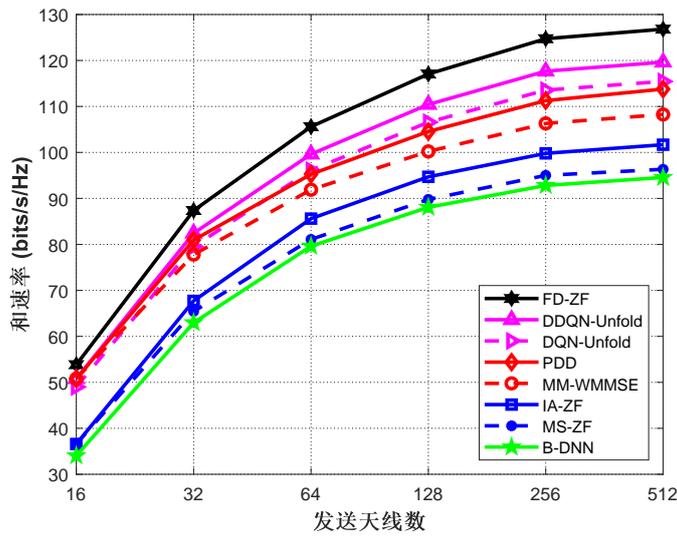


图 4.10 系统和速率性能随天线数 M_s 的变化

代WMMSE算法应用于数字预编码设计，其性能优于ZF预编码。此外，PDD和IA-ZF分别优于MM-WMMSE和MS-ZF，因为PDD和IA选择的波束比MM和MS更好。

图4.9呈现了当射频链路数 $N_{RF} = 22$ 时，系统和速率性能随用户数 K 的变化。由图可见，所有算法实现的和速率性能随着用户数的增加而提升。DDQN-Unfold能够实现接近FD-ZF纯数字预编码的性能，并且性能差距会随着用户数的增加而增加。从结果可知，提出的DDQN-Unfold提供了比其他基线算法更好的和速率，证明了其在降低用户间干扰方面的优势。此外，其他算法与DDQN-Unfold之间的性能差距随着用户数的增加而扩大，这表明DDQN-Unfold具有应用于包含大量用户的大规模系统的潜力。

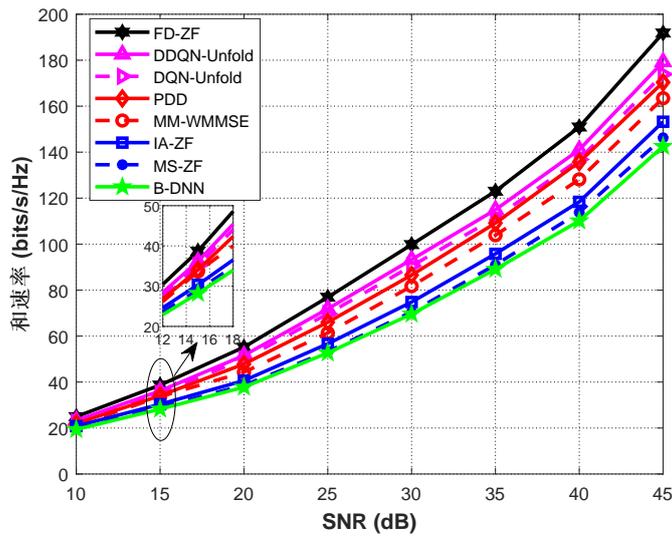


图 4.11 系统和速率性能随SNR的变化

图4.10展示了系统和速率性能随天线数 M_s 的变化。当 $M_s = 16$ 时，DDQN-Unfold实现了与PDD和MM-WMMSE几乎相同的和速率性能。这主要是因为只有16个备选波束用于16条射频链路（即所有波束都被选择），即这三种方案选择的波束相同，且设计的数字预编码矩阵也几乎相同。类似地，IA-ZF和MS-ZF也实现了相同的性能，因为它们都采用了ZF预编码，选择的波束和设计的数字预编码矩阵都相同。此外，当 $M_s = 16$ 时，WMMSE和ZF预编码算法的不同是造成前三种方案与后两种方案之间差距的主要原因。可以看到，所有方案的性能都会随着 M_s 的增加而提升，且FD-ZF实现了最佳的和速率性能，后续依次是DDQN-Unfold、DQN-Unfold、PDD、MM-WMMSE、IA-ZF、MS-ZF和B-DNN。这些方案之间的差距随着 M_s 的增加而扩大，这是由于天线数量的增加会导致角度分辨率的提升，使得不同波束之间的差异变大，从而导致不同波束选择算法选择不同波束对应的性能差距随之扩大。此外，当 M_s 较大时，比如 $M_s = 256$ ，天线的角度分辨率足够大，和速率性能的提升速度随着 M_s 的增加而减慢。

图4.11描述了系统和速率性能随SNR的变化。可以观察到，所有算法实现的和速率性能随SNR单调增加。此外，DDQN-Unfold显著优于基线算法，并且性能差距随SNR的增加而扩大。这主要是因为，在低SNR下，不同波束之间的差异较小，导致不同算法得到的波束选择结果和数字预编码矩阵对应的性能差距较小。而对于高SNR，例如45 dB时，与PDD相比，DDQN-Unfold具有大约6%的性能增益，而与其他基线算法相比，则具有超过10%的性能增益。这验证了所提出的DDQN-Unfold的性能优势。因此，可以得出结论，DDQN-Unfold为该问题提供了一个高效的解决方案，尤其是在高SNR场景下。

表 4.4 不同信噪比下联合和分别训练网络的和速率性能

SNR (dB)	10	15	20	25	30	35	40
联合训练网络(bits/s/Hz)	23.32	36.16	51.56	71.83	93.27	114.95	141.07
分别训练网络	96.81%	95.93%	94.68%	92.87%	90.75%	88.42%	85.71%

表4.4比较了所提出的DDQN-Unfold联合训练（即DDQN和模型驱动网络联合训练）和分别训练实现的和速率性能。具体而言，分开训练网络是指，当训练DDQN时，基于迭代优化算法（例如：迭代WMMSE算法）计算数字预编码矩阵，随后将启发式算法选择的波束（例如：选择具有最高能量的波束）作为模型驱动网络的输入。表中的百分比是通过将分别训练网络实现的和速率除以联合训练网络实现的和速率计算得到的。可以看到，联合训练网络显著优于分别训练网络，并且两种方案之间的性能差距随着SNR的增加而扩大。

4.6.4 泛化性能

首先，分析所提出的联合训练网络DDQN-Unfold的泛化性能，以及对应提升泛化性能的方法。针对参数配置为 (N_{RF_0}, M_{s_0}, K_0) 的系统训练的网络可以直接迁移并用于具有更小的参数配置 (N_{RF_1}, M_{s_1}, K_1) 的系统，而不需要重新训练一个网络。对于DDQN子网络，具有较大参数系统的原始信道维度为 $M_{s_0} \times K_0$ ，包含 M_{s_0} 个波束。先通过降维，将信道矩阵降维为 $\bar{M}_s \times K_0$ ，其中选取 \bar{M}_s 使满足 $M_{s_1} < \bar{M}_s < M_{s_0}$ 。则已训练好的DDQN的网络实际输入信道维度为 $\bar{M}_s \times K_0$ 。为了将该网络用于测试系统 (N_{RF_1}, M_{s_1}, K_1) 对应的样本，其维度为 $M_{s_1} \times K_1$ ，需要对这些低维样本做一些处理，使得其维度和该训练好的网络输入维度 $\bar{M}_s \times K_0$ 匹配。具体而言，对较小的信道矩阵应该进行填零操作，即填充 $\bar{M}_s - M_{s_1}$ 个全0行向量和 $(K_0 - K_1)$ 个全零列向量。经过填零之后的信道样本维度为 $\bar{M}_s \times K_0$ ，可直接输入已训练好的网络。此外，DDQN的时间步骤需要设为 N_{RF_1} 。另外，对于模型驱动子网络，输入为经过波束选择之后的等效信道矩阵 $\bar{\mathbf{H}} \in \mathbb{C}^{N_{RF_0} \times K_0}$ ，但 (N_{RF_1}, M_{s_1}, K_1) 系统的测试数据维度是 $N_{RF_1} \times K_1$ 。因此，需要对 $\bar{\mathbf{H}}$ 填充 $(N_{RF_0} - N_{RF_1})$ 个全零行向量和 $(K_0 - K_1)$ 个全零列向量。此外，在特定SNR场景下进行训练得到的网络可以直接用于测试其他SNR场景对应的样本，因为SNR是网络输入的一部分。为提升网络对SNR的泛化性，训练数据集可以包括不同SNR场景的样本。

图4.12(a)描述了当系统参数配置中的射频链路数 N_{RF} 和天线数 M_s 失配的情况下DDQN-Unfold的性能。在 $K = 16$ ， $\text{SNR} = 40\text{dB}$ ， $N_{RF} = 30$ ， $M_s = 256$ 的参数配置下训练DDQN-Unfold网络，并将该网络在不同的 N_{RF} 和 M_s 上进行测试，同时固定 $K = 16$ ， $\text{SNR} = 40\text{dB}$ 。由图可知，尽管存在训练阶段和测试阶段系统参数 N_{RF} 和 M_s 的不一致，但

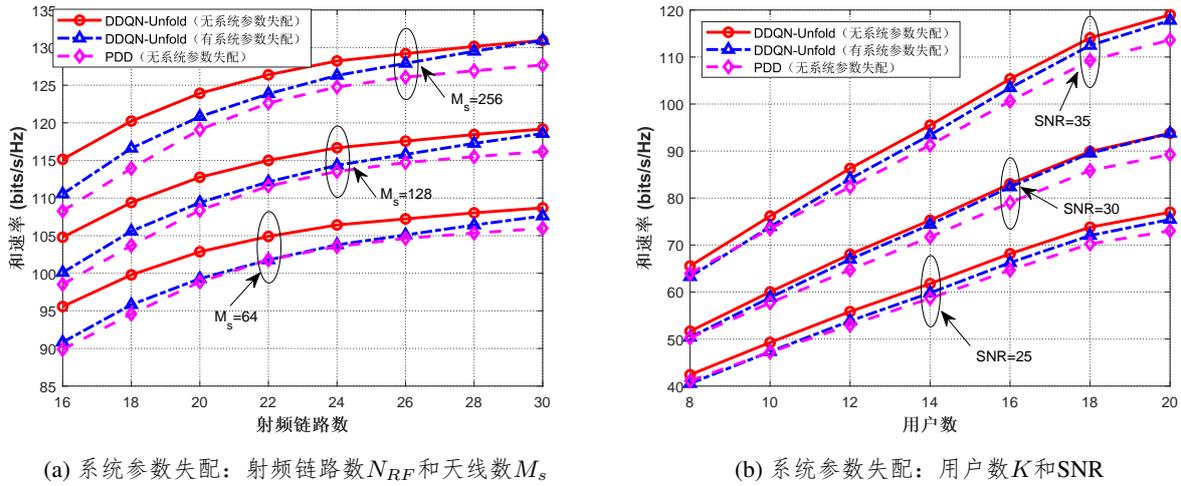


图 4.12 网络在不同系统参数失配场景下的和速率泛化性能

其导致的性能损失较小。此外，存在系统参数失配的DDQN-Unfold的性能仍然超过PDD算法，这验证了网络在不同系统参数配置 N_{RF} 和 M_s 下网络较好的泛化性。此外，该场景下由于系统参数失配造成的性能损失会随着 N_{RF} 和 M_s 增加而降低。这主要是因为当训练阶段和测试阶段的系统参数配置越接近时，网络输入、训练得到的网络也越接近，匹配程度更高，性能损失更小。

图4.12(b)展示了当系统参数配置中的用户数 K 和SNR失配的情况下DDQN-Unfold的性能。在 $K = 20$, $SNR = 30dB$, $N_{RF} = 20$, $M_s = 256$ 的参数配置下训练DDQN-Unfold网络，并将该网络在不同的 K 和SNR上进行测试，同时固定 $N_{RF} = 20$, $M_s = 256$ 。由结果可知，即使存在一定的由于参数不匹配导致的性能损失，DDQN-Unfold的性能仍然可以超过PDD算法。这验证了网络在不同系统参数配置 K 和SNR下较好的泛化性。此外，相比低SNR，在高SNR下其和速率性能更接近不存在系统参数失配场景下的性能，且性能损失也会随着 K 增加而降低。

4.6.5 公平性、鲁棒性和复杂度对比

图4.13展示了在 $K = 6$, $SNR = 30dB$, $N_{RF} = 8$, $M_s = 128$ 场景下，每个用户所选择波束的能量，其能量分布一定程度上反映了用户间的公平性。第 k 个用户所选择波束的能量为等效信道矩阵 $\bar{\mathbf{H}}$ 第 k 列的 l_2 范数，即 $\|\bar{\mathbf{h}}_k\|$ 。可以看出，DDQN-Unfold和IA所选择的不同用户间的波束能量比PDD、MM和MS更均衡，证明其可以更大程度上保证用户间的公平性。这主要是因为IA选择对齐到每个用户的波束，而DDQN-Unfold则在设计过程中考虑了用户间的公平性(4.17)。

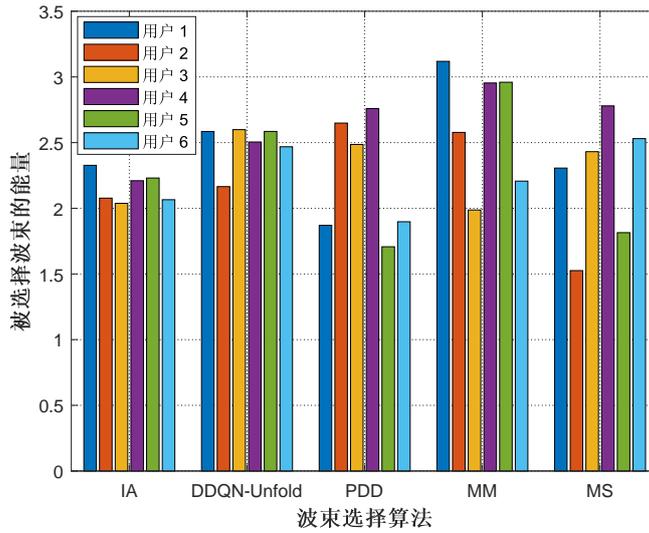


图 4.13 每个用户选择波束的能量

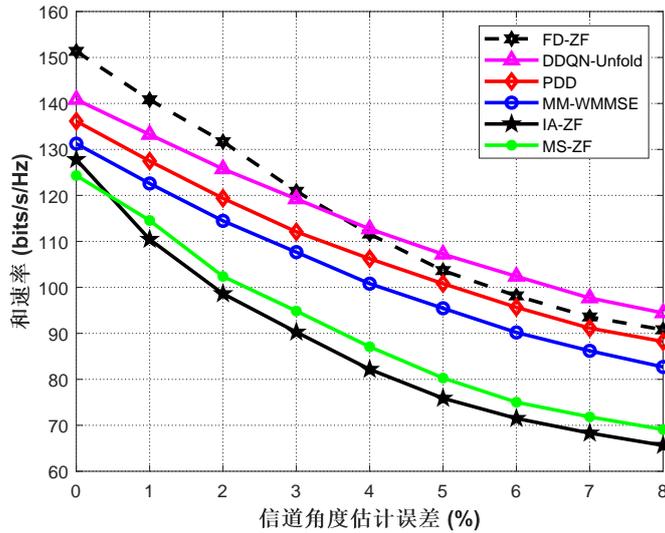


图 4.14 系统和速率随信道估计误差的变化

图4.14描述了各算法在不同信道估计误差下的和速率性能。这里的信道估计误差指信道中AoD的估计误差 $\frac{\Delta\phi}{M_s}$ ，其中 $\Delta\phi$ 表示信道模型(4.5)中LoS的真实角度和估计角度之间的误差，而 $\frac{\pi}{M_s}$ 是天线的角度分辨率。由仿真结果可知，和速率性能随着角度估计误差的增加而下降。提出的DDQN-Unfold能够在有角度估计误差的场景下最优的性能，随后依次是PDD、MM-WMMSE、MS-ZF和IA-ZF，这说明了DDQN-Unfold在处理信道估计误差方面有一定鲁棒性。另外，当角度估计误差达到4%时，FD-ZF实现的和速率性能严重下降，而DDQN-Unfold能够超过FD-ZF的性能。主要原因如下：(1) FD-ZF根据原始信道矩阵 \mathbf{H} 计算数字预编码矩阵，而其他的方案则是根据维度更低的等效信道矩阵 $\bar{\mathbf{H}}$ 来

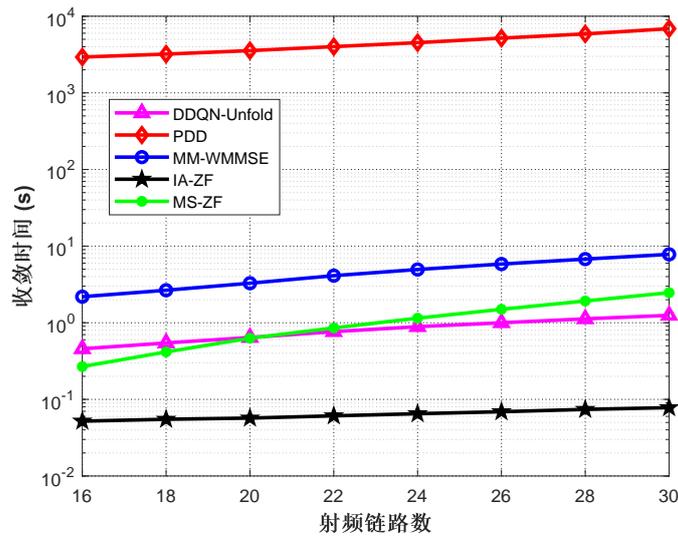


图 4.15 不同算法的收敛时间随射频链路数 N_{RF} 的变化

构建数字预编码矩阵；(2) 迭代WMMSE算法和模型驱动网络的鲁棒性相比ZF更强，因为ZF在有信道估计误差的场景下，会造成较为严重的用户间干扰。此外，其他基线算法与DDQN-Unfold之间的差距随着角度估计误差的增加而扩大，这是因为DRL网络和模型驱动网络在训练过程中见识了大量训练样本，从而提升了其鲁棒性。

图4.15展示了不同算法的收敛时间与射频链路数 N_{RF} 之间的关系，考虑用户数等于射频链路数，即 $K = N_{RF}$ 的场景。通过平均500个样本的推理时间，来近似计算DDQN-Unfold在推理阶段所需的收敛时间。由图可知，收敛时间随 N_{RF} 的增加而单调增加。与表4.3中的计算复杂度分析相比可以看出，PDD具有最高的计算复杂度且需要的收敛时间最长，其次是MM-WMMSE、MS-ZF、DDQN-Unfold和IA-ZF。当 $N_{RF} \geq 22$ 时，MS-ZF的收敛时间超过了DDQN-Unfold。综合考虑收敛时间、和速率性能和硬件成本等因素，可以得出结论，提出的DDQN-Unfold是解决联合波束选择和数字预编码设计问题的一个高效的算法，其在性能、部署成本和计算复杂度之间实现了良好的平衡。

4.7 本章小结

本章研究了带有DLA的毫米波MU-MIMO系统中的联合波束选择和预编码设计问题。为了解决这一混合整数规划难题，提出了一种高效的联合网络设计方案。具体而言，设计基于DRL的网络来优化波束选择矩阵，并采用模型驱动网络获得数字预编码矩阵，两个子网络联合设计和训练。仿真结果表明，提出的联合网络设计方案显著优于现有的迭代优化算法，且具有更低的计算复杂度和更强的鲁棒性。由此，可以得出结论：针对该问题，本

章所提出的方案可以作为迭代优化算法在实际系统中的替代方案。未来的工作可以采用本章提出的联合网络设计框架，来解决通信系统中其他具有挑战性的混合整数规划问题。其中，DRL网络和模型驱动网络可分别用于混合整数规划问题中离散变量和连续变量的优化。以无人机辅助通信系统为例，DRL网络可用于优化离散的调度变量，而模型驱动网络则可用于优化连续的无人机轨迹变量。对于带有智能反射面的通信系统，DRL网络和模型驱动网络可分别用于优化离散的经过量化的相位变量和连续的数字预编码矩阵变量。

5 基于双时间尺度端到端学习的信道获取和混合预编码设计

5.1 引言

前三章针对数字预编码、信道估计、波束选择等物理层收发机的不同模块分别设计了相应的模型驱动网络，以较低的计算复杂度实现了较好的性能。而模型驱动网络等深度学习网络架构，可以通过端到端设计和训练的方式联合设计物理层收发机的不同模块，从而进一步降低复杂度和提升系统性能。本章将提出双时间尺度端到端学习的框架，用于联合设计大规模MIMO通信系统中的信道估计、信道反馈和混合预编码等模块。

回顾具有混合预编码架构的传统大规模MIMO通信系统设计。发射端(TX)首先向接收端(RX)发送导频信号，接收端使用稀疏信号恢复算法估计角度域中的信道参数，从而恢复出CSI。随后，接收端量化估计得到的CSI相关的参数，并反馈给发送端。发送端收到反馈的量化CSI参数，从而恢复出CSI。最后，基于恢复出的CSI设计混合预编码矩阵。传统的信道估计方案、信道反馈方案和混合预编码设计方案分别单独设计。这些传统的方案通常在系统满足如下条件时才能取得良好的性能：足够长的导频长度用于信道估计，以及大量的信道反馈比特数且信道量化误差较小。

由此可见，传统通信系统设计存在两个问题：(1) 单独设计每个通信系统的模块，导致较高的复杂度和性能瓶颈，且对导频长度和信道反馈比特数要求较高；(2) 在大规模MIMO的场景下，传统混合预编码算法是根据瞬时高维原始信道设计的，而高维原始信道的获取需要大量的信令开销，从而造成较大的传输时延和严重的CSI失配。为解决问题一，端到端深度学习框架可用于联合设计这些模块^[27]。深度学习可在导频长度较短和信道反馈比特数较少的情况下，实现较好的系统性能^[219]。此外，在端到端联合设计通信系统模块的过程中，该数据驱动的框架可隐式地学习CSI的分布，因此无需精确的信道模型^[27]。另外，DNN的并行计算可显著降低其计算复杂度和运行时间。针对问题二，文献^[110]提出了双时间尺度混合预编码算法，可显著降低信令开销，提高系统对由传输时延导致的CSI误差的鲁棒性。

然而，现有的双时间尺度迭代优化算法计算复杂度较高，难以联合设计通信系统中的信道估计、信道反馈和混合预编码等模块。另外，由于模拟预编码矩阵的恒模约束以及其需要根据高维原始信道来设计，而高维原始信道矩阵的估计和反馈，以及混合预编码矩阵的联合设计具有一定的挑战性。因此，双时间尺度端到端联合收发机的设计仍是一个悬而未决的问题^[219]。为了解决这些问题，本章提出了一种基于端到端DNN的联合收发机架构，其封装了FDD大规模MIMO通信系统的所有模块，旨在最小化系统误码率（Bit-Error Rate, BER）。该架构包括：接收端的信道估计、CSI量化和反馈等模块；发送端的导频优化、数模混合预编码矩阵设计等模块。这些模块由DNN组成，且可以被联合设计和训练。具体而言，提出的DNN架构，在接收端将收到的导频信号映射为用于CSI反馈的信息比特流反馈给发送端，在发送端将该信息比特流映射为混合预编码矩阵。将所提出的DNN架构中的反馈信息比特流建模为二进制神经元的输出。为了实现基于梯度的网络反向传播训练方式，设计直通（Straight-Through, ST）梯度估计器^[265]来近似二进制层的梯度。

为了降低由传输延迟引起的CSI不匹配以及由于反馈高维原始信道矩阵导致的较大信令开销，提出了由长时间尺度网络和短时间尺度网络所组成的双时间尺度网络。时间轴被划分为一系列超帧，一个超帧内信道统计特性不变，且超帧定义了长时间尺度。每个超帧被划分为一组帧，一个帧包含固定数量的若干时隙，而时隙定义了短时间尺度，在每个时隙的时间范围内，瞬时信道样本保持不变。在每个超帧中，长时间尺度的收发端模拟预编码矩阵，每帧根据高维原始信道更新一次。短时间尺度的收发端数字预编码矩阵，每个时隙根据低维等效信道更新一次。具体而言，发送端发送低维导频，接收端通过短时间尺度网络估计低维等效信道，将量化后的低维等效CSI反馈给发送端，而发送端根据该信息进行收发端数字预编码矩阵的设计。在每一帧，发送端仅发送一次高维导频，接收端通过长时间尺度网络估计高维原始信道，将量化后的高维原始信道反馈给发送端，而发送端根据该信息，在每一帧内仅更新一次收发端模拟预编码矩阵。此外，研究了如何提升双时间尺度网络的泛化能力。通常，系统参数的变化可以分为两种类型：（1）会引起网络输入分布的变化的参数，例如：SNR的变化；（2）会引起网络中某些层的维度变化的参数，例如信道反馈比特数的变化。对于前者，基于更大范围的系统参数（例如范围更广的具有不同SNR的输入样本）训练网络。对于后者，对网络进行了改进，提出了两阶段训练方法。

本章内容安排如下。5.2节介绍了系统模型，并对FDD大规模MIMO系统中的各模块端到端联合设计问题进行了数学建模。5.3节针对研究的通信系统和场景设计了一个新颖的深度学习框架，并基于该框架提出了双时间尺度网络。5.4节中给出了实现细节，并为双时间尺度网络设计了一种两阶段训练方法。5.5节提出了增强双时间尺度网络泛化能力的方法，并分析了通信传输中的信令开销。5.6节给出了仿真结果，并在5.7节对本章进行了总结。

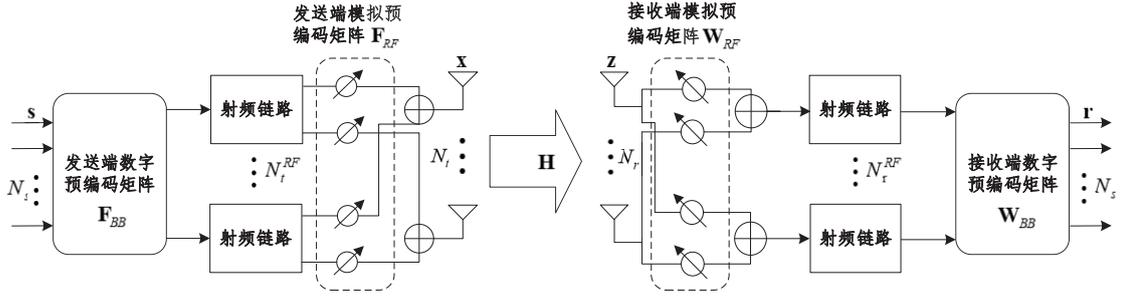


图 5.1 毫米波大规模MIMO混合预编码系统

5.2 系统模型和问题建模

本章节介绍毫米波MIMO混合预编码系统模型并进行了问题建模。

5.2.1 端到端毫米波MIMO系统

5.2.1.1 信号模型

考虑一个端到端毫米波频分双工的MIMO系统，其中发送端装配有 N_t 根发送天线和 N_t^{RF} 条射频链路，发送 N_s 个数据流至接收端，且 $N_s \leq N_t^{RF} \leq N_t$ 。接收端用户装配有 N_r 根接收天线和 N_r^{RF} 条射频链路，且 $N_s \leq N_r^{RF} \leq N_r$ 。在发送端，射频链路连接着模拟移相器网络，将 N_t^{RF} 个数字域的信号映射为 N_t 个模拟信号，传送到发送天线端。在接收端， N_r 根接收天线连接着模拟移相器网络和 N_r^{RF} 条射频链路。

发送端传输 N_s 个并行的数据 $\mathbf{S}_b \in \{0, 1\}^{N_s \times \log_2 M}$ ，由 $N_s \times \log_2 M$ 维取值为0或1的比特向量组成。采用 M 维调制方式，该数据被映射为符号矢量 $\mathbf{s} \in \mathbb{C}^{N_s \times 1}$ ，满足 $\mathbb{E}\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}_{N_s}$ 。如图5.1所示，符号矢量 \mathbf{s} 依次经过发送端数字预编码矩阵 $\mathbf{F}_{BB} \in \mathbb{C}^{N_t^{RF} \times N_s}$ 的处理，发送端模拟预编码矩阵 $\mathbf{F}_{RF} \in \mathbb{C}^{N_t \times N_t^{RF}}$ 的处理，得到经过编码的信号 $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$

$$\mathbf{x} = \mathbf{F}_{RF} \mathbf{F}_{BB} \mathbf{s}. \quad (5.1)$$

其中 \mathbf{F}_{RF} 表示仅能调相位的，通过模拟移相器实现的模拟预编码矩阵，需满足恒模约束 $|\mathbf{F}_{RF}[mn]| = 1, \forall m, n$ 。需要对数字预编码矩阵 \mathbf{F}_{BB} 做功率归一化 $\|\mathbf{F}_{RF} \mathbf{F}_{BB}\|_F^2 = P_T$ ，以满足发送端的功率约束，其中 P_T 指最大发送功率。经过预编码的信号 \mathbf{x} 经过信道。接收端天线接收到的模拟域符号矢量 $\mathbf{z} \in \mathbb{C}^{N_r \times 1}$ 可以表示为

$$\mathbf{z} = \mathbf{H} \mathbf{F}_{RF} \mathbf{F}_{BB} \mathbf{s} + \mathbf{n}, \quad (5.2)$$

其中 $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ 表示信道矩阵， $\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I}_{N_r})$ 为加性高斯噪声。

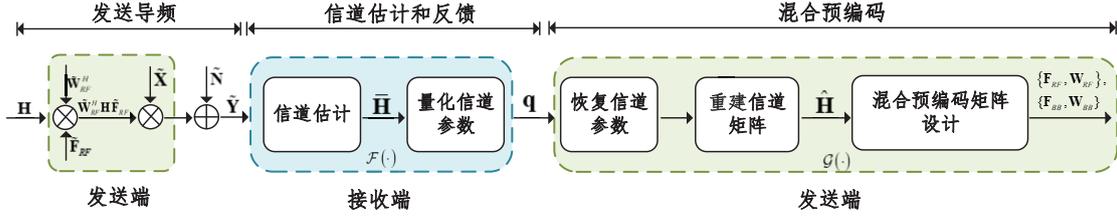


图 5.2 传统模块化的通信系统：在发送端执行导频训练和混合预编码设计，在接收端完成信道反馈。

类似地，在接收端，接收信号需依次经过接收端模拟预编码矩阵 $\mathbf{W}_{RF} \in \mathbb{C}^{N_r \times N_r^{RF}}$ ，以及接收端数字编码矩阵 $\mathbf{W}_{BB} \in \mathbb{C}^{N_r^{RF} \times N_s}$ 的处理。接收端经过处理的检测信号矢量可以写成

$$\mathbf{r} = \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \mathbf{H} \mathbf{F}_{RF} \mathbf{F}_{BB} \mathbf{s} + \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \mathbf{n}, \quad (5.3)$$

其中 \mathbf{W}_{RF} 需要满足硬件恒模约束 $|\mathbf{W}_{RF}|_{rs}| = 1, \forall r, s$ 。最终，检测信号矢量 \mathbf{r} 被解调用于恢复 N_s 个数据流，并用于恢复原始传输比特 $\hat{\mathbf{S}}_b$ 。

接下来，阐述该混合预编码系统的通信过程，如图 5.2 所示，其中包括信道估计、CSI 量化和反馈、混合预编码等模块。

5.2.1.2 基于导频训练的信道估计和信道反馈

发送端需根据信道矩阵 \mathbf{H} 来设计混合预编码矩阵。本章假定发送端和接收端没有关于信道的先验知识，所以在发送数据之前，需要通过发送导频来获取信道。发送端发送长度为 L 的导频矩阵 $\tilde{\mathbf{X}} \in \mathbb{C}^{N_t^{RF} \times L}$ ，接收端接收到信号 $\tilde{\mathbf{Y}} \in \mathbb{C}^{N_r^{RF} \times L}$ ：

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{W}}_{RF}^H \mathbf{H} \tilde{\mathbf{F}}_{RF} \tilde{\mathbf{X}} + \tilde{\mathbf{N}}, \quad (5.4)$$

其中 $\tilde{\mathbf{F}}_{RF} \in \mathbb{C}^{N_t \times N_t^{RF}}$ 和 $\tilde{\mathbf{W}}_{RF} \in \mathbb{C}^{N_r \times N_r^{RF}}$ 分别表示模拟发送导频和模拟接收导频，它们的列是从 DFT 矩阵中选出来的，以使得恒模约束满足 [158]。此外， $\mathbf{N} \in \mathbb{C}^{N_r \times L}$ 表示高斯白噪声矩阵， $\tilde{\mathbf{N}} = \tilde{\mathbf{W}}_{RF}^H \mathbf{N}$ 。第 l 次发送的导频矢量 $\tilde{\mathbf{x}}_l$ ($\tilde{\mathbf{X}}$ 的第 l 列) 需要满足功率约束，即 $\|\tilde{\mathbf{x}}_l\|^2 \leq P$ 。

接收端根据收到的信号 $\tilde{\mathbf{Y}}$ 估计出 \mathbf{H} [157–159]，并从中恢复相应信道参数，比如 AoA、AoD、信道增益等。随后，这些信息被压缩为 B 比特，并被反馈至发送端

$$\mathbf{q} = \mathcal{F}(\tilde{\mathbf{Y}}), \quad (5.5)$$

其中映射 $\mathcal{F} : \mathbb{C}^{N_r^{RF} \times L} \rightarrow \{\pm 1\}^B$ 表示压缩和反馈方案。

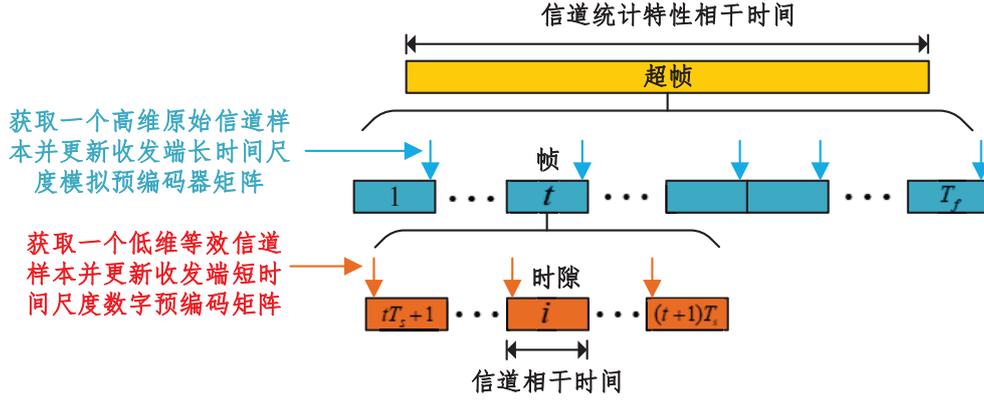


图 5.3 双时间尺度混合预编码的帧结构

5.2.1.3 混合预编码

发送端收到接收端反馈的信息 \mathbf{q} ，从中恢复出信道参数，例如：AoA、AoD、信道增益等，用于重建信道矩阵 $\hat{\mathbf{H}}$ ^[181]。随后，根据重建的信道矩阵 $\hat{\mathbf{H}}$ ，设计发送端混合预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}\}$ 和接收端混合预编码矩阵 $\{\mathbf{W}_{RF}, \mathbf{W}_{BB}\}$ ^[106-109]。在发送端完成的信道矩阵重建和混合预编码矩阵设计可以写成

$$\{\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}\} = \mathcal{G}(\mathbf{q}), \quad (5.6)$$

其中发送端恢复 \mathbf{q} 并将其映射为混合预编码矩阵，即 $\mathcal{G}(\mathbf{q})$ 。

5.2.2 双时间尺度帧结构

在实际系统中，由于对高维原始信道 \mathbf{H} 的估计和反馈所需的通信开销和计算复杂度极高，因此难以针对每个 \mathbf{H} 联合设计数字和模拟预编码矩阵。为了解决上述问题，本章提出双时间尺度方案。如图5.3所示，关注一个时间较长的超帧，在此期间信道统计特性不变。它由 T_f 个帧组成，每帧由 T_s 个时隙组成。基于这种划分，定义了以下时间尺度的概念：

- 长时间尺度：信道统计特性在每个超帧保持不变，一个超帧包含 T_f 个帧；
- 短时间尺度： \mathbf{H} 在每个时隙内保持不变。

定义低维等效信道矩阵 $\mathbf{H}_{eq} = \mathbf{W}_{RF}^H \mathbf{H} \mathbf{F}_{RF} \in \mathbb{C}^{N_r^{RF} \times N_t^{RF}}$ ，其维度远低于高维原始信道矩阵 $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ 。因此，通过发送导频，在每个时隙估计并反馈 \mathbf{H}_{eq} 是可行的。然而，在大规模MIMO场景中，在每个时隙获取实时的 \mathbf{H} 会导致极大的通信开销，只能在每帧获取一个 \mathbf{H} 样本。因此，接收端一般可以在每帧获取一个 \mathbf{H} 样本，并且能够在每个时隙获取

实时的 \mathbf{H}_{eq} 。所以，不能在每个时隙根据 \mathbf{H} 同时优化模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 和数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ 。模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 和数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ 需要分别基于高维原始信道 \mathbf{H} 和实时低维等效信道 \mathbf{H}_{eq} ，在不同的时间尺度设计。如图5.3所示，在每帧的最后一个时隙，根据估计得到的 $\hat{\mathbf{H}}$ 更新长时间尺度的模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 。在每个时隙中，根据估计得到的 $\hat{\mathbf{H}}_{eq}$ 更新短时间尺度的数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ ，此时长时间尺度的 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 保持不变。

5.2.3 问题的数学建模

双时间尺度联合信道估计、信道反馈、混合预编码问题可以被建模为

$$\min_{\mathcal{X}} \sum_{t,i} P_e(\mathbf{F}_{RF}^t, \mathbf{F}_{BB}^i, \mathbf{W}_{RF}^t, \mathbf{W}_{BB}^i) \quad (5.7a)$$

$$\text{s.t. } |[\tilde{\mathbf{F}}_{RF}]_{mn}| = 1, \forall m, n, \quad (5.7b)$$

$$|[\tilde{\mathbf{W}}_{RF}]_{rs}| = 1, \forall r, s, \quad (5.7c)$$

$$|[\mathbf{F}_{RF}^t]_{mn}| = 1, \forall m, n, t, \quad (5.7d)$$

$$|[\mathbf{W}_{RF}^t]_{rs}| = 1, \forall r, s, t, \quad (5.7e)$$

$$\|\tilde{\mathbf{x}}_l\|^2 \leq P, \forall l, \quad (5.7f)$$

$$\|\tilde{\mathbf{x}}_{eq,l}\|^2 \leq P, \forall l, \quad (5.7g)$$

$$\|\mathbf{F}_{RF}^t \mathbf{F}_{BB}^i\|_F^2 = P_T, \forall i, t, \quad (5.7h)$$

$$\mathbf{q}^t = \mathcal{F}(\tilde{\mathbf{W}}_{RF}^H \mathbf{H}^t \tilde{\mathbf{F}}_{RF} \tilde{\mathbf{X}} + \tilde{\mathbf{N}}^t), \forall t, \quad (5.7i)$$

$$\mathbf{q}_{eq}^i = \mathcal{F}_{eq}(\mathbf{H}_{eq}^i \tilde{\mathbf{X}}_{eq} + \tilde{\mathbf{N}}_{eq}^i), \forall i, \quad (5.7j)$$

$$\{\mathbf{F}_{RF}^t, \mathbf{W}_{RF}^t\} = \mathcal{G}(\mathbf{q}^t), \forall t, \quad (5.7k)$$

$$\{\mathbf{F}_{BB}^i, \mathbf{W}_{BB}^i\} = \mathcal{G}_{eq}(\mathbf{q}_{eq}^i), \forall i, \quad (5.7l)$$

其中 $\mathcal{X} \triangleq \{\mathbf{F}_{RF}^t, \mathbf{F}_{BB}^i, \mathbf{W}_{RF}^t, \mathbf{W}_{BB}^i, \tilde{\mathbf{X}}, \tilde{\mathbf{X}}_{eq}, \tilde{\mathbf{W}}_{RF}, \tilde{\mathbf{F}}_{RF}, \mathbf{q}^t, \mathbf{q}_{eq}^i, \mathcal{F}(\cdot), \mathcal{F}_{eq}(\cdot), \mathcal{G}(\cdot), \mathcal{G}_{eq}(\cdot), \forall t, i\}$ 。特别地， $\{\tilde{\mathbf{X}}, \tilde{\mathbf{W}}_{RF}, \tilde{\mathbf{F}}_{RF}\}$ 和 $\tilde{\mathbf{X}}_{eq}$ 分别表示用于估计原始信道矩阵 \mathbf{H} 和低维等效信道矩阵 \mathbf{H}_{eq} 的导频， $\tilde{\mathbf{x}}_l$ 和 $\tilde{\mathbf{x}}_{eq,l}$ 分别表示 $\tilde{\mathbf{X}}$ 和 $\tilde{\mathbf{X}}_{eq}$ 的第 l 列， \mathbf{q}^t 表示在第 t 帧的原始信道矩阵 \mathbf{H}^t 压缩后的反馈比特， \mathbf{q}_{eq}^i 表示在第 i 时隙的等效信道矩阵 \mathbf{H}_{eq}^i 压缩后的反馈比特。此外， \mathbf{F}_{RF}^t 和 \mathbf{W}_{RF}^t 分别表示第 t 帧的发送端和接收端模拟预编码矩阵， \mathbf{F}_{BB}^i 和 \mathbf{W}_{BB}^i 分别表示第 i 时隙的发送端和接收端数字预编码矩阵， $\mathcal{F}(\cdot)$ 和 $\mathcal{F}_{eq}(\cdot)$ 分别表示针对 \mathbf{H} 和 \mathbf{H}_{eq} 的信道反馈网络， $\mathcal{G}(\cdot)$ 和 $\mathcal{G}_{eq}(\cdot)$ 分别表示模拟和数字预编码网络。上述变量和网络的优化和设计旨在最小化该通信系统的误码率 $P_e(\cdot)$ ，而误码率是一个关于变量 $\{\mathbf{F}_{RF}^t, \mathbf{F}_{BB}^i, \mathbf{W}_{RF}^t, \mathbf{W}_{BB}^i\}$ 的复杂非线性

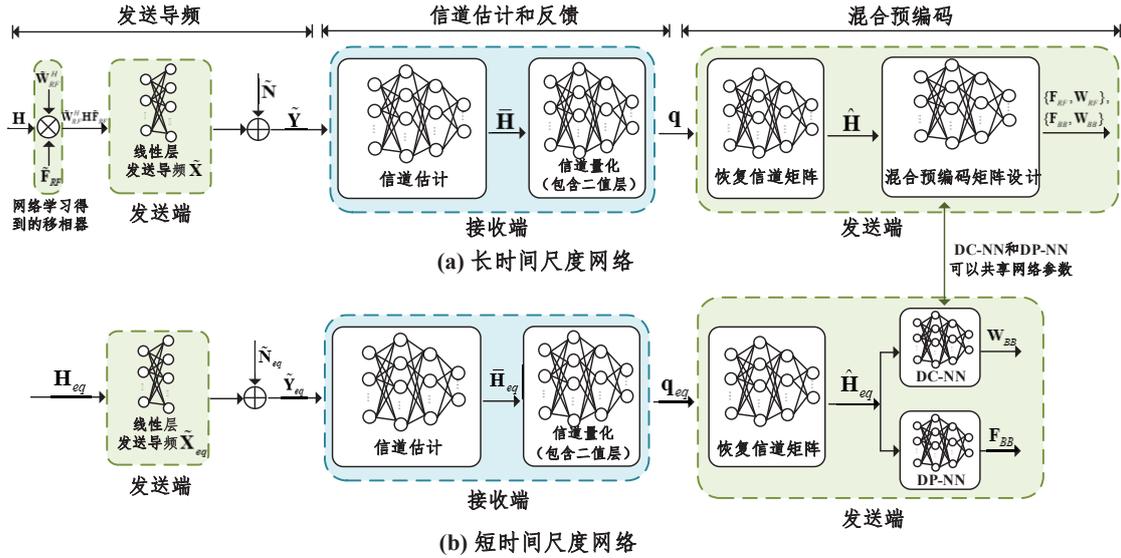


图 5.4 双时间尺度网络结构：(a)长时间尺度网络；(b)短时间尺度网络。

函数。约束(5.7b)-(5.7c)和(5.7d)-(5.7e)分别表示针对模拟导频和模拟预编码矩阵的恒模约束。约束(5.7f)-(5.7g)和(5.7h)分别表示针对数字导频和数字预编码矩阵的功率约束。为表述清楚起见，后续章节略去脚标 t 和 i 。

5.3 混合预编码端到端深度学习网络架构

本章节提出基于端到端深度学习的联合训练框架，如图5.2所示。基于该框架，设计了双时间尺度网络结构，用于求解问题(5.7)。该结构由长时间尺度网络和短时间尺度网络组成。

5.3.1 端到端深度学习框架

旨在提出一个基于深度学习的框架，以提供较低的误码率性能，同时具有较短的导频序列 L 和较少的信道反馈比特数 B 。具体来说，DNN被用于模拟频分双工毫米波MIMO系统，包含如下几个阶段：导频训练，信道估计，信道反馈，混合预编码和数据传输。将在章节5.3.2，章节5.3.3和章节5.3.4中分别描述如何在长时间尺度网络中联合训练导频矩阵 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{W}}_{RF}, \tilde{\mathbf{F}}_{RF}\}$ ，信道反馈网络 $\mathcal{F}(\cdot)$ 和混合预编码网络 $\mathcal{G}(\cdot)$ 。图5.4(a)描述了本章提出的深度学习框架的流程，包含了上述过程，其中混合预编码网络的具体架构和数据传输的具体过程请见图5.5。如图5.4(a)所示，接收端网络的输入时接收到的导频信号 $\tilde{\mathbf{Y}}$ ，输出是信道反馈矢量 \mathbf{q} 。在发送端，训练导频矩阵 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{W}}_{RF}, \tilde{\mathbf{F}}_{RF}\}$ ，网络的输入是反馈比特 \mathbf{q} ，输出是混合预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}\}$ 。对比图5.2所示的传统通信系统，可以

看出每一个模块都被一个DNN替代。这些网络可以进行端到端联合训练，将比特交叉熵（Bit-wise Cross Entropy, BCE）作为损失函数，具体请见章节5.3.4。

为进一步降低通信传输开销，以及由于传输时延导致的信道失配误差，提出了由长时间尺度网络和短时间尺度网络组成的双时间尺度网络。具体而言，长时间尺度网络直接采用图5.4(a)的DNN框架，且长时间尺度网络在每帧的最后一个时隙运行一次。图5.4(b)所示的短时间尺度网络是基于图5.4(a)修改的，且在每个时隙运行一次。具体而言，短时间尺度训练导频矩阵 $\tilde{\mathbf{X}}_{eq}$ ，信道反馈网络 $\mathcal{F}_{eq}(\cdot)$ ，数字预编码网络 $\mathcal{G}_{eq}(\cdot)$ 是联合设计的，具体请分别见章节5.3.2，章节5.3.3，和章节5.3.4的短时间尺度网络部分。如图5.4(b)所示，在接收端的DNN输入是接收到的导频信号 $\tilde{\mathbf{Y}}_{eq}$ ，输出是低维等效信道对应的反馈矢量 \mathbf{q}_{eq} 。在发送端，训练导频矩阵 $\tilde{\mathbf{X}}_{eq}$ ，DNN的输入是低维等效信道对应的反馈矢量 \mathbf{q}_{eq} ，输出是数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ 。该系统的通信过程总结在章节5.3.5中。接下来详细描述图5.4中每个模块的细节。

5.3.2 导频设计和信道估计

在导频训练阶段，接收端需要在每帧的前 $T_s - 1$ 个时隙估计 \mathbf{H}_{eq} ，在每帧的最后一个时隙估计 \mathbf{H} 。

5.3.2.1 基于长时间尺度网络的导频设计

为了估计 \mathbf{H} ，发送端发送数字导频矩阵 $\tilde{\mathbf{X}} \in \mathbb{C}^{N_t^{RF} \times L}$ ，以及模拟预编码矩阵（在信道估计阶段称为模拟导频矩阵） $\tilde{\mathbf{F}}_{RF} \in \mathbb{C}^{N_t \times N_t^{RF}}$ ，其中 L 表示导频长度。接着，收到的导频矩阵经过模拟接收矩阵 $\tilde{\mathbf{W}}_{RF} \in \mathbb{C}^{N_r \times N_r^{RF}}$ ，表示为

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{W}}_{RF}^H \mathbf{H} \tilde{\mathbf{F}}_{RF} \tilde{\mathbf{X}} + \tilde{\mathbf{N}}, \quad (5.8)$$

其中 $\tilde{\mathbf{N}} = \tilde{\mathbf{W}}_{RF}^H \mathbf{N}$ ， $\mathbf{N} \in \mathbb{C}^{N_r \times L}$ 表示AWGN矩阵。

为建模导频训练过程，设计适应当前信道统计特性的导频矩阵，以更高效准确地估计 \mathbf{H} ，将网络的输入和输出分别设为 \mathbf{H} 和 $\tilde{\mathbf{Y}}$ ，同时将导频矩阵 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ 设为DNN的训练参数。相比传统方法采用高斯导频 $\tilde{\mathbf{X}}$ ，或者选取DFT矩阵的列组成 $\{\tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ ^[158]，训练得到的导频 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ 可以更高效准确地估计信道，因为经过DNN训练得到的导频矩阵更适应当前的信道统计特性。为确保模拟导频矩阵 $\tilde{\mathbf{F}}_{RF}$ 和 $\tilde{\mathbf{W}}_{RF}$ 满足恒模约束，除了将对应的矩阵元素设为训练参数，在DNN训练完成后令其除以自身的模值，即 $\frac{[\tilde{\mathbf{F}}_{RF}]_{ij}}{|\tilde{\mathbf{F}}_{RF}|_{ij}}$ 。

此外, 为保证导频矩阵 $\tilde{\mathbf{X}}$ 满足功率约束(5.7f), 通过放缩 $\tilde{\mathbf{X}}$ 使得 $\|\tilde{\mathbf{x}}_l\|^2 = P, \forall l$, 其中 $\tilde{\mathbf{x}}_l$ ($\tilde{\mathbf{X}}$ 的第 l 列)表示第 l 次传输的导频。

注意到可以改变 $\tilde{\mathbf{F}}_{RF}$ 和 $\tilde{\mathbf{W}}_{RF}$ 用于 L 次不同的导频传输, 信道估计过程可以表示为

$$\tilde{\mathbf{y}}_l = \tilde{\mathbf{W}}_{RF,l}^H \mathbf{H} \tilde{\mathbf{F}}_{RF,l} \tilde{\mathbf{x}}_l + \tilde{\mathbf{n}}_l, \quad (5.9)$$

其中 $\tilde{\mathbf{F}}_{RF,l} \in \mathbb{C}^{N_t \times N_t^{RF}}$ 和 $\tilde{\mathbf{W}}_{RF,l} \in \mathbb{C}^{N_r \times N_r^{RF}}, l = 1, 2, \dots, L$, 分别表示导频训练阶段第 l 次导频传输的发端和收端模拟预编码矩阵。此外, $\tilde{\mathbf{x}}_l$ 和 $\tilde{\mathbf{y}}_l$ 分别表示发送导频矩阵 $\tilde{\mathbf{X}}$ 的第 l 列和接收导频矩阵 $\tilde{\mathbf{Y}}$ 的第 l 列。这里 $\tilde{\mathbf{F}}_{RF,l}$ 和 $\tilde{\mathbf{W}}_{RF,l}$ 被设为DNN的训练参数。因此, 每次导频传输的发送和接收端模拟预编码矩阵不同。这可以激活多个毫米波MIMO信道的不同角度模式, 从而实现更好的系统性能。

5.3.2.2 基于短时间尺度网络的导频设计

为了估计 \mathbf{H}_{eq} , 发送端发送导频矩阵 $\tilde{\mathbf{X}}_{eq} \in \mathbb{C}^{N_t^{RF} \times L}$. 接收端收到对应的导频信号矩阵为

$$\tilde{\mathbf{Y}}_{eq} = \mathbf{H}_{eq} \tilde{\mathbf{X}}_{eq} + \tilde{\mathbf{N}}_{eq}, \quad (5.10)$$

其中 $\mathbf{H}_{eq} = \mathbf{W}_{RF}^H \mathbf{H} \mathbf{F}_{RF}$, $\tilde{\mathbf{N}}_{eq} = \mathbf{W}_{RF}^H \mathbf{N}$, 以及 $\mathbf{N} \in \mathbb{C}^{N_r \times L}$ 为高斯噪声矩阵。

为了建模导频训练过程, 设计适应当前信道统计特性的导频信号矩阵, 以更高效准确地估计 \mathbf{H}_{eq} , 将网络的输入和输出分别设为 \mathbf{H}_{eq} 和 $\tilde{\mathbf{Y}}_{eq}$, 训练参数为 $\tilde{\mathbf{X}}_{eq}$ 。在信道估计阶段, 和长时间尺度网络不同, 短时间尺度网络的模拟预编码矩阵 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 不是通过训练得到, 而是直接采用上一帧在数据传输阶段使用的 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 。即短时间尺度网络估计的是 \mathbf{H}_{eq} , 而 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 为低维等效信道 \mathbf{H}_{eq} 的一部分, 该部分内容会在章节5.3.4详细阐述。类似于长时间尺度网络, 通过缩放 $\tilde{\mathbf{X}}_{eq}$ 满足功率约束。

不同于传统的信道估计方案, 本章提出的方案中, 接收端不需要知道发送端传输的原始导频矩阵 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{X}}_{eq}, \tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ 。这是因为导频矩阵被设成网络的训练参数, 以适配当前信道的统计特性。由于提出的网络架构可以从接收到的导频信号矩阵 $\{\tilde{\mathbf{Y}}, \tilde{\mathbf{Y}}_{eq}\}$ 中提取有用信息, 接收端可以仅通过网络提取 $\{\tilde{\mathbf{Y}}, \tilde{\mathbf{Y}}_{eq}\}$ 中的信息, 来估计信道矩阵 $\{\mathbf{H}, \mathbf{H}_{eq}\}$ 。

5.3.3 信道反馈

在每帧的前 $T_s - 1$ 个时隙, 接收端反馈 \mathbf{H}_{eq} 。在每帧的最后一个时隙, 接收端反馈 \mathbf{H} 。

5.3.3.1 基于长时间尺度网络的信道反馈

在每帧的最后一个时隙，接收端根据收到的 $\tilde{\mathbf{Y}}$ 估计 \mathbf{H} ，并从中提取有用信息，将其压缩量化为 B 比特反馈给发送端。该过程可用一个 R 层的FNN实现，即信道反馈矢量可表示为

$$\mathbf{q} = \text{sgn}(\mathbf{W}_R \sigma_{R-1}(\cdots \sigma_1(\mathbf{W}_1 \tilde{\mathbf{y}} + \mathbf{b}_1) \cdots) + \mathbf{b}_R). \quad (5.11)$$

其中 $\mathbf{q} \in \{\pm 1\}^B$ ， $\tilde{\mathbf{y}} \triangleq \text{Vec}(\tilde{\mathbf{Y}})$ 表示将矩阵 $\tilde{\mathbf{Y}}$ 矢量化。网络的输入为矢量 $\tilde{\mathbf{y}}$ 实部、虚部分开的表示，即 $\tilde{\mathbf{y}} \triangleq [\Re(\tilde{\mathbf{y}}^T), \Im(\tilde{\mathbf{y}}^T)]^T$ 。 $\{\mathbf{W}_r, \mathbf{b}_r\}_{r=1}^R$ 表示网络的训练参数， σ_r 表示第 r 层的激活函数。符号函数 $\text{sgn}(\cdot)$ 为最后一层的激活函数，用于产生信道反馈矢量 \mathbf{q} 。最后一层为二值层，即 \mathbf{q} 的每个元素取值为0或1。

5.3.3.2 基于短时间尺度网络的信道反馈

关于 \mathbf{H}_{eq} 的信道反馈流程可以类似设计。在每帧的前 $T_s - 1$ 个时隙，接收端根据收到的 $\tilde{\mathbf{Y}}_{eq}$ 估计 \mathbf{H}_{eq} ，并从中提取有用信息，将其压缩量化为 B_{eq} 比特反馈给发送端。该过程可用一个 R_{eq} 层的FNN实现，即信道反馈矢量可表示为

$$\mathbf{q}_{eq} = \text{sgn}(\mathbf{W}_{R_{eq}} \sigma_{R_{eq}-1}(\cdots \sigma_1(\mathbf{W}_1 \tilde{\mathbf{y}}_{eq} + \mathbf{b}_1) \cdots) + \mathbf{b}_{R_{eq}}), \quad (5.12)$$

其中 $\tilde{\mathbf{y}}_{eq} \triangleq \text{Vec}(\tilde{\mathbf{Y}}_{eq})$ 表示矩阵 $\tilde{\mathbf{Y}}_{eq}$ 的矢量化结果，网络的输入为矢量 $\tilde{\mathbf{y}}_{eq}$ 实部、虚部分开的表示，即 $\tilde{\mathbf{y}}_{eq} \triangleq [\Re(\tilde{\mathbf{y}}_{eq}^T), \Im(\tilde{\mathbf{y}}_{eq}^T)]^T$ 。 $\{\mathbf{W}_r, \mathbf{b}_r\}_{r=1}^{R_{eq}}$ 表示网络的训练参数。符号函数 $\text{sgn}(\cdot)$ 为最后一层的激活函数，用于产生信道反馈矢量 $\mathbf{q}_{eq} \in \{\pm 1\}^{B_{eq}}$ （ \mathbf{q}_{eq} 的每个元素取值都为0或1）。注意到 \mathbf{q}_{eq} 的维度要远小于 \mathbf{q} 的维度，即 $B_{eq} < B$ ，因为 \mathbf{H}_{eq} 的维度远小于 \mathbf{H} 的维度。因此，可设计层数和参数量更少的FNN来输出信道反馈矢量 \mathbf{q}_{eq} 。

5.3.4 混合预编码设计

在每一帧的前 $T_s - 1$ 个时隙，根据 \mathbf{q}_{eq} ，使用短时间尺度网络来更新数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ 。在每帧的最后一个时隙，根据 \mathbf{q} ，使用长时间尺度网络来更新数字和模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}\}$ 。这里假定收发端之间信道反馈矢量 \mathbf{q}_{eq} 和 \mathbf{q} 的传输是无误的。

5.3.4.1 基于长时间尺度网络的收发端混合预编码设计

在每帧的最后一个时隙，发送端收到 \mathbf{q} 用于恢复 $\hat{\mathbf{H}}$ 。随后，发送端根据恢复出的 $\hat{\mathbf{H}}$ ，用DNN设计混合预编码矩阵。如图5.5所示，该混合预编码网络包含5个子网络，发送

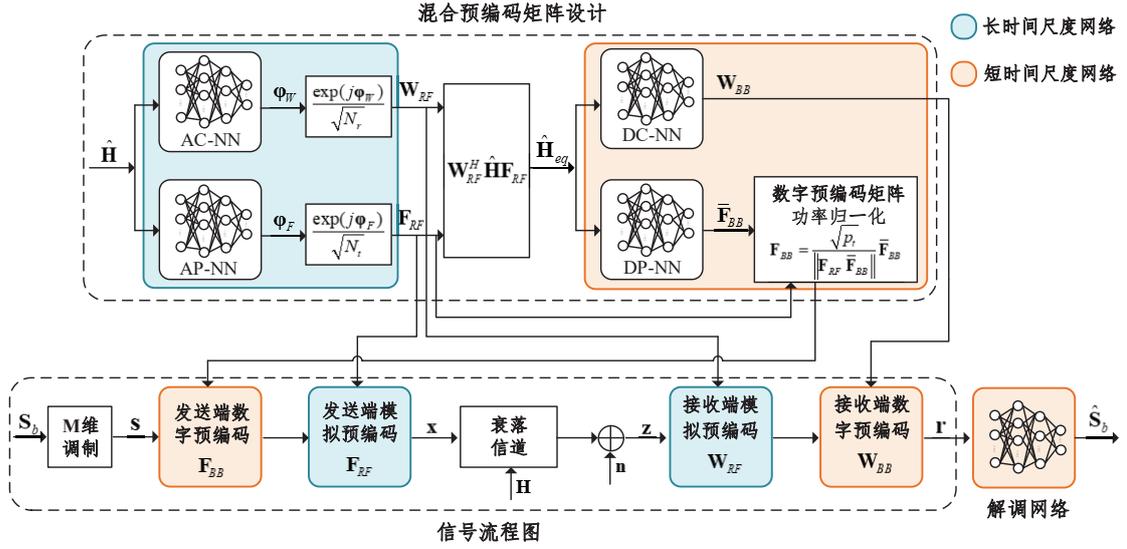


图 5.5 用于设计混合预编码的网络架构及数据传输阶段的信号流程图

端模拟预编码网络 (Analog Precoder-Neural Network, AP-NN)，发送端数字预编码网络 (Digital Precoder-Neural Network, DP-NN)，接收端模拟预编码网络 (Analog Combiner-Neural Network, AC-NN)，接收端数字预编码网络 (Digital Combiner-Neural Network, DC-NN)，以及解调网络 (Demodulator-Neural Network, DE-NN)。具体而言，先将恢复出的 $\hat{\mathbf{H}}$ 的实部、虚部分开存储，转换为一个 $2N_t N_r \times 1$ 维的实数矩阵。将其输入 AP-NN 和 AC-NN，分别输出发送端和接收端模拟预编码器的相位 $\varphi_F \in \mathbb{R}^{N_t N_t^{RF} \times 1}$ 和 $\varphi_W \in \mathbb{R}^{N_r N_r^{RF} \times 1}$ ，可得到满足恒模约束的矢量

$$\bar{\mathbf{f}}_{RF} = \frac{1}{\sqrt{N_t}} e^{j\varphi_F}, \quad \bar{\mathbf{w}}_{RF} = \frac{1}{\sqrt{N_r}} e^{j\varphi_W}. \quad (5.13)$$

随后，模拟预编码矩阵 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 可以写成

$$\mathbf{F}_{RF} = \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{f}}_{RF}), \quad \mathbf{W}_{RF} = \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{w}}_{RF}), \quad (5.14)$$

其中 $\mathcal{J}_{v \rightarrow m}$ 表示将矢量转换为矩阵的操作。 $\hat{\mathbf{H}}_{eq}$ 可根据估计出的 $\hat{\mathbf{H}}$ 和 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 计算得到

$$\hat{\mathbf{H}}_{eq} = \mathbf{W}_{RF}^H \hat{\mathbf{H}} \mathbf{F}_{RF}. \quad (5.15)$$

类似地，将等效信道矩阵 $\hat{\mathbf{H}}_{eq} \in \mathbb{C}^{N_r^{RF} \times N_t^{RF}}$ 的实部、虚部分别进行存储，并将其转换为一个维度为 $2N_t^{RF} N_r^{RF} \times 1$ 的实数矩阵，输入 DP-NN 和 DC-NN，分别输出发送端和接收端的数字预编码矩阵 $\{\bar{\mathbf{w}}_{BB, re}, \bar{\mathbf{w}}_{BB, im}\}$ 和 $\{\bar{\mathbf{f}}_{BB, re}, \bar{\mathbf{f}}_{BB, im}\}$ (实部、虚部分开存储)。数字预编码矩阵可以表示为

$$\begin{aligned} \mathbf{W}_{BB} &= \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{w}}_{BB, re} + j\bar{\mathbf{w}}_{BB, im}), \\ \bar{\mathbf{F}}_{BB} &= \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{f}}_{BB, re} + j\bar{\mathbf{f}}_{BB, im}). \end{aligned} \quad (5.16)$$

将(5.16)中的 $\bar{\mathbf{F}}_{BB}$ 和(5.14)中的 \mathbf{F}_{RF} 经过下述功率归一化, 可得到使功率约束(5.7h)满足的数字预编码矩阵

$$\mathbf{F}_{BB} = \frac{\sqrt{P_T}}{\|\mathbf{F}_{RF}\bar{\mathbf{F}}_{BB}\|} \bar{\mathbf{F}}_{BB}. \quad (5.17)$$

5.3.4.2 基于短时间尺度网络的收发端数字预编码设计

在每帧的前 $T_s - 1$ 个时隙, 发送端收到信道反馈矢量 \mathbf{q}_{eq} , 用于恢复 $\hat{\mathbf{H}}_{eq}$ 。随后, 根据恢复出的 $\hat{\mathbf{H}}_{eq}$, 发送端使用一个DNN设计 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$, 此时 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 固定不变, 即采用上一帧最后一个时隙, 长时间尺度网络计算得到的 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 。如图5.5所示, 短时间尺度网络包含DP-NN和CP-NN, 结合(5.16)-(5.17), 可分别用于产生 \mathbf{F}_{BB} 和 \mathbf{W}_{BB} 。

5.3.4.3 信号流程图

离线训练的目的在于根据输入的训练样本 $\{\mathbf{H}, \mathbf{n}, \mathbf{S}_b\}$ 和标签 \mathbf{S}_b , 训练DNN的参数 Θ 。信道和噪声一般服从某一分布, 根据该分布产生信道和噪声样本, 用于网络训练。

信号流如图5.5所示, 整个过程模拟了发送信号 \mathbf{S}_b 经过发送端混合预编码 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}\}$ 、信道衰落 \mathbf{H} 、噪声 \mathbf{n} 、接收端混合预编码 $\{\mathbf{W}_{RF}, \mathbf{W}_{BB}\}$, 接收端恢复出该信号 $\hat{\mathbf{S}}_b$ 的过程。其中混合预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}\}$ 是根据步(5.13)-(5.17)产生的。信号模型(5.3)的执行是通过将训练样本 $\{\mathbf{H}, \mathbf{n}, \mathbf{S}_b\}$ 输入网络, 产生混合预编码矩阵, 最终得到接收信号 \mathbf{r} 。将接收信号 \mathbf{r} 的实部、虚部分开, 转换为一个实值矢量, 输入解调网络, 产生恢复信号 $\hat{\mathbf{S}}_b$ 。通过最小化 \mathbf{S}_b 和 $\hat{\mathbf{S}}_b$ 的BCE, 网络的训练参数 Θ 用SGD进行迭代更新。在预测阶段, 信号流中的各模块采用DNN输出的混合预编码矩阵。

5.3.4.4 损失函数和误码率

使用BCE作为DNN训练阶段的损失函数:

$$\mathcal{L}(\Theta) = -\frac{1}{|\mathcal{B}|} \sum_{\mathbf{S}_b \in \mathcal{B}} \sum_{i=1}^{N_s} \sum_{j=1}^{\log_2 M} \left([\mathbf{S}_b]_{i,j} \ln([\hat{\mathbf{S}}_b(\Theta)]_{i,j}) + (1 - [\mathbf{S}_b]_{i,j}) \ln(1 - [\hat{\mathbf{S}}_b(\Theta)]_{i,j}) \right), \quad (5.18)$$

其中 \mathcal{B} 表示训练数据集, \mathbf{S}_b 表示传输符号矩阵, 由维度为 $N_s \times \log_2 M$ 的0-1二元比特组组成。 $[\hat{\mathbf{S}}_b(\Theta)]_{i,j} \in [0, 1]$ 代表恢复出的符号矩阵, 表示对应每个元素传输的比特为1的概率。由于 $[\hat{\mathbf{S}}_b(\Theta)]$ 是网络的输出, 可以表示为网络的训练参数 Θ 的函数。注意到最大化BCE可等价于最大化可达通信速率, 同时可以训练得到标准的比特解码器^[220]。

回顾(5.7)中的优化问题，训练集的误码率可以表示为：

$$P_e(\Theta) \triangleq P_e(\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}) = \frac{\sum_{\mathbf{S}_b \in \mathcal{B}} \sum_{i=1}^{N_s} \sum_{j=1}^{\log_2 M} |[\mathbf{S}_b]_{i,j} - [\hat{\mathbf{S}}_{b,1hot}(\Theta)]_{i,j}|}{|\mathcal{B}| N_s \log_2 M}, \quad (5.19)$$

其中当 $[\hat{\mathbf{S}}_b(\Theta)]_{i,j} < 0.5$ 时， $[\hat{\mathbf{S}}_{b,1hot}(\Theta)]_{i,j} = 0$ ，否则 $[\hat{\mathbf{S}}_{b,1hot}(\Theta)]_{i,j} = 1$ 。

5.3.4.5 深度神经网络的部署

提出的混合预编码网络包含四个子网络：AP-NN、DP-NN、AC-NN和DC-NN，四个网络进行联合训练。对于这四个子网络有两种不同的部署方式：（1）四个子网络都部署在发送端；（2）AP-NN和DP-NN部署在发送端，而AC-NN和DC-NN部署在接收端。如果采用第一种部署方式，发送端需要使用四个子网络，设计四个发送端和接收端的混合预编码矩阵。随后，发送端需要将接收端的两个混合预编码矩阵传输到接收端。如果采用第二种部署方式，发送端预编码矩阵通过AP-NN和DP-NN在发送端设计，接收端预编码矩阵通过AC-NN和DC-NN在接收端设计。第一种方式不需要接收端有高效的计算资源，而第二种方式不需要发送端反馈设计好的接收端预编码矩阵到接收端。

5.3.5 双时间尺度通信过程

图5.6描述了基于双时间尺度混合预编码的FDD毫米波MIMO系统的通信流程。在每帧的前 $T_s - 1$ 个时隙，发送端发送导频信号 $\tilde{\mathbf{X}}_{eq}$ ，接收端根据接收到的 $\tilde{\mathbf{Y}}_{eq}$ 估计 \mathbf{H}_{eq} ，进行压缩量化得到 \mathbf{q}_{eq} ，并将其反馈至发送端。接着，发送端根据 \mathbf{q}_{eq} 恢复出 $\hat{\mathbf{H}}_{eq}$ 并设计数字预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ ，同时保持模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 不变。最后，按照如图5.5所示的信号流传输数据。在每帧的最后一个时隙，发送端首先发送数字导频和模拟导频 $\{\tilde{\mathbf{X}}, \tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ 至接收端，接收端根据收到的 $\tilde{\mathbf{X}}$ 恢复出 \mathbf{H} ，进行压缩量化得到 \mathbf{q} ，并将其反馈至发送端。随后，发送端根据 \mathbf{q} 恢复出 $\hat{\mathbf{H}}$ ，输入DNN得到混合预编码矩阵 $\{\mathbf{F}_{BB}, \mathbf{F}_{RF}, \mathbf{W}_{BB}, \mathbf{W}_{RF}\}$ 。最后，传输实际要发送的数据 \mathbf{s} 。由于 \mathbf{H}_{eq} 的维度远小于 \mathbf{H} ， \mathbf{q}_{eq} 的维度远小于 \mathbf{q} 的维度。

上述提出的工作在FDD系统中的双时间尺度网络可经过如下修改直接拓展到TDD系统：（1）移除信道反馈模块；（2）修改导频训练阶段，让接收端发送导频。在TDD系统中，上行信道和下行信道满足互易性。为获取用于设计混合预编码的下行信道矩阵，发送端首先根据接收端发送的导频估计上行信道。随后，下行信道矩阵可根据上下行信道的互易性获得。因此，TDD系统不需要信道反馈。

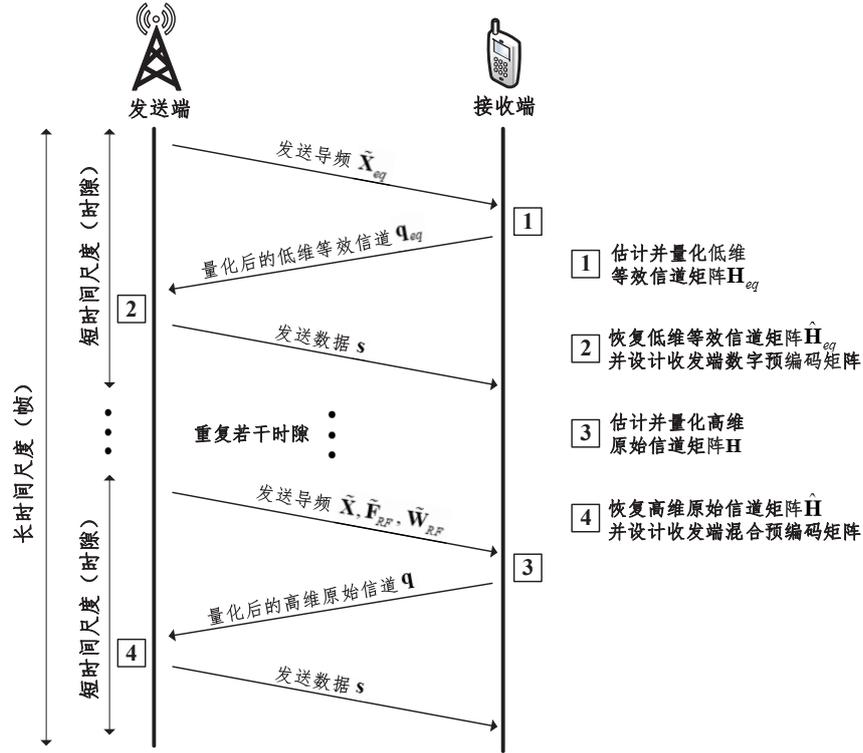


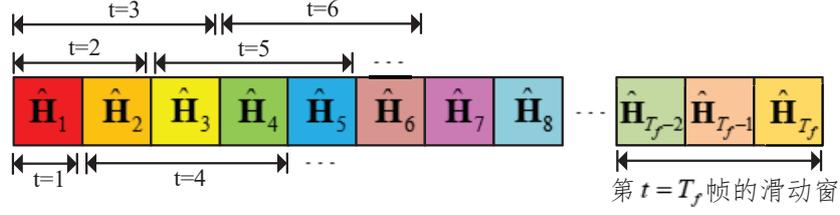
图 5.6 大规模MIMO混合预编码系统的收发端通信过程

5.4 端到端深度学习网络结构和系统部署

本章节描述了双时间尺度网络的结构、训练方法和系统部署，以及模拟预编码矩阵的优化方法。

5.4.1 双时间尺度网络结构

为了使网络结构具有一般性和普适性，本章采用具有全连接结构的DNN，且网络最后一层采用Sigmoid函数作为激活函数，其他层采用ReLU函数作为激活函数。本章主要提出双时间尺度深度学习框架，网络可以替换成其他具有更复杂结构和激活函数的网络。具体而言，一个4层的DNN被用于信道估计，每层的神经元数目 $[l_1, l_2, l_3, l_4] = [N_r L, 256, 128, N_r N_t]$ 。信道量化和恢复模块，分别采用一个3层的网络 $[l_1, l_2, l_3] = [N_r N_t, 128, B]$ 和一个4层的网络 $[l_1, l_2, l_3, l_4] = [B, 256, 128, N_r N_t]$ 。对于发送端和接收端模拟预编码网络，AC-NN采用 $[l_1, l_2, l_3, l_4] = [N_r N_t, 256, 128, N_r N_r^{RF}]$ ，AP-NN采用 $[l_1, l_2, l_3, l_4] = [N_r N_t, 256, 128, N_t N_t^{RF}]$ 。因为数字预编码矩阵的维度远小于模拟预编码矩阵的维度，采用神经元数目更少的4层神经网络，其中DC-NN采用 $[l_1, l_2, l_3, l_4] = [N_r^{RF} N_t^{RF}, 64, 32, N_r^{RF} N_s]$ ，DP-NN采用 $[l_1, l_2, l_3, l_4] = [N_r^{RF} N_t^{RF}, 64, 32, N_t^{RF} N_s]$ 。此外，DE-


 图 5.7 大小为 $D = 3$ 的高维原始信道矩阵缓存滑动窗

NN 采用 4 层网络结构，其中 $[l_1, l_2, l_3, l_4] = [2N_s,$

$64, 32, N_s \log_2 M]$ 。另外，采用批归一化技术和残差网络 ResNet 中的残差块解决梯度爆炸和梯度消失问题，从而提升系统性能。

5.4.2 模拟预编码矩阵的优化

注意到长时间尺度变量（模拟预编码矩阵 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} ）根据当前的 CSI 统计特性进行设计。因此， \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 的优化需根据一定数量的样本 \mathbf{H} 。然而，每帧只能获取一个样本 \mathbf{H} 。为解决该矛盾，设计了滑动平均的方式，以充分利用样本 $\mathbf{H}^{[110]}$ ，即对 DNN 当前帧的输出和前一帧的模拟编码器相位做加权平均

$$\varphi_F^{t+1} = (1 - \gamma_t)\varphi_F^t + \gamma_t\bar{\varphi}_F^t, \quad \varphi_W^{t+1} = (1 - \gamma_t)\varphi_W^t + \gamma_t\bar{\varphi}_W^t \quad (5.20)$$

其中 φ_F^t 和 $\bar{\varphi}_F^t$ 分别表示当前 t 时刻的发送端模拟编码器相位和上一帧 AP-NN 的输出， φ_W^t 和 $\bar{\varphi}_W^t$ 分别表示当前 t 时刻的接收端模拟编码器相位和上一帧 AC-NN 的输出， $\{\gamma_t, t = 1, 2, \dots, T_f\}$ 为用于滑动平均的步长序列，需满足以下条件^[110]： $\lim_{t \rightarrow \infty} \gamma_t = 0$ ， $\sum_t \gamma_t = \infty$ ，and $\sum_t (\gamma_t)^2 < \infty$ 。

此外，为了使 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 更好地适配 CSI 统计特性并充分利用样本 \mathbf{H} ，设计了缓存容量为 D 的滑动窗 \mathcal{D} ，存储之前若干帧恢复出来的高维原始信道样本 $\hat{\mathbf{H}}$ ，如图 5.7 所示。由此，在第 t 帧 AC-NN 和 AP-NN 的输入，为从第 $t - D + 1$ 帧到当前第 t 帧所有恢复出的高维原始信道样本 \mathbf{H} ，即 $[\hat{\mathbf{H}}_{t-D+1}, \hat{\mathbf{H}}_{t-D+2}, \dots, \hat{\mathbf{H}}_t]$ 。

5.4.3 训练方法

5.4.3.1 带有离散输出层的网络训练

由于 DNN 的最后一层采用 $\text{sgn}(\cdot)$ 作为激活函数，输出为 0 或 1，导数几乎处处为 0，在原点不可导。因此，不能用传统的反向传播算法训练该 DNN。本章设计一种新颖的

的训练方式，核心思想是在梯度反向传播的过程中，用光滑且处处可导的函数来替代 $\text{sgn}(\cdot)$ ，这里采用Sigmoid-Adjusted ST函数 $2\text{sigm}(x) - 1$ 来逼近 $\text{sgn}(x)$ 作为激活函数，其中 $\text{sigm}(x) = 1/(1 + \exp(-x))$ 为Sigmoid函数。为使训练过程更稳定，采用逐步逼近的方法^[265]，即随着训练过程的进行，逐渐增加替代函数的斜率，使其逐步逼近符号函数。这样做的好处在于，一开始该DNN的训练过程就和一般DNN一样，比较容易快速地达到一定效果，随后再增加替代函数的斜率，可避免训练过程的数值不稳定，使训练收敛得更快更稳定。本章采用的替代函数为

$$2\text{sigm}(\alpha^{(i)}x) - 1 = \frac{2}{(1 + \exp(-\alpha^{(i)}x))} - 1, \quad (5.21)$$

其中 $\alpha^{(i)}$ 为第 i 个训练周期的参数，需满足条件： $\alpha^{(i)} \geq \alpha^{(i-1)}$ 。

5.4.3.2 双时间尺度网络的训练和预测过程

网络采用双时间尺度的训练方案，即长时间尺度网络和短时间尺度网络交替迭代地训练。具体而言，在每一帧的前 $T_s - 1$ 个时隙，训练短时间尺度网络，输入为训练样本 $\{\mathbf{H}, \mathbf{F}_{RF}, \mathbf{W}_{RF}, \mathbf{n}, \mathbf{S}_b\}$ ，其中 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 根据长时间尺度网络及其对应输入 \mathbf{H} 计算得到。在每帧的最后一个时隙，训练长时间尺度网络，输入为 $\{\mathbf{H}, \mathbf{n}, \mathbf{S}_b\}$ 。即长时间尺度网络在每帧训练一次，短时间尺度网络每一时隙训练一次，两者交替训练直至收敛。在预测阶段同理，每帧的前 $T_s - 1$ 个时隙，发送导频 $\tilde{\mathbf{X}}_{eq}$ ，短时间尺度网络根据公式(5.16)-(5.17)，输出 $\{\mathbf{F}_{BB}, \mathbf{W}_{BB}\}$ 。在每帧的最后一个时隙，发送导频 $\tilde{\mathbf{X}}$ ，长时间尺度网络根据公式(5.13)-(5.17)和(5.20)，输出数字和模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}\}$ 。

5.5 性能分析和系统扩展

本章节提出提升网络泛化性能的方法，并分析了不同方案的通信传输开销。此外，将该单载波双时间尺度混合预编码系统拓展到了多载波OFDM系统。

5.5.1 网络的泛化能力

DNN的泛化能力可以分为两类。第一类，参数的改变仅导致网络输入统计特性的改变，比如信道参数 N_{cl} 、信噪比、噪声方差 σ_n^2 等。对于这类参数，网络泛化能力的提升可以通过在训练数据集中加入不同参数的样本。以信噪比为例，网络的训练数据集应包含多组不同信噪比的样本。第二类，参数的改变会改变网络某些层的输入和输出维度，比如信

道反馈比特数 B ，导频长度 L ，天线数 (N_t, N_r) 和射频链路数 (N_t^{RF}, N_r^{RF}) 。训练一个适配不同上述参数的网络具有一定的挑战性。接下来，阐述如何提升网络针对第二种参数的泛化能力。

5.5.1.1 针对不同信道反馈比特数 B 的泛化能力

为了训练一个可以工作在不同信道反馈比特数 B 的具有较强泛化能力的网络^[219]，提出一种新颖的两步骤训练方式。第一步，训练一个修改版本的网络，该网络中信道反馈网络的输出 \mathbf{q} 并不是二值的，而是经过 P 个神经元经过Tanh激活函数产生的 -1 到 1 范围内的实数。该修改过的网络用于获得训练导频和信道估计网络。完成该网络的训练之后，得到Tanh激活函数输出的经验概率分布函数（Probability Distribution Function, PDF），接着基于Lloyd-Max算法，根据不同量化比特数 Q 设计最优标量量化器。第二步，固定接收端网络的训练参数，训练发送端混合预编码网络的参数。针对不同的反馈比特数 B ，对 P 个神经元的输出值采用不同的量化精度。具体而言，发送端收到接收端反馈的 Q 比特量化的 P 个信号，发送端的网络将这些经过量化的信号映射成混合预编码矩阵。总的信道反馈比特数 $B = P \times Q$ ，因此通过改变每个信号的量化精度 Q ，就可以实现将一个训练好的网络泛化到不同的信道反馈比特数 B 。

5.5.1.2 针对不同导频长度 L 的泛化能力

训练在较大的参数 L_0 下的网络，可以直接泛化到具有更小参数 L_1 的场景。以短时间尺度网络为例，接收导频信号矩阵 $\tilde{\mathbf{Y}}_{eq} \in \mathbb{C}^{N_r^{RF} \times L_0}$ 的最后 $L_0 - L_1$ 列被设为 $\mathbf{0}$ ，或者等价地，发送导频矩阵 $\tilde{\mathbf{X}}_{eq} \in \mathbb{C}^{N_t^{RF} \times L_0}$ 的最后 $L_0 - L_1$ 列，以及噪声矩阵 $\tilde{\mathbf{N}}_{eq} \in \mathbb{C}^{N_r^{RF} \times L_0}$ 的最后 $L_0 - L_1$ 列被设为 $\mathbf{0}$ ：

$$\left[\tilde{\mathbf{Y}}'_{eq}, \mathbf{0} \right] = \mathbf{H}_{eq} \left[\tilde{\mathbf{X}}'_{eq}, \mathbf{0} \right] + \left[\tilde{\mathbf{N}}'_{eq}, \mathbf{0} \right], \quad (5.22)$$

其中 $\tilde{\mathbf{X}}'_{eq} \in \mathbb{C}^{N_t^{RF} \times L_1}$ ， $\tilde{\mathbf{Y}}'_{eq} \in \mathbb{C}^{N_r^{RF} \times L_1}$ ，和 $\tilde{\mathbf{N}}'_{eq} \in \mathbb{C}^{N_r^{RF} \times L_1}$ 。长时间尺度网络对参数 L 的泛化能力可以类似分析。

5.5.1.3 针对不同射频链路数 N_t^{RF} 和 N_r^{RF} 的泛化能力

根据系统参数 $(N_{t_0}^{RF}, N_{r_0}^{RF})$ 训练的网络可直接用于具有更小系统参数的场景 $(N_{t_1}^{RF}, N_{r_1}^{RF})$ ，而不需要重新训练一个网络。在导频训练阶段，为了确保具有更小系统参数的场景的网络输入，即 $\mathbf{H}'_{eq} \in \mathbb{C}^{N_{r_1}^{RF} \times N_{t_1}^{RF}}$ ，和已在较大系统参数下训练好的网络是一样的，

即 $\mathbf{H}_{eq} \in \mathbb{C}^{N_{r_0}^{RF} \times N_{t_0}^{RF}}$, 采用填零的方法在 \mathbf{H}'_{eq} 加上 $N_{t_0}^{RF} - N_{t_1}^{RF}$ 全零列和 $N_{r_0}^{RF} - N_{r_1}^{RF}$ 全零行:

$$\begin{bmatrix} \tilde{\mathbf{Y}}'_{eq} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{H}'_{eq} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}'_{eq} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{N}}'_{eq} \\ \mathbf{0} \end{bmatrix}, \quad (5.23)$$

其中 $\tilde{\mathbf{Y}}'_{eq} \in \mathbb{C}^{N_{r_1}^{RF} \times L}$ 表示接收到的导频信号矩阵, $\tilde{\mathbf{N}}'_{eq} \in \mathbb{C}^{N_{r_1}^{RF} \times L}$ 表示噪声矩阵, $\tilde{\mathbf{X}}'_{eq} \in \mathbb{C}^{N_{t_1}^{RF} \times L}$ 表示从具有更大系统参数的网络 $(N_{t_0}^{RF}, N_{r_0}^{RF})$ 发送的导频矩阵。注意到只有训练导频矩阵的前 $N_{t_1}^{RF}$ 行, 即 $\tilde{\mathbf{X}}'_{eq}$ 实际被用于更小的系统 $(N_{t_1}^{RF}, N_{r_1}^{RF})$ 。

此外, 采用针对混合预编码矩阵的填零。对于模拟预编码矩阵 $\mathbf{F}_{RF} \in \mathbb{C}^{N_t \times N_t^{RF}}$, 将其最后 $N_{t_0}^{RF} - N_{t_1}^{RF}$ 列设为 $\mathbf{0}$, 即 $[\mathbf{F}'_{RF}, \mathbf{0}]$, 其中 \mathbf{F}'_{RF} 表示具有较大参数 $(N_{t_0}^{RF}, N_{r_0}^{RF})$ 系统的 AP-NN 输出的模拟预编码矩阵的前 $N_{t_1}^{RF}$ 列。其他预编码矩阵可以类似处理。关于长时间尺度网络针对系统参数 N_t 和 N_r 的泛化性可以类似分析。

5.5.2 通信传输开销

本章节分析提出的基于双时间尺度网络混合预编码通信系统和其他相应系统的通信传输开销, 即信道反馈开销。考虑一个包含 T_f 个帧的超帧, 每一个帧包含 T_s 个时隙, 分析结果如下所示。

- 传统单时间尺度系统: 令 B_c 表示信道矩阵 $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ 的每个元素量化的比特数, 因此一个超帧的传输开销为 $Q_{cs} = T_f T_s B_c N_r N_t$ 比特。
- 传统双时间尺度系统: 在每帧的前 $T_s - 1$ 个时隙, 接收端反馈低维等效信道矩阵 $\mathbf{H}_{eq} \in \mathbb{C}^{N_{r_1}^{RF} \times N_{t_1}^{RF}}$ 的量化比特。在每帧的最后一个时隙, 接收端反馈高维原始信道矩阵 \mathbf{H} 的量化比特。因此, 一个超帧的传输开销为 $Q_{ct} = T_f B_c ((T_s - 1) N_{r_1}^{RF} N_{t_1}^{RF} + N_r N_t)$ 比特。
- 单时间尺度端到端深度学习系统: 该系统在每个时隙采用长时间尺度网络更新收发端数字和模拟预编码矩阵。因此, 每个时隙的传输开销为向量 \mathbf{q} 的维度, 即 B , 其中 $B \ll B_c N_r N_t$, 一个超帧的传输开销为 $Q_s = T_f T_s B$ 比特。
- 双时间尺度端到端深度学习系统: 令 B_t 表示向量 \mathbf{q}_{eq} 的维度, 有 $B_t < B$ 。因此, 该系统在一个超帧的传输开销为 $Q_t = T_f ((T_s - 1) B_t + B)$ 比特。

根据上述结果, 可以看出相比其他通信系统, 本章提出的基于端到端深度学习的双时间尺度混合预编码通信系统显著降低了通信传输开销。

5.5.3 扩展到OFDM系统

本章节介绍如何将提出的单载波双时间尺度网络混合预编码毫米波通信系统，拓展到多载波OFDM系统。主要需要考虑如下几点^[113]：

- 在OFDM系统中，收发端数字预编码矩阵可以针对每个子载波独立设计，但不同的子载波需要共享相同的收发端模拟预编码矩阵。
- 为保证可扩展性和泛化性，DNN的结构需要和单载波系统保持一致，即每层的神经元数和层数需要一致。
- 由于OFDM系统中子载波的数目通常很大，设计的系统期望网络训练和网络预测的时间不会随着子载波数的增加而增加¹。

信号传输模型和子载波相关，检测到的第 k 个子载波上对应的信号为

$$\mathbf{r}[k] = \mathbf{W}_{BB}^H[k] \mathbf{W}_{RF}^H \mathbf{H}[k] \mathbf{F}_{RF} \mathbf{F}_{BB}[k] \mathbf{s} + \mathbf{W}_{BB}^H[k] \mathbf{W}_{RF}^H \mathbf{n}, \quad (5.24)$$

其中 $k \in \mathcal{K} \triangleq \{1, 2, \dots, K\}$ 表示OFDM系统的子载波。

5.5.3.1 基于长时间尺度网络的导频训练和信道反馈

为估计 $\mathbf{H}[k]$ ，发送端发送导频信号矩阵 $\tilde{\mathbf{X}}[k] \in \mathbb{C}^{N_t^{RF} \times L}$ ，经过发送端模拟预编码矩阵 $\tilde{\mathbf{F}}_{RF} \in \mathbb{C}^{N_t \times N_t^{RF}}$ 的处理。随后，接收到的信号经过 $\tilde{\mathbf{W}}_{RF} \in \mathbb{C}^{N_r \times N_r^{RF}}$ 的处理得到

$$\tilde{\mathbf{Y}}[k] = \tilde{\mathbf{W}}_{RF}^H \mathbf{H}[k] \tilde{\mathbf{F}}_{RF} \tilde{\mathbf{X}}[k] + \tilde{\mathbf{N}}[k], \quad (5.25)$$

其中 $\tilde{\mathbf{N}}[k] = \tilde{\mathbf{W}}_{RF}^H \mathbf{N}[k]$ ， $\mathbf{N}[k] \in \mathbb{C}^{N_r \times L}$ 表示AWGN矩阵。为了建模导频训练过程并得到用于估计 $\mathbf{H}[k]$ 的最优导频，该子网络的输入和输出分别为 $\mathbf{H}[k]$ 和 $\tilde{\mathbf{Y}}[k]$ ，网络训练参数为 $\{\tilde{\mathbf{X}}[k], \tilde{\mathbf{F}}_{RF}, \tilde{\mathbf{W}}_{RF}\}$ 。

接收端根据收到的 $\tilde{\mathbf{Y}}[k]$ 估计 $\mathbf{H}[k]$ 。随后，接收端提取有用信息并将其压缩量化为 $B[k]$ 比特，反馈至发送端。该过程可用一个 R 层的FNN实现，信道反馈矢量可表示为

$$\mathbf{q}[k] = \text{sgn}(\mathbf{W}_R \sigma_{R-1}(\dots \sigma_1(\mathbf{W}_1 \tilde{\mathbf{y}}[k] + \mathbf{b}_1) \dots) + \mathbf{b}_R), \quad (5.26)$$

其中 $\mathbf{q}[k] \in \{\pm 1\}^{B[k]}$ ， $\tilde{\mathbf{y}}[k] \triangleq \text{Vec}(\tilde{\mathbf{Y}}[k])$ 表示矩阵 $\tilde{\mathbf{Y}}[k]$ 的矢量化，网络的输入是 $\tilde{\mathbf{y}}[k]$ 的实部和虚部分开表示的矢量，即 $\tilde{\mathbf{y}}[k] \triangleq [\Re(\tilde{\mathbf{y}}[k]^T), \Im(\tilde{\mathbf{y}}[k]^T)]^T$ 。

¹由于OFDM系统中子载波的数目通常很大，如果网络训练和网络预测的时间会随着子载波数的增加而增加，则会大大影响系统的运行效率，导致较高的系统时延。

5.5.3.2 基于短时间尺度网络的导频训练和信道反馈

为估计 $\mathbf{H}_{eq}[k]$ ，发送端发送导频信号矩阵 $\tilde{\mathbf{X}}_{eq}[k] \in \mathbb{C}^{N_t^{RF} \times L}$ ，接收端收到的导频信号矩阵

$$\tilde{\mathbf{Y}}_{eq}[k] = \mathbf{H}_{eq}[k]\tilde{\mathbf{X}}_{eq}[k] + \tilde{\mathbf{N}}_{eq}[k], \quad (5.27)$$

其中 $\mathbf{H}_{eq}[k] = \mathbf{W}_{RF}^H \mathbf{H}[k] \mathbf{F}_{RF}$ ， $\tilde{\mathbf{N}}_{eq}[k] = \mathbf{W}_{RF}^H \mathbf{N}[k]$ ， $\mathbf{N}[k] \in \mathbb{C}^{N_r \times L}$ 表示AWGN矩阵。为了建模用于估计 $\mathbf{H}_{eq}[k]$ 的导频训练过程，该子网络的输入和输出分别为 $\tilde{\mathbf{X}}_{eq}[k]$ 和 $\tilde{\mathbf{Y}}_{eq}[k]$ ，网络训练参数为 $\tilde{\mathbf{X}}_{eq}[k]$ 。

接收端根据接收到的 $\tilde{\mathbf{Y}}_{eq}[k]$ 估计 $\mathbf{H}_{eq}[k]$ 。随后，接收端从中提取有用的信息并将其压缩量化成 $B_{eq}[k]$ 比特，反馈至发送端。该过程可用一个 R_{eq} 层的FNN实现，接收端的信道反馈矢量可表示为

$$\mathbf{q}_{eq}[k] = \text{sgn}(\mathbf{W}_{R_{eq}} \sigma_{R_{eq}-1}(\cdots \sigma_1(\mathbf{W}_1 \tilde{\mathbf{y}}_{eq}[k] + \mathbf{b}_1) \cdots) + \mathbf{b}_{R_{eq}}), \quad (5.28)$$

其中 $\mathbf{q}_{eq}[k] \in \{\pm 1\}^{B_{eq}[k]}$ ， $\tilde{\mathbf{y}}_{eq}[k] \triangleq \text{Vec}(\tilde{\mathbf{Y}}_{eq}[k])$ 表示矩阵 $\tilde{\mathbf{Y}}_{eq}[k]$ 的矢量化，网络的输入是 $\tilde{\mathbf{y}}_{eq}[k]$ 的实部和虚部分开表示的矢量，即 $\tilde{\mathbf{y}}_{eq}[k] \triangleq [\Re(\tilde{\mathbf{y}}_{eq}^T[k]), \Im(\tilde{\mathbf{y}}_{eq}^T[k])]^T$ 。

5.5.3.3 基于长时间尺度网络的混合预编码设计

发送端收到信道反馈矢量 $\mathbf{q}[k]$ 用于恢复 $\hat{\mathbf{H}}[k]$ 。接着，发送端采用一个DNN，根据恢复的 $\hat{\mathbf{H}}[k]$ 设计混合预编码矩阵。注意到模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 由不同子载波共享，输入 $\hat{\mathbf{H}}$ 到子网络AP-NN和AC-NN，分别输出 $\varphi_F \in \mathbb{R}^{N_t N_t^{RF} \times 1}$ 和 $\varphi_W \in \mathbb{R}^{N_r N_r^{RF} \times 1}$ 。有两种关于 $\hat{\mathbf{H}}$ 的选择方法：(1) 任意选择一个估计得到的某一个子载波的信道矩阵，比如第 l 个子载波的信道矩阵 $\hat{\mathbf{H}}[l]$ ；(2) 所有子载波信道矩阵的平均，即 $\sum_{k=1}^K \hat{\mathbf{H}}[k]$ 。得到 φ_F 和 φ_W 之后， \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 根据公式(5.13)和(5.14)产生。随后， \mathbf{F}_{RF} ， \mathbf{W}_{RF} 和 $\hat{\mathbf{H}}[k]$ 可以用于产生低维等效信道矩阵

$$\hat{\mathbf{H}}_{eq}[k] = \mathbf{W}_{RF}^H \hat{\mathbf{H}}[k] \mathbf{F}_{RF}. \quad (5.29)$$

随后， $\hat{\mathbf{H}}_{eq}[k] \in \mathbb{C}^{N_r^{RF} \times N_t^{RF}}$ 输入子网络DP-NN和DC-NN，两个子网络分别输出 $\{\bar{\mathbf{w}}_{BB,re}[k], \bar{\mathbf{w}}_{BB,im}[k]\}$ 和 $\{\bar{\mathbf{f}}_{BB,re}[k], \bar{\mathbf{f}}_{BB,im}[k]\}$ 。接着， $\mathbf{W}_{BB}[k]$ 和 $\mathbf{F}_{BB}[k]$ 通过下式计算得到：

$$\begin{aligned} \mathbf{W}_{BB}[k] &= \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{w}}_{BB,re}[k] + j\bar{\mathbf{w}}_{BB,im}[k]), \\ \bar{\mathbf{F}}_{BB}[k] &= \mathcal{J}_{v \rightarrow m}(\bar{\mathbf{f}}_{BB,re}[k] + j\bar{\mathbf{f}}_{BB,im}[k]). \end{aligned} \quad (5.30)$$

最后， $\bar{\mathbf{F}}_{BB}[k]$ 通过功率归一化以满足功率约束。

5.5.3.4 基于短时间尺度网络的数字预编码设计

发送端收到信道反馈矢量 $\mathbf{q}_{eq}[k]$ 用于恢复 $\hat{\mathbf{H}}_{eq}[k]$ 。随后，发送端采用一个DNN，根据恢复的 $\hat{\mathbf{H}}_{eq}[k]$ 设计数字预编码矩阵 $\{\mathbf{F}_{BB}[k], \mathbf{W}_{BB}[k]\}$ ，而模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ 保持固定不变。

5.5.3.5 训练过程

单载波系统中网络的训练样本为 $\{\mathbf{H}, \mathbf{n}, \mathbf{S}_b\}$ ，相应地，多载波系统中网络的训练样本应修改为 $\{\hat{\mathbf{H}}, \mathbf{H}[k], \mathbf{n}, \mathbf{S}_b\}$ 。注意到导频训练子网络和信道反馈子网络的训练方法，与单载波系统是相同的。接下来介绍如何训练长时间尺度混合预编码子网络，短时间尺度数字预编码子网络的训练方法类似。将每个训练样本输入混合预编码子网络， $\hat{\mathbf{H}}$ 被用于输入AP-NN和AC-NN子网络分别产生 \mathbf{F}_{RF} 和 \mathbf{W}_{RF} 。接着， \mathbf{F}_{RF} ， \mathbf{W}_{RF} 和 $\mathbf{H}[k]$ 被用于产生第 k 个子载波的低维等效信道矩阵 $\mathbf{H}_{eq}[k]$ 。将 $\mathbf{H}_{eq}[k]$ 输入DP-NN和DC-NN子网络，分别产生 $\mathbf{F}_{BB}[k]$ 和 $\mathbf{W}_{BB}[k]$ 。另一方面， $\mathbf{H}[k]$ 输入信号流图作为衰落信道，该训练样本被用于模拟第 k 个子载波上信号的传输。通过最小化 \mathbf{S}_b 和 $\hat{\mathbf{S}}_b$ 之间的BCE损失函数执行上述网络的端到端训练过程。通过上述训练方式，可得到匹配所有子载波对应信道的模拟预编码矩阵 $\{\mathbf{F}_{RF}, \mathbf{W}_{RF}\}$ ，以及针对每一个子载波的数字预编码矩阵 $\{\mathbf{F}_{BB}[k], \mathbf{W}_{BB}[k]\}$ 。即可以直接采用前述章节提出的单载波网络的结构，而不需要增加额外的模块。

5.6 仿真结果

本节通过仿真验证了基于DNN的端到端联合信道估计、信道反馈和混合预编码方案的有效性。首先，说明了仿真参数设置、仿真的算法，研究了提出的网络在训练阶段的收敛性能。随后，将该方案与基线算法进行系统的BER性能比较，并分析了网络的泛化能力。

5.6.1 仿真参数设置

系统的参数配置描述如下。设置发送端的发送天线数 $N_t = 64$ 和射频链路数 $N_t^{RF} = 8$ ，接收端的接收天线数 $N_r = 32$ 和射频链路数 $N_r^{RF} = 4$ ，以及数据流数为 $N_s = 4$ 。将训练数据集的信噪比设为 $\text{SNR} = 10 \text{ dB}$ ，导频长度设为 $L = 28$ ，信道反馈比特数设为 $B = 64$ 。滑动窗口的大小设为 $D = 3$ ，一帧内包含的时隙数为 $T_s = 10$ 。使用深度学习库“Pytorch”来实现提出的端到端DNN架构。使用“Adam”优化器作为网络训练算法，批次大小为128，动态学习率 η 从 10^{-2} 逐渐衰减到 10^{-5} 。为了加快收敛速度，网络的每一层都采用批归一化

和随机丢弃的技术进行处理。在训练阶段，缓慢地增加Sigmoid函数(5.21)的模拟退火参数 $\alpha^i = 2 + 0.2i$ ，其中 i 表示训练周期，每个训练周期包括200个训练样本批次。

本章采用广泛使用的窄带毫米波信道模型来产生信道样本^[109]，它由 N_{cl} 个簇和 N_{ray} 条传播路径组成，对应的信道矩阵可以表示为

$$\mathbf{H} = \sqrt{\frac{N_t N_r}{N_{cl} N_{ray}}} \sum_{i=1}^{N_{cl}} \sum_{l=1}^{N_{ray}} \alpha_{il} \mathbf{a}_r(\phi_{il}^r) \mathbf{a}_t^H(\phi_{il}^t), \quad (5.31)$$

其中 $\alpha_{il} \sim \mathcal{CN}(0, \sigma_\alpha^2)$ 表示第 i 簇中第 j 条路径的复增益系数，服从均值为0，方差为 σ_α^2 的复高斯分布。 ϕ_{il}^r 和 ϕ_{il}^t 分别表示第 i 簇中第 j 条路径的AoA和AoD， $\mathbf{a}_r(\phi_{il}^r)$ 和 $\mathbf{a}_t(\phi_{il}^t)$ 分别表示接收和发送的阵列响应矢量。对于一个包含 N 根天线、方位角为 ϕ 的均匀线性阵列，其信道响应向量可以表示为

$$\mathbf{a}(\phi) = \frac{1}{\sqrt{N}} [1, e^{-j2\pi \frac{d}{\lambda} \sin(\phi)}, \dots, e^{-j2\pi \frac{d}{\lambda} (N-1) \sin(\phi)}]^T, \quad (5.32)$$

其中 d 和 λ 分别表示相邻天线之间的距离和载波波长。设簇的数目为 $N_{cl} = 3$ ，每个簇中包含的信道路径的数目为 $N_{ray} = 4$ 。复增益系数服从均值为0、方差为1的复高斯分布，即 $\alpha_{il} \sim \mathcal{CN}(0, 1)$ 。AoA和AoD均服从 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 之间的均匀分布，即 $\phi_{il}^r \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ 和 $\phi_{il}^t \sim \mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ 。

为使信道模型(5.31)更符合系统的实际情况，引入由信道时延 τ 和信道的最大多普勒频移 f_d 导致的信道失配 $\exp(j2\pi f_d \tau \cos(\phi_{il}^r))$ 。因此，实际信道矩阵表示为^[112]

$$\mathbf{H} = \sqrt{\frac{N_t N_r}{N_{cl} N_{ray}}} \sum_{i=1}^{N_{cl}} \sum_{l=1}^{N_{ray}} \alpha_{il} \mathbf{a}_r(\phi_{il}^r) \mathbf{a}_t^H(\phi_{il}^t) \times \exp(j2\pi f_d \tau \cos(\phi_{il}^r)). \quad (5.33)$$

其中信道时延 τ 与信道反馈比特数成正比^[112]，即 $\frac{\tau_t}{\tau_s} = \frac{Q_t}{Q_s}$ ，其中 τ_t 和 τ_s 分别表示双时间尺度算法和单时间尺度算法的信道时延。在仿真中设 $\tau_s = 1 \text{ ms}$ 。

对于基线算法，考虑两种混合预编码算法：（1）文献[109]中提出的迭代优化算法（Optimization Algorithm, OPT）；（2）文献[266]中提出的信道匹配算法（Channel Matching Algorithm, CMA）。采用最优的Lloyd-Max算法对信道参数进行量化反馈，并采用OMP算法^[157]对信道矩阵进行估计。具体而言，比较了以下算法的性能：

- **T-DNN**: 提出的双时间尺度网络，包括长时间尺度网络和短时间尺度网络，并考虑了由信道时延导致的信道失配。
- **S-DNN**: 提出的单时间尺度网络，即仅使用长时间尺度网络在每个时隙同时更新模拟和数字预编码矩阵。

- **OPT(CMA)**: 完整的信道矩阵 \mathbf{H} 在发送端是完美已知的, 并使用OPT (或CMA) 算法来设计混合预编码矩阵。
- **OPT(CMA)/Lloyd**: 接收端对信道参数完美已知, 信道参数包括: 复增益系数的实部和虚部 $\{\Re(\alpha_{il}), \Im(\alpha_{il}), \forall i, l\}$, AoA和AoD参数 $\{\phi_{il}^r, \phi_{il}^t, \forall i, l\}$ 。旨在通过一个 B 比特有限容量反馈链路, 传输由Lloyd-Max算法量化的信道参数, 将这些参数无差错地传输到发送端^[219]。值得注意的是, 每个信道参数都被分配 $\frac{B}{4N_{cl}N_{ray}}$ 个量化比特。通过采用(5.31)中的信道模型, 发送端可以重构信道矩阵 $\hat{\mathbf{H}}$, 随后采用OPT (或CMA) 算法进行混合预编码。
- **OPT(CMA)/OMP**: 接收端基于广泛使用的OMP算法在导频训练阶段估计信道矩阵, 随后在具有无限容量的链路上将估计的信道矩阵无差错地反馈给发送端。发送端根据该恢复出来的信道矩阵, 使用OPT (CMA) 算法设计混合预编码矩阵。
- **OPT(CMA)/Delay**: 考虑由信道时延导致的信道失配(5.33)场景下的OPT (CMA) 方案。
- **OPT(CMA)/Lloyd/Delay**: 考虑由信道时延导致的信道失配(5.33)场景下的OPT (CMA) /Lloyd方案。
- **OPT(CMA)/OMP/Delay**: 考虑由信道时延导致的信道失配(5.33)场景下的OPT (CMA) /OMP方案。

5.6.2 网络的收敛性能

图5.8(a)展示了在不同学习率下的损失函数收敛曲线, 即BCE的收敛性能。可以看到, 较小的学习率可以实现更好的性能, 而较大的学习率具有更快的收敛速度。值得注意的是, 从 10^{-2} 逐渐下降到 10^{-5} 的动态学习率, 收敛速度较快且实现了较好的性能。图5.8(b)展示了不同训练批次大小下BCE损失函数的收敛性能。可以观察到, 批次越大, 则收敛性能越稳定。

5.6.3 系统的BER性能

图5.9(a)描述了在不考虑实际信道失配的单时间尺度场景下, 所提出的单时间尺度网络(S-DNN)和基线算法在不同SNR下的BER性能。可以看到, 所有算法实现的BER都随着SNR的增加而有所下降。所提出的S-DNN实现的BER性能优于基线算法OPT/Lloyd、

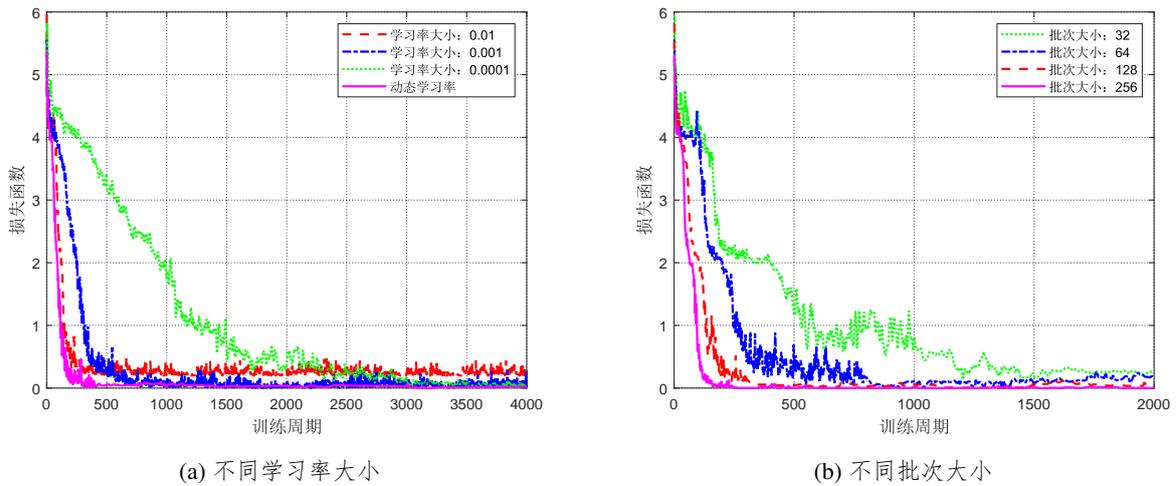


图 5.8 双时间尺度网络的收敛性能

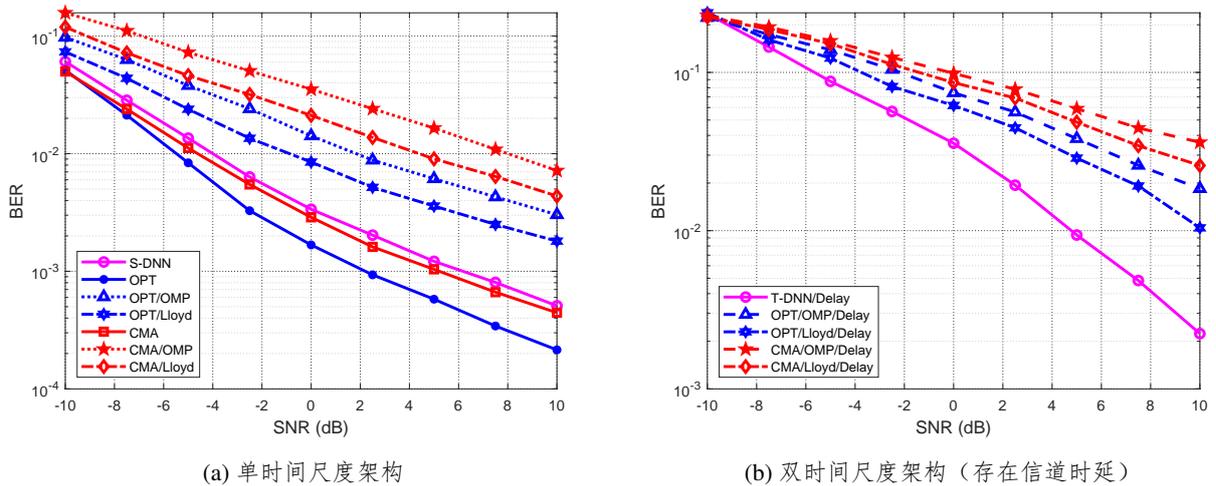


图 5.9 系统BER性能随SNR的变化

OPT/OMP、CMA/Lloyd和CMA/OMP，其性能差距随着SNR的增加而扩大。因此，可以得出结论：提出的联合训练S-DNN架构显著优于信道估计、信道反馈和混合预编码等各系统模块独立设计的传统方案。此外，由于OPT是一种可以保证找到局部最优解的迭代优化算法，而CMA是一种启发式算法，因此OPT的性能比CMA更好但计算复杂度更高。此外，S-DNN实现的BER性能可以逼近具有完美信道状态信息和无限反馈精度的OPT所实现的BER性能下界。由此可见，S-DNN是一个可用于联合设计导频训练、信道估计、信道反馈和混合预编码等不同系统模块的有效框架。

图5.9(b)描述了在考虑实际信道失配的场景下，所提出的双时间尺度网络（T-DNN）和基线算法的BER性能。由图可见，当考虑由信道时延导致的实际信道失配时，T-DNN的

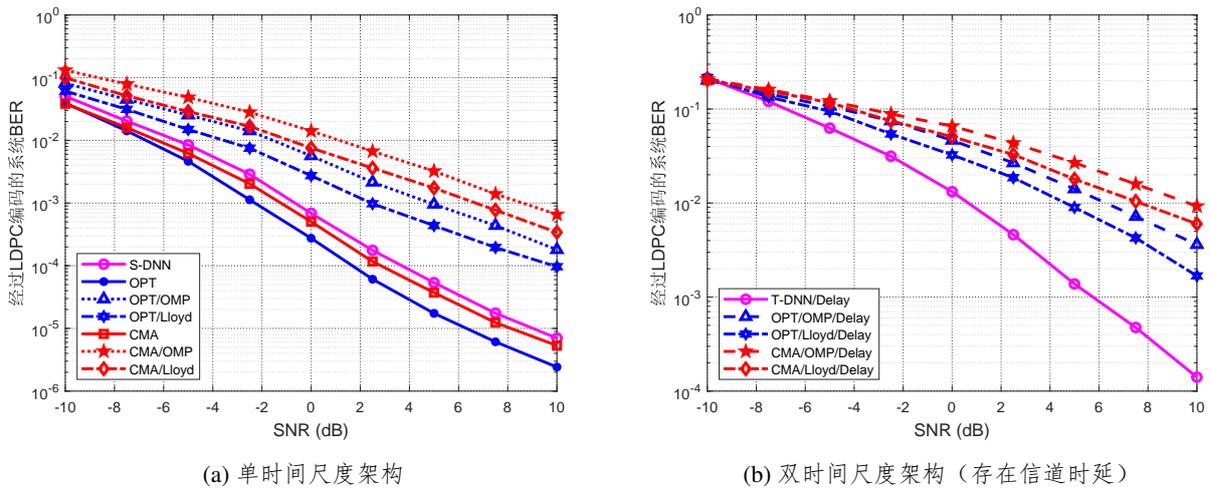


图 5.10 经过LDPC编码的系统BER性能随SNR的变化

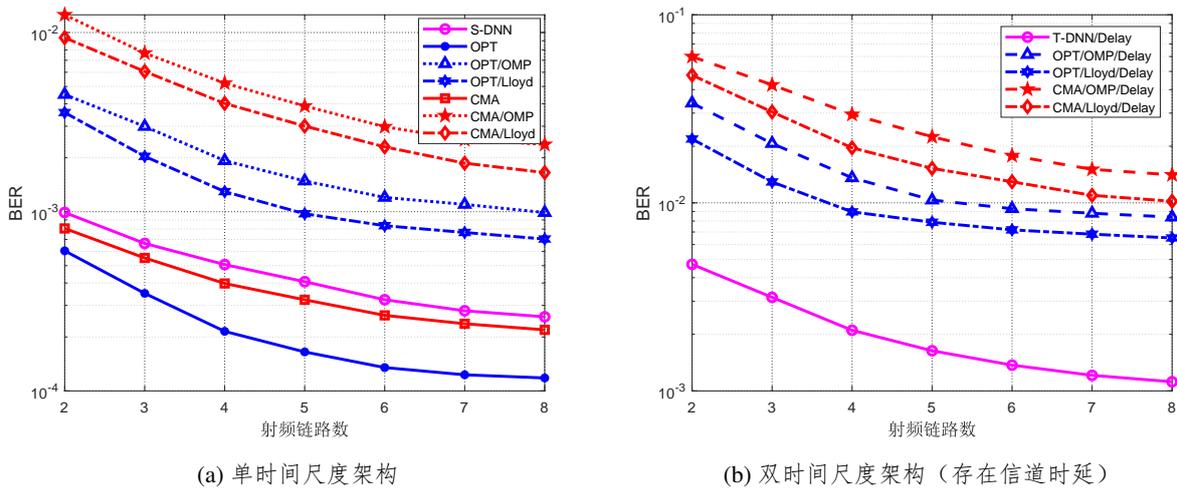


图 5.11 系统BER性能随射频链路数 N_{RF} 的变化

性能明显优于其他基线方案。这验证了T-DNN在降低信令传输开销以及信道状态信息失配等方面的有效性。图5.10展示了所提出的S-DNN和T-DNN，以及基线方案在编码系统中的BER性能，这里采用了低密度奇偶校验（Low-Density Parity-Check, LDPC）编码。与图5.9所示的未编码系统的BER性能相比，在高SNR的场景下，使用LDPC编码显著降低了BER，提升了系统性能。

图5.11(a)展示了单时间尺度场景下，所提出的S-DNN和基线算法实现的系统BER性能随射频链路数 N_{RF} 的变化。可以看到所有算法的BER随 N_{RF} 的增加而下降。所提出的S-DNN实现的性能显著优于OPT/Lloyd、OPT/OMP、CMA/Lloyd和CMA/OMP算法，并且性能差距随着射频链路数目的增加而扩大。此外，S-DNN实现的BER性能逼近具有完

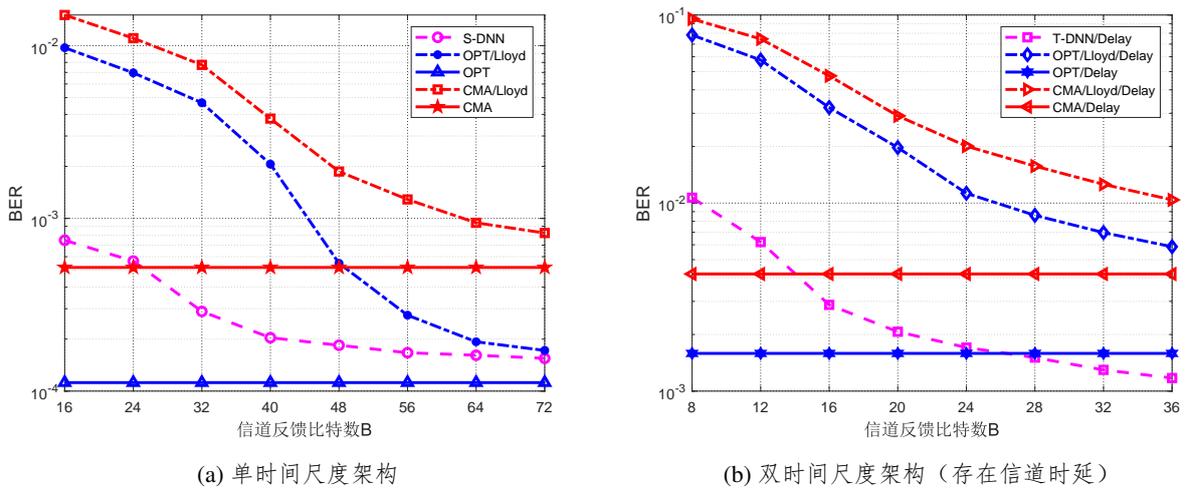


图 5.12 系统BER性能随信道反馈比特数 B 的变化

美信息状态信息和无限反馈精度的OPT所实现的BER下界。图5.11(b)展示了在考虑实际信道失配的场景下，所提出的T-DNN和基线算法实现的系统BER性能随 N_{RF} 的变化。由图可见，T-DNN实现了最优的BER性能，远超越其他基线算法，这进一步证明了T-DNN网络架构的优越性。

图5.12(a)描述了单时间尺度场景下，所提出的S-DNN和基线算法实现的系统BER性能随信道反馈比特数 B 的变化。由图可知，在信道反馈比特数 B 相同的情况下，S-DNN的性能显著优于OPT/Lloyd和CMA/Lloyd，并且当 B 较小时，性能增益更为显著。这表明当反馈比特数 B 有限时，例如 $B = 16$ ，端到端网络联合设计方案S-DNN显著优于信道估计、信道反馈和混合预编码分别单独设计的传统方案。值得注意的是，S-DNN在 $B = 40$ 时实现的性能逼近OPT/Lloyd在 $B = 64$ 时的性能，这表明S-DNN可以显著降低信道反馈比特数。此外，S-DNN可以超越在无限反馈精度下的CMA实现的性能，且S-DNN可以较少的信道反馈比特数逼近OPT实现的BER性能下界。图5.12(b)描述了在考虑实际信道失配的场景下，所提出的T-DNN和基线算法实现的系统BER性能随 B 的变化。在 B 相同的情况下，T-DNN性能显著优于OPT/Lloyd/Delay和CMA/Lloyd/Delay的性能。此外，T-DNN在 $B = 12$ 时，实现了与OPT/Lloyd/Delay在 $B = 36$ 时几乎相同的BER性能，这验证了所提出的T-DNN可以显著降低信道反馈比特数，同时保持较好的系统性能。此外，当 $B > 28$ 时，T-DNN超越了具有无限反馈精度的OPT所实现的BER性能，这是因为T-DNN对由时延导致的信道失配具有更强的鲁棒性。与S-DNN相比，T-DNN可以显著减少信道反馈比特数，因为接收端只需将低维等效信道的状态信息 \mathbf{H}_{eq} 反馈给发送端。

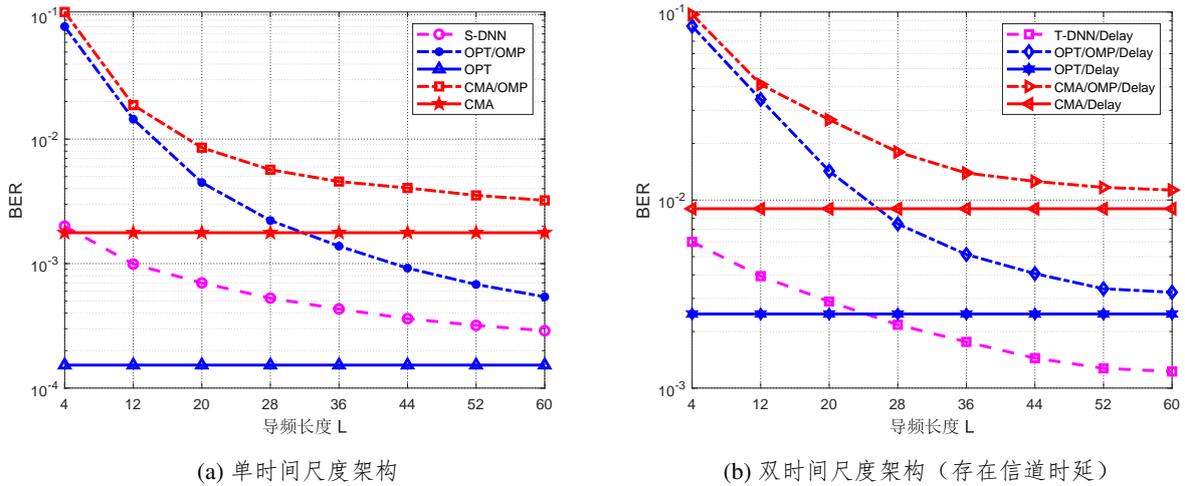


图 5.13 系统BER性能随导频长度 L 的变化

图5.13(a)展示了单时间尺度场景下，所提出的S-DNN和基线算法实现的系统BER性能随导频长度 L 的变化。在 L 相同时，S-DNN的性能显著优于OPT/OMP和CMA/OMP，且在 L 较小时性能增益更为明显。这表明当信道状态信息的观测样本有限时，例如 $L = 12$ ，端到端联合设计方案显著优于各模块分别单独设计。此外，S-DNN在 $L = 28$ 时实现了与OPT/OMP在 $L = 60$ 时几乎相同的BER性能。这表明S-DNN在显著减少导频开销的情况下仍然可以实现较好的性能。另外，S-DNN可以超越具有完美信道状态信息的CMA实现的性能，并逼近具有完美信道状态信息的OPT实现的BER性能下界。图5.13(b)展示了在考虑实际信道失配的场景下，所提出的T-DNN和基线算法实现的系统BER性能与导频长度 L 的关系。在 L 相同时，T-DNN的性能显著优于OPT/OMP/Delay和CMA/OMP/Delay。此外，T-DNN在 $L = 20$ 时取得了与OPT/OMP/Delay在 $L = 60$ 时几乎相同的性能，这表明所提出的T-DNN在显著降低导频长度的情况下仍可以实现较好的性能。当 $L > 24$ 时，T-DNN可以超越在完美信道状态信息条件下的OPT/Delay，这验证了T-DNN在处理信道失配时的有效性。

图5.14描述了双时间尺度架构下系统BER性能随时延 τ 的变化。随着 τ 的增加，传统的单时间尺度预编码算法BER上升显著，S-DNN的BER性能也有一定上升。而T-DNN由于节省了大量信道反馈比特，BER性能上升最为缓慢。尤其当 $\tau > 3$ ms时，T-DNN的性能超越OPT/Delay。该结果验证了双时间尺度端到端联合学习框架在处理由信道时延引起的信道失配的有效性。

图5.15展示了不同模拟移相器量化比特数 Q_{RF} 的系统BER性能随SNR的变化。随着 Q_{RF} 的增加，S-DNN的BER下降，并且在 $Q_{RF} = 8$ 时的性能可以接近无限量化精

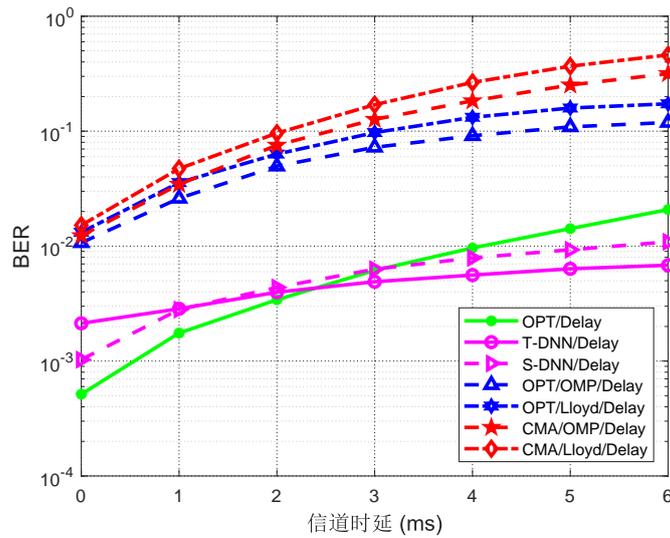


图 5.14 双时间尺度架构下系统BER性能随时延 τ 的变化

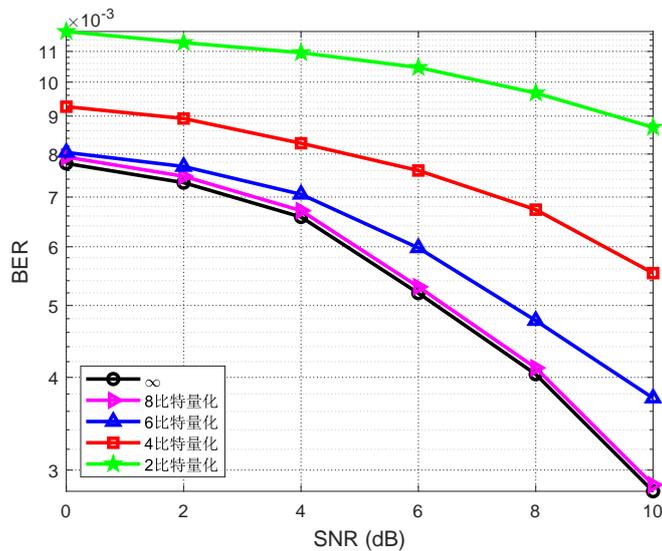


图 5.15 不同模拟移相器量化比特数 Q_{RF} 的系统BER性能随SNR的变化

度 $Q_{RF} = \infty$ 的移相器实现的性能。

图5.16验证了提出的S-DNN和T-DNN在OFDM系统中仍取得了较好的BER性能。为模拟OFDM系统中的毫米波宽带频率选择性衰落信道，采用群延迟信道模型，时延扩展和用户移动速度分别设置为1 μ s和3 km/h。物理资源块的数量设为24个，每个物理资源块包含12个子载波。因此，子载波数目总共有288个，子载波间隔设为30 kHz。可以看到，在OFDM系统中S-DNN和T-DNN的性能接近于在窄带毫米波信道环境中的单子载波系统下的性能，并且显著优于基线算法。

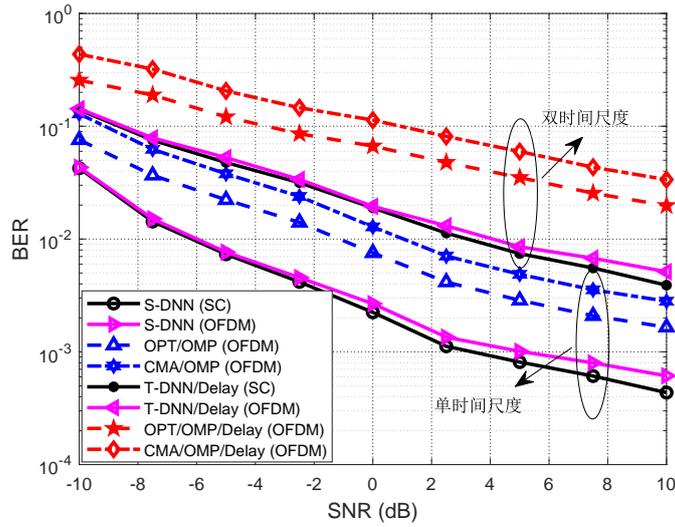


图 5.16 OFDM系统的BER性能

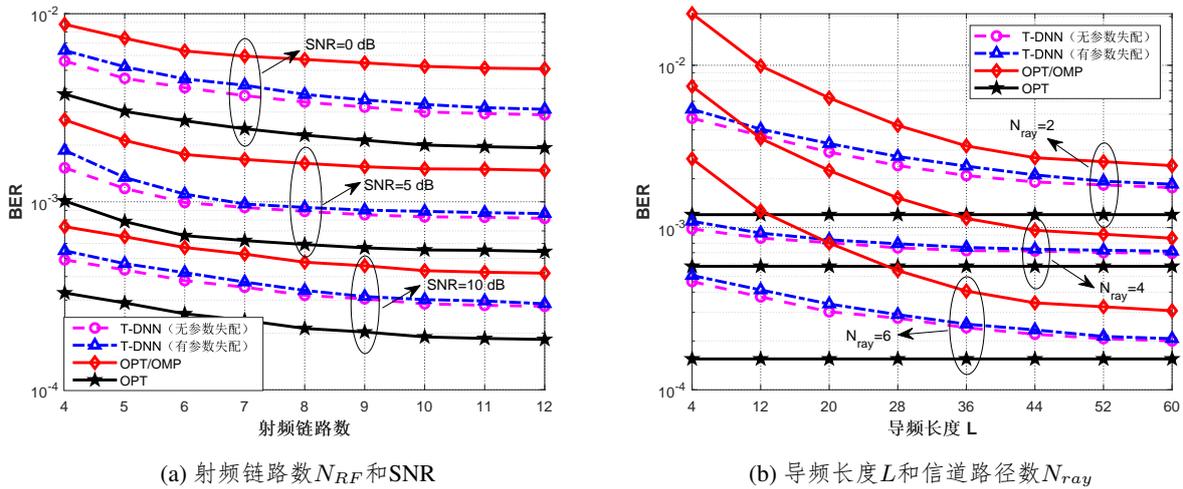


图 5.17 网络对于不同系统参数失配的泛化能力

5.6.4 网络的泛化能力

图5.17(a)描述了所提出的网络对于在系统参数 N_{RF} 和SNR失配场景下的泛化能力。在 $N_{RF} = 12$, $N_{ray} = 4$, $L = 36$, 和 $SNR \in \{0, 5, 10\}$ dB的参数配置下训练网络。并在固定 $N_{ray} = 4$ 和 $L = 36$, 变化 N_{RF} 和SNR的系统参数配置下对网络进行测试。由图可见, 因为训练阶段和测试阶段的系统参数 N_{RF} 和SNR存在失配, 网络存在较小的性能损失。存在系统参数失配的T-DNN性能仍优于OPT/OMP, 并且接近完美信道状态信息下的OPT实现的性能。这表明所提出的T-DNN对不同 N_{RF} 和SNR具有较好的泛化能力。此外, 这说明当没有关于系统参数的完美先验知识时, 在更广泛的系统参数上进行网络训练, 例如训练数据

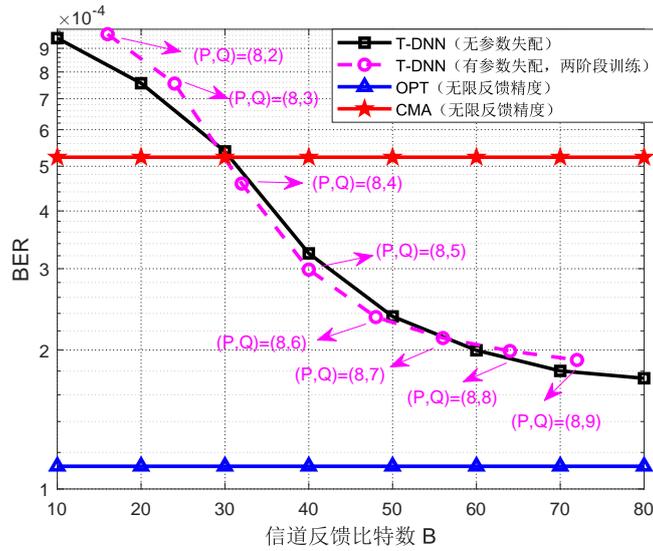


图 5.18 基于两阶段训练方式的网络对于信道反馈比特数 B 的泛化能力

集包含较广泛的SNR，可以提升网络对不同系统参数的泛化能力。另外，有系统参数失配导致的网络性能损失随着 N_{RF} 和SNR的增加而降低。这主要是因为当训练数据集和测试数据集的系统参数间失配程度变小时，由于系统参数失配导致的网络性能损失会变小。

图5.17(b)描述了所提出的网络对于在系统参数 L 和 N_{ray} 失配场景下的泛化能力。在 $N_{RF} = 12$, SNR= 10 dB, $L = 60$ 和 $N_{ray} \in \{2, 4, 6\}$ 的参数配置下训练网络。并在固定 $N_{RF} = 12$ 和SNR= 10 dB, 变化 L 和 N_{ray} 的系统参数配置下对网络进行测试。尽管与不存在系统参数失配场景下的T-DNN相比, 参数失配会导致较小的性能损失, 但其性能仍优于OPT/OMP。这验证了所提出的T-DNN对不同 L 和 N_{ray} 具有较好的泛化能力。此外, 该性能损失随着 L 的增加而降低, 这是因为当导频长度 L 较短时, 网络倾向于充分利用训练数据的分布, 并调整其训练参数以适应该特定的分布, 从而导致对不同分布参数的泛化能力较差。相比之下, 当训练导频序列的长度不再是制约因素时, 例如 $L = 64$ 时, 网络可以更好地处理具有不同信道路径数 N_{ray} 的, 分布更广、更复杂的信道样本。另外, 在更广泛的系统参数上训练网络, 例如训练数据集包含不同的 N_{ray} , 可以提升网络的泛化能力。

图5.18展示了基于两阶段训练方式的网络对于信道反馈比特数 B 的泛化能力。由图可知, 采用本章提出的改进两阶段训练方法, 由于系统参数 B 的失配所导致的性能下降可以忽略不计, 它提供了一个可以处理不同 B 的通用网络架构和训练方式。此外, 当 B 足够大时, 存在系统参数失配场景下, T-DNN 的性能显著超越无限反馈精度的CMA, 且接近无限反馈精度的OPT实现的BER性能下界。结果表明, 通过设置不同的 Q 值, 所提出的改进两阶段训练方法可以显著提高网络对于 B 的泛化能力。

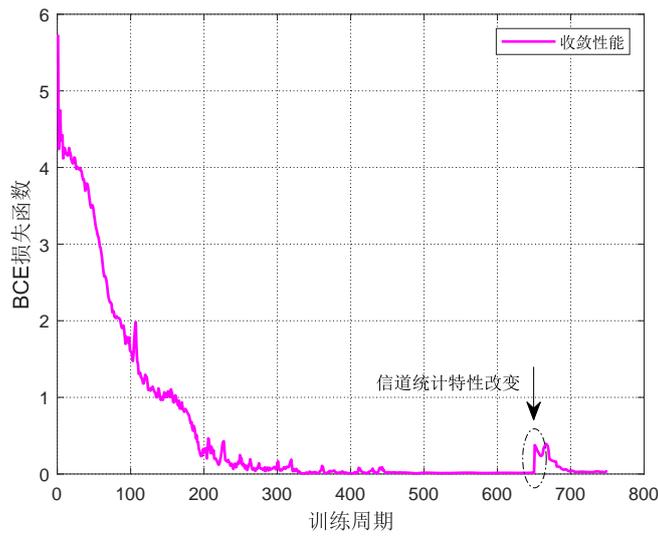


图 5.19 迁移学习过程中网络损失函数BCE的变化

在实际应用中，信道统计特性连续不断地变化，但相邻超帧的信道统计特性变化不大。因此，可以搜集信道统计特性变化后的信道样本，以便基于之前训练过的网络进行在线训练和微调。具体而言，使用迁移学习进行网络在线训练，其中网络的大部分层的参数被固定，只有最后几层的网络参数被训练更新。由此，部分网络参数可以被快速更新，网络可以快速收敛，从而快速适应新的信道统计特性。图5.19描述了信道统计特性变化时，迁移学习过程中网络的损失函数BCE的变化。可以看到，当信道统计特性发生变化时，网络的BCE损失函数值先增加，随后在短时间内减小，这表明网络可以快速适应信道统计特性的变化。当信道统计特性变化速度变快或变慢时，需要自适应地调整帧结构中帧和时隙长度。例如，当信道统计特性变化很快时，则需要缩短帧和时隙的长度，以获得更多高维的原始信道样本来跟踪信道统计特性的变化。此外，收发端长时间尺度模拟预编码矩阵需要被更频繁地更新，以更好更快地适应信道统计特性的变化。

5.7 本章小结

本章提出了一个基于深度学习的FDD毫米波大规模MIMO系统框架。该框架由基于DNN的导频训练、信道估计、信道反馈和混合预编码等模块组成。为了进一步降低信令传输开销，以及减轻由信道时延导致的信道失配，基于该深度学习框架提出了一种新颖的双时间尺度网络。其由长时间尺度网络和短时间尺度网络组成。此外，针对该包含用于信道反馈的二值离散层网络架构，设计了两阶段训练方法。该提出的双时间尺度网络可以扩展到多载波OFDM系统。仿真结果表明，该网络实现的误码率性能明显优于传统方案，且

信令和导频传输开销显著降低。未来的工作可以将该框架扩展到多用户和多小区通信系统，以及未来6G系统中更新颖、更具挑战性的系统，例如智能反射面系统等。

6 基于特征重要性码本和星座点设计的端到端鲁棒语义通信系统

6.1 引言

前四章针对物理层收发机的不同模块设计了模型驱动网络、DRL网络和端到端自编码器网络等深度学习方案，但传输的都是IID的比特信息流，而未考虑信源编码和语义信息的传输。本章将在前四章的基础上，考虑基于信源信道联合编码的语义物理层收发机设计。众所周知，现有的物理层收发机注重高效的符号传输和准确的符号恢复^[1]，使用BER作为收发机的性能指标。然而，6G网络中各种新兴的应用产生了前所未有的巨大数据量，使传统的通信系统达到了性能瓶颈^[267]。随着深度学习的发展，语义通信作为一种提取和传输数据中与任务相关信息的关键技术，受到了广泛关注^[58]。例如，在针对目标检测任务的图像传输中，目标的位置和形状与目标检测任务相关，而图像的背景与任务无关，因此无需传输。此外，语义通信对较差的信道环境具有较强的鲁棒性，适合用于对可靠性要求高的应用。这些优势促使研究者在设计通信系统时，考虑传输相较于比特流更高层的语义信息，以提高传输的准确性和效率。

尽管基于深度学习的端到端语义通信系统在某些任务中实现了较好的性能，但其对噪声的鲁棒性仍需进一步研究。已有研究分析了不同类型图像和文本噪声的成因和特征，并提出了去噪算法^[268]。然而，在语义通信系统中存在着一种特殊的语义噪声，会引起对语义信息的误解和解码错误，从而导致发送端传输的语义信息和接收端重建的语义信息之间存在失真。语义噪声可以在不同的阶段产生，包括语义信息编码、数据传输和语义信息解码阶段^[230]。在语义信息编码阶段，语义噪声会造成经过语义信息编码后的信号与原始信号之间的信息失真，这与语义编码网络的表征能力有关。在数据传输阶段，由于信道衰落导致的信号失真，以及恶意攻击者发送的干扰信号都会引入语义噪声。在语义信息解码阶段，解码错误、不正确的表征和含义的混淆都会在接收端引入语义噪声。对于不同类别的信源，例如文本和图像，语义噪声略有不同^[228]。文本中的语义噪声是指语义模糊，其中句子中某些单词的微小变化会导致深度学习模型对句子的语义误解，例如同义词替换或随机颠倒字母顺序^[269]。图像中的语义噪声可以建模为对抗扰动^[270]，即对图像像素做一些细

微的人眼无法察觉的改动，从而误导深度学习模型输出错误的结果，导致系统性能的恶化。

针对端到端鲁棒语义通信系统在安全性和可靠性方面的研究，在未来的无人驾驶和智慧医疗等6G场景中至关重要。虽然在图像处理领域已经有一些对抗扰动的建模，但在某些场景下不够鲁棒，且忽略了图像传输过程中无线信道和传输开销对系统的影响。因此，物理层收发机中的语义噪声模型还没有得到充分的研究。本章建模了图像在语义通信系统传输过程中的语义噪声，并设计了端到端鲁棒语义通信系统来对抗语义噪声，同时降低传输开销。具体而言，针对不同场景建模了物理层收发机中存在的两种语义噪声模型。一是在发送端实时地产生样本依赖的语义噪声，即在每张图像上添加不同的语义噪声。针对难以获取CSI和传输符号的场景，提出了一种在接收端产生的非样本依赖语义噪声。其针对不同输入样本，产生相同的语义噪声，添加到不同的传输图像中，造成大多数图像解码失败。为提升通信系统对语义噪声的鲁棒性，提出了一种带有权重扰动的对抗训练方法，将带有语义噪声的样本加入训练数据集中，以高效求解系统的鲁棒优化问题。随后，设计了基于ViT架构^[271]的掩蔽VQ-VAE，作为端到端鲁棒语义通信系统的核心架构，并提出掩蔽原始图像中语义噪声出现概率较高图像块的策略。此外，设计了一个由发送端和接收端共享的离散码本，用于表示经过编码的特征矢量。它侧重于表示任务相关的重要特征，忽略噪声相关的细节特征，从而降低了语义噪声对端到端语义通信系统的影响。

为进一步提升系统鲁棒性，设计FIM用于动态学习特征重要性，并将其融合到掩蔽VQ-VAE架构中。该FIM能抑制与任务无关和与噪声相关的编码特征。因此，发送端只需要传输码本中与任务相关的重要特征对应的标号索引。此外，SNR被设计为FIM输入的一部分，这确保了所提出的网络架构可以在广泛的SNR动态范围内高效运行。针对FIM和掩蔽VQ-VAE中的码本，设计了基于语义相似度的损失函数，从而进一步提升系统鲁棒性，并给出了鲁棒性能分析。值得一提的是，已有的语义通信系统设计将信源数据直接映射为信道符号进行传输，即星座点可以出现在星座图的任何地方，被称为全分辨率星座图。然而，由于当前数字通信系统的量化精度有限，全分辨率星座图在实际数字通信系统中难以实现^[231-233]。而本章提出的掩蔽VQ-VAE模型设计了一个基于离散码本的数字语义通信系统，在现有的数字通信系统中更易实现，因为编码特征的标号索引可以通过现有的有限星座图直接映射为传输符号，在信道上高效传输。

本章内容安排如下。6.2节对通信系统中存在的语义噪声进行了建模，并提出了一个可用于对抗语义噪声的通信系统总体框架。6.3节设计了带有特定掩蔽策略的掩蔽VQ-VAE，以及用于表示编码特征的离散码本。6.4节通过设计具有动态SNR的FIM来提升端到端语义通信系统的鲁棒性。6.5节展示了仿真结果。6.6节对本章进行了总结。

6.2 鲁棒端到端语义通信系统框架

本章节建模了语义噪声，并提出了包含对抗训练的鲁棒端到端语义通信系统框架，用于对抗语义噪声。

6.2.1 端到端语义通信系统

如图6.1(a)所示，发送端将信源数据 \mathbf{s} 映射为一串发送符号 \mathbf{x} ，通过无线信道发送到接收端。接收符号 \mathbf{y} 在接收端被解码并用于恢复信源数据 $\hat{\mathbf{s}}$ 。发送端和接收端都部署DNN，且对这些DNN进行联合设计。具体而言，发送端的网络包含语义编码器和信道编码器，而接收端的网络包含语义解码器和信道解码器。语义编码器从信源数据中提取语义信息，将其映射为经过编码的特征向量，语义解码器从接收到的信道中恢复出信源数据。信道编码器和信道解码器旨在消除由无线信道带来的信号畸变。

本章研究传输的信源数据为图片的语义通信系统。考虑一个具有 N_t 根发送天线和 N_r 根接收天线的系统。经过发送端网络编码的传输符号流可以表示为

$$\mathbf{x} = f_2(f_1(\mathbf{s}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2), \quad (6.1)$$

其中 $\mathbf{x} \in \mathbb{C}^{N_t \times 1}$ ， $\boldsymbol{\theta}_1$ 和 $\boldsymbol{\theta}_2$ 分别表示语义编码器 $f_1(\cdot)$ 的训练参数和信道编码器 $f_2(\cdot)$ 的训练参数。随后，接收符号 $\mathbf{y} \in \mathbb{C}^{N_r \times 1}$ 表示为

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad (6.2)$$

其中 $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ 表示信道矩阵， $\mathbf{n} \in \mathbb{C}^{N_r \times 1} \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I})$ 为AWGN。相应地，解码之后的符号（恢复出的信源数据）表示为

$$\hat{\mathbf{s}} = g_1(g_2(\mathbf{y}; \boldsymbol{\Theta}_2); \boldsymbol{\Theta}_1), \quad (6.3)$$

其中 $\boldsymbol{\Theta}_1$ 和 $\boldsymbol{\Theta}_2$ 分别表示语义解码器 $g_1(\cdot)$ 的训练参数和信道解码器 $g_2(\cdot)$ 的训练参数。为表述清楚起见，记 $\boldsymbol{\theta}$ 为训练参数， $f_{\boldsymbol{\theta}}(\cdot)$ 表示该语义通信系统的所有网络。因此，接收端恢复出的信源数据可以写成 $\hat{\mathbf{s}} = f_{\boldsymbol{\theta}}(\mathbf{s})$ 。该语义通信系统旨在最小化发送端发送的信源数据和接收端恢复出的信源数据之间的语义误差，同时尽可能降低传输开销。

6.2.2 语义噪声模型

本节讨论发送端语义噪声模型和接收端语义噪声模型。

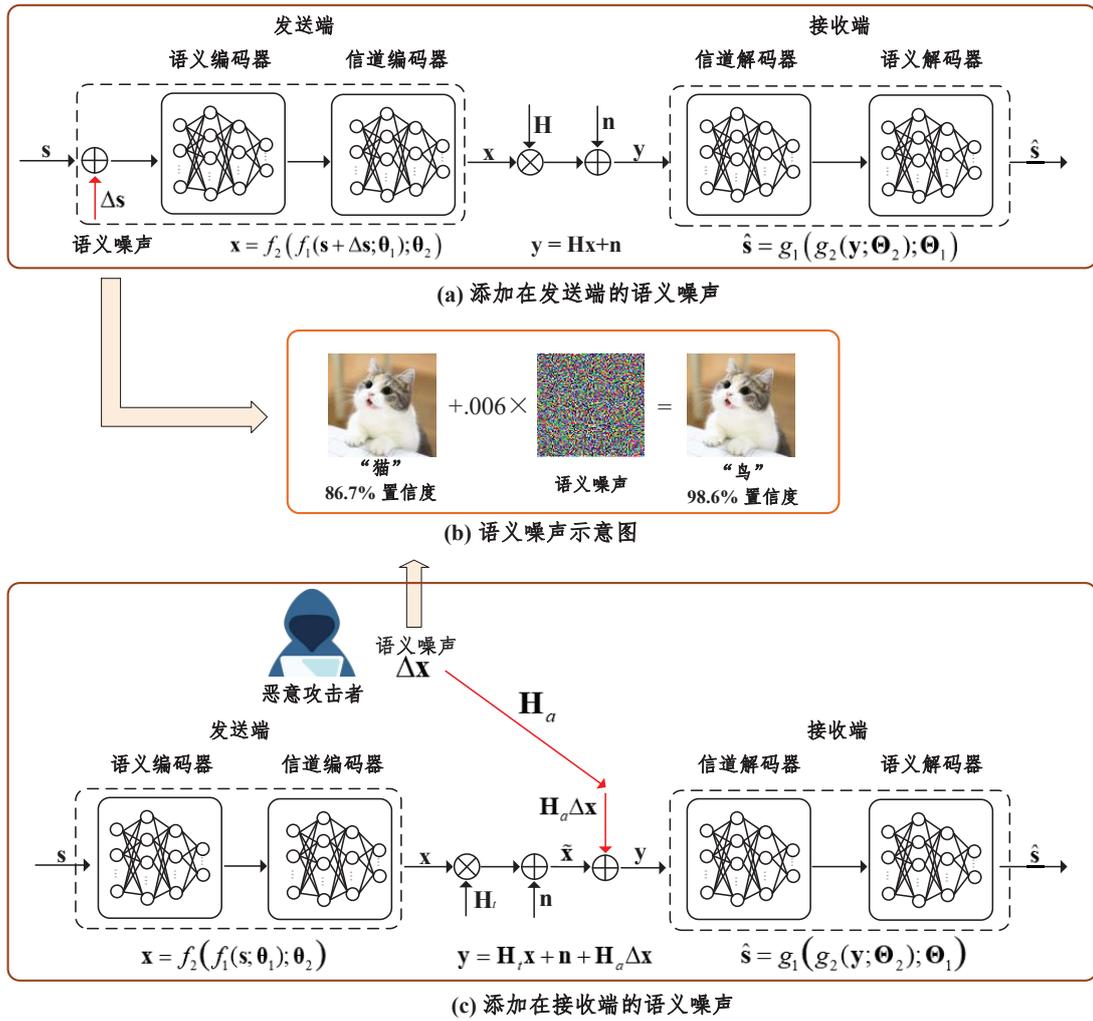


图 6.1 语义噪声模型及鲁棒语义通信系统框架

6.2.2.1 发送端语义噪声

发送端语义噪声一般在发送端编码时产生，如图6.1(a)所示。考虑如下场景：有一个恶意攻击者攻击网站，下载图像数据集，将语义噪声加在该数据集的每张图片上，再将该修改过的数据集上传至网站。语义噪声对编码过程有严重的影响，并且会误导深度学习模型使其产生错误的结果，从而导致任务失败。然而，由于语义噪声功率很小，所以很难被合法用户检测到。合法用户通常会把这些带有语义噪声的数据当作正常的数据使用。此外，语义噪声同样存在于自然界中。例如，通过拍摄一些经过人为处理的特殊图片，也包含语义噪声，同样会误导深度学习模型使其产生错误的结果，从而导致任务失败，例如图像分类出错^[249]。

语义通信系统的目标是最小化某一特定任务的损失函数，比如：图像重建任务的MSE，图像分类任务的交叉熵。与之相反，语义噪声旨在最大化任务的损失函数，从而

尽可能使得任务失败。令 $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_I\}$ 为一批从训练数据集中采样出来的图片样本。针对第 i 张图片 \mathbf{s}_i 产生的语义噪声，可被建模为求解如下优化问题

$$\mathcal{P}1: \max_{\Delta \mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}_i + \Delta \mathbf{s}_i), \mathbf{z}_i) \quad (6.4a)$$

$$\text{s.t.} \quad \|\Delta \mathbf{s}_i\|_p \leq \epsilon, \quad (6.4b)$$

其中 \mathbf{s}_i 和 $f_{\theta}(\mathbf{s}_i + \Delta \mathbf{s}_i)$ 分别表示第 i 张输入网络的图片和网络对应的输出， $\Delta \mathbf{s}_i$ 为针对第 i 张图片产生的语义噪声， $\mathcal{L}(\cdot)$ 为针对某一任务的神经网络的损失函数。此外， \mathbf{z}_i 表示和输入 \mathbf{s}_i 相关的标签，比如：分类任务中图像所属的类别标签，以及图像重建任务中的原始图像。此外， $\|\cdot\|_p$ 为 p -范数，约束(6.4b)限制了语义噪声功率，避免图像中的语义噪声被识别出来。除非特殊说明，本章采用 $p = \infty$ ，即无穷范数。

为了求解该问题，可以采用文献[243]提出的FGSM算法，核心思想是将损失函数线性化

$$\mathcal{L}(f_{\theta}(\mathbf{s}_i + \Delta \mathbf{s}_i), \mathbf{z}_i) \approx \mathcal{L}(f_{\theta}(\mathbf{s}_i), \mathbf{z}_i) + (\Delta \mathbf{s}_i)^T \nabla_{\mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}_i), \mathbf{z}_i). \quad (6.5)$$

取 $\Delta \mathbf{s}_i = -\alpha \nabla_{\mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}_i), \mathbf{z}_i)$ 可以将线性化之后的损失函数最小化，其中 α 为缩放因子，用于控制语义噪声的功率，使其小于等于 ϵ ，从而满足功率约束(6.4b)。由此，可以得到功率为 ϵ 的语义噪声

$$\Delta \mathbf{s}_i = \epsilon \text{sign}(\nabla_{\mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}_i), \mathbf{z}_i)), \quad (6.6)$$

当 $x \geq 0$ 时， $\text{sign}(x) = 1$ ；当 $x < 0$ 时， $\text{sign}(x) = -1$ 。由此，带有语义噪声的被污染的图像样本可以写成 $\mathbf{s}'_i = \mathbf{s}_i + \Delta \mathbf{s}_i$ 。该语义噪声是通过梯度下降法一步迭代产生的。为了提升其对系统的影响，产生更有效的语义噪声 $\Delta \mathbf{s}_i$ ，提出使用如下迭代方式产生语义噪声

$$\mathbf{s}'_i{}^{(k+1)} = \Pi_{\epsilon}(\mathbf{s}'_i{}^{(k)} + \alpha \cdot \text{sign}(\nabla_{\mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}'_i{}^{(k)}), \mathbf{z}_i))), \quad (6.7)$$

其中 k 表示第 k 次迭代， Π 为投影算子。选择满足条件 $K\alpha > \epsilon$ 的 α ，确保可以充分利用功率限制 ϵ 产生有效的语义噪声，其中 K 为迭代次数。

6.2.2.2 接收端语义噪声

接收端语义噪声一般产生于信号传输和解码阶段，将会导致解码的失败和接收端对传输信号的语义的误解，如图6.1(c)所示。该语义噪声产生的原因有硬件非理想特性，由信道引起的传输信号畸变，恶意攻击者的攻击等^[248]。考虑一个发送机传输经过编码的信号 \mathbf{x} 至接收端，一个恶意攻击者发送语义噪声 $\Delta \mathbf{x}$ 用于攻击该次传输。由此，接收机接收到的信号为

$$\mathbf{y} = \mathbf{H}_t \mathbf{x} + \mathbf{n} + \mathbf{H}_a \Delta \mathbf{x} \triangleq \tilde{\mathbf{x}} + \mathbf{H}_a \Delta \mathbf{x}, \quad (6.8)$$

其中 $\tilde{\mathbf{x}} \triangleq \mathbf{H}_t \mathbf{x} + \mathbf{n}$ 表示没有语义噪声的接收信号， \mathbf{H}_t 为发送端到接收端之间的信道， \mathbf{H}_a 为攻击者到接收端之间的信道， \mathbf{n} 是高斯白噪声。注意到本章节提出的接收端语义噪声模型(6.8)为一般化的模型，其中如果移除 \mathbf{H}_a ，可以表征由接收端硬件非理想特性或者信道衰落引入的语义噪声。

为了产生章节6.2.2.1中的样本依赖的语义噪声（即针对每个不同样本产生不同的针对该样本的语义噪声），需要假设攻击者知道：（1）攻击者和接收端之间的信道 \mathbf{H}_a ；（2）接收端的接收信号 $\tilde{\mathbf{x}}$ 。某些场景下，该假设在实际无线通信系统中并不一定实际。因此，为了更好地匹配实际系统的工作原理，需要建模一种非样本依赖的语义噪声模型（即针对每个不同样本产生相同的语义噪声） $\Delta \mathbf{x}$ ，该噪声对该数据集 \mathcal{S} 中所有的样本都具有一定攻击效果，且假定攻击者仅仅知道攻击者和接收端之间的信道统计特性，而不需要知道真实的信道样本¹。具体而言，先根据信道的统计特性产生 N 个信道样本 $\{\mathbf{H}_a^{(1)}, \mathbf{H}_a^{(2)}, \dots, \mathbf{H}_a^{(N)}\}$ 并搜集一批接收信号 $\{\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \dots, \tilde{\mathbf{x}}^{(N)}\}$ 。随后，根据 $\tilde{\mathbf{x}}^{(n)}$ 和 $\mathbf{H}_a^{(n)}$ for $n = 1, 2, \dots, N$ ，而不是当前真实的接收信号和信道样本，产生非样本依赖的语义噪声 $\Delta \mathbf{x}^{(n)}$ 。选择合适的 α 满足 $K\alpha > \epsilon$ 确保充分利用功率约束 ϵ 产生有效的语义噪声。为了最大化接收端收到的语义噪声功率，从而对通信系统产生尽可能大的影响，攻击者需要充分利用信道信息 \mathbf{H}_a 。如果语义噪声 $\Delta \mathbf{x}$ 乘上信道的共轭转置 \mathbf{H}_a^* ，经过信道传输之后到达接收端的语义噪声功率将被最大化。针对所有样本产生的非样本依赖语义噪声矢量 $\{\Delta \mathbf{x}^{(n)}\}$ 被加权求和并功率归一化。产生该语义噪声的详细细节请见表6.1，其中 θ_d 表示解码网络 $g_{\theta_d}(\cdot)$ 的训练参数， $g_{\theta_d}(\tilde{\mathbf{x}}^{(n)})$ 表示将第 n 个样本 $\tilde{\mathbf{x}}^{(n)}$ 输入解码网络 $g_{\theta_d}(\cdot)$ 得到的输出。接下来，根据章节6.2.2.1提出的样本依赖语义噪声模型设计了鲁棒语义通信系统。非样本依赖语义噪声可以类似处理，提出的鲁棒语义通信系统同样可以高效地对抗非样本依赖语义噪声。

注 6.1. 值得注意的是，在原始图像中天然存在语义噪声。以图像分类任务为例，这里由语义噪声导致的“误解”指的是“误分类”，比如：发送端发送了一张狗的图片，但接收端误分类为猫。对于一个数据集，由于原始图像中天然存在的语义噪声，误分类率不可能为0。此外，不同数据集的语义噪声功率通常是不同的。具体而言，一些较为简单的数据集，其语义噪声功率相对较低，例如：*MNIST*数据集，很容易达到较高的准确率。相比之下较为复杂的数据集通常语义噪声功率较高，例如：*ImageNet*数据集。此外，不同的深度学习模型处理语义噪声的能力也不同。具有较复杂网络结构的高效模型通常可以更有效地消除语义噪声的影响，从而实现更高的分类准确率，例如：*ResNet-101*。本章提出的语义噪声模

¹发送端语义噪声的产生需已知：发送端和接收端之间的信道、传输图像；接收端语义噪声的产生需已知：攻击者和接收端之间的信道统计特性、接收端收到的符号。

型加强了这种“误解”，降低了网络模型对图片分类的准确率，因此对语义通信系统提出了更高的鲁棒性要求。

表 6.1 接收端非样本依赖语义噪声的生成算法

算法 6.1 接收端非样本依赖语义噪声的生成算法

- 1: **输入:** 生成的信道样本 $\{\mathbf{H}_a^{(1)}, \mathbf{H}_a^{(2)}, \dots, \mathbf{H}_a^{(N)}\}$ 和搜集到的接收信号样本 $\{\tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \dots, \tilde{\mathbf{x}}^{(N)}\}$, 以及对应的真实标签 $\{\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(N)}\}$, 迭代次数 K , 噪声功率 ϵ , 缩放因子 α , 加权系数 ρ ;
 - 2: **输出:** 接收端非样本依赖语义噪声 $\Delta \mathbf{x}$;
 - 3: **初始化:** $\delta_1 = \mathbf{0}$, $\Delta \mathbf{x}^{(n)} = \mathbf{0}$, 以及 $\delta_{norm} = \mathbf{0}$;
 - 4: **for** $n = 1 : N$ **do**
 - 5: % 用于产生针对第 n 个样本的样本依赖语义噪声
 - 6: **for** $k = 1 : K$ **do**
 - 7: $\delta_{norm} = \frac{\mathbf{H}_a^{(n)*} \nabla_{\tilde{\mathbf{x}}_k^{(n)}} \mathcal{L}(g_{d_\theta}(\tilde{\mathbf{x}}_k^{(n)}), \mathbf{z}^{(n)})}{\|\mathbf{H}_a^{(n)*} \nabla_{\tilde{\mathbf{x}}_k^{(n)}} \mathcal{L}(g_{d_\theta}(\tilde{\mathbf{x}}_k^{(n)}), \mathbf{z}^{(n)})\|_2}$;
 - 8: $\tilde{\mathbf{x}}_{k+1}^{(n)} = \tilde{\mathbf{x}}_k^{(n)} + \alpha \mathbf{H}_a^{(n)} \delta_{norm}$;
 - 9: $\delta_{k+1} = \delta_k + \alpha \delta_{norm}$;
 - 10: **end for**
 - 11: 加权平均: $\Delta \mathbf{x}^{(n)} = \Delta \mathbf{x}^{(n-1)} + \rho \frac{\delta_K}{\|\delta_K\|_2}$;
 - 12: 归一化: $\Delta \mathbf{x}^{(n)} = \epsilon \frac{\Delta \mathbf{x}^{(n)}}{\|\Delta \mathbf{x}^{(n)}\|_2}$;
 - 13: **end for**
 - 14: 输出接收端非样本依赖语义噪声 $\Delta \mathbf{x}$.
-

6.2.3 对抗训练

6.2.3.1 基本对抗训练方式

对抗训练是一种可用于对抗语义噪声的训练方式，其核心思想是将被语义噪声污染过的训练样本加入到训练数据集中^[243]。具体而言，网络训练参数 θ 和语义噪声 $\Delta \mathbf{s}_i$ ，交替更新用于提升模型的鲁棒性。该过程可以被建模成如下 min-max 优化问题

$$\mathcal{P}2: \min_{\theta} \frac{1}{I} \sum_{i=1}^I \max_{\Delta \mathbf{s}_i} \mathcal{L}(f_{\theta}(\mathbf{s}_i + \Delta \mathbf{s}_i), \mathbf{z}_i) \quad (6.9a)$$

$$\text{s.t. } \|\Delta \mathbf{s}_i\|_p \leq \epsilon, \quad (6.9b)$$

其中 I 表示训练样本的数量。

为求解问题(P2)，需要交替执行如下两步：(1) 根据(6.7)或者算法表格6.1计算 \mathbf{s}'_i ，得到语义噪声 $\Delta\mathbf{s}_i = \mathbf{s}'_i - \mathbf{s}_i$ 。注意到 $\boldsymbol{\theta}$ 在这一步是固定的，且需要将样本 \mathbf{s}'_i 加入训练数据集。(2) 基于训练样本 \mathbf{s}'_i ，使用SGD更新网络参数 $\boldsymbol{\theta}$ ，从而最小化损失函数。

6.2.3.2 带有权重扰动的对抗训练

为了进一步提升语义通信系统对抗语义噪声的鲁棒性，本章节在基本对抗训练方法的基础上增加了权重扰动 $\boldsymbol{\nu}$ ，并将该问题重新建模为

$$\mathcal{P3}: \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\nu}} \frac{1}{I} \sum_{i=1}^I \max_{\Delta\mathbf{s}_i} \mathcal{L}(f_{\boldsymbol{\theta}+\boldsymbol{\nu}}(\mathbf{s}_i + \Delta\mathbf{s}_i), \mathbf{z}_i) \quad (6.10a)$$

$$\text{s.t.} \quad \|\Delta\mathbf{s}_i\|_p \leq \epsilon, \quad \|\boldsymbol{\nu}\|_p \leq \gamma \|\boldsymbol{\theta}\|_p. \quad (6.10b)$$

直观来看，语义噪声 $\Delta\mathbf{s}_i$ 和权重扰动 $\boldsymbol{\nu}$ 分别使得第 i 个样本的损失函数值和所有样本的损失函数值增加。因此，问题中的两个“max”可以更高效地求解该问题的内部最大化子问题，从而可以找到该min-max问题更好的解^[253]。提出表6.2中的算法求解问题(P3)，其中权重扰动 $\boldsymbol{\nu}$ 的更新公式为

$$\boldsymbol{\nu} \leftarrow \Pi_{\gamma} \left(\boldsymbol{\nu} + \eta \frac{\nabla_{\boldsymbol{\nu}} \frac{1}{I} \sum_{i=1}^I \mathcal{L}(f_{\boldsymbol{\theta}+\boldsymbol{\nu}}(\mathbf{s}_i + \Delta\mathbf{s}_i), \mathbf{z}_i)}{\left\| \nabla_{\boldsymbol{\nu}} \frac{1}{I} \sum_{i=1}^I \mathcal{L}(f_{\boldsymbol{\theta}+\boldsymbol{\nu}}(\mathbf{s}_i + \Delta\mathbf{s}_i), \mathbf{z}_i) \right\|} \|\boldsymbol{\theta}\| \right), \quad (6.11)$$

该表达式可以通过类似于表达式(6.7)的方法推导得到。

表 6.2 用于求解问题P3的对抗训练算法

算法 6.2 用于求解问题P3的对抗训练算法

- 1: **输入**: 包含输入图像 \mathbf{s}_i 及其对应标签 \mathbf{z}_i 的训练数据集;
 - 2: **输出**: 模型的训练参数 $\boldsymbol{\theta}$;
 - 3: **for** $m = 1 : M$ **do**
 - 4: 固定 $\boldsymbol{\theta}$ 和 $\boldsymbol{\nu}$ ，根据(6.7)或表6.1计算 \mathbf{s}'_i ，得到语义噪声 $\Delta\mathbf{s}_i = \mathbf{s}'_i - \mathbf{s}_i$ ，并将 \mathbf{s}'_i 作为训练样本加入训练数据集;
 - 5: 固定 $\boldsymbol{\theta}$ 和 $\Delta\mathbf{s}_i$ ，通过采用(6.11)中的一步网络反向传播来更新 $\boldsymbol{\nu}$ ，从而最大化 $\mathcal{L}(\cdot)$;
 - 6: 固定 $\boldsymbol{\nu}$ ，根据训练样本 \mathbf{s}'_i ，通过采用SGD更新 $\boldsymbol{\theta}$ ，从而最小化 $\mathcal{L}(\cdot)$ 。
 - 7: **end for**
-

6.3 基于掩蔽矢量量化变分自编码器的码本设计

本节设计了基于掩蔽矢量量化变分自编码器的码本设计（Vector Quantized-Variational AutoEncoder, VQ-VAE）。新颖的掩蔽技术和离散码本设计被用于提升语义通信系统的鲁

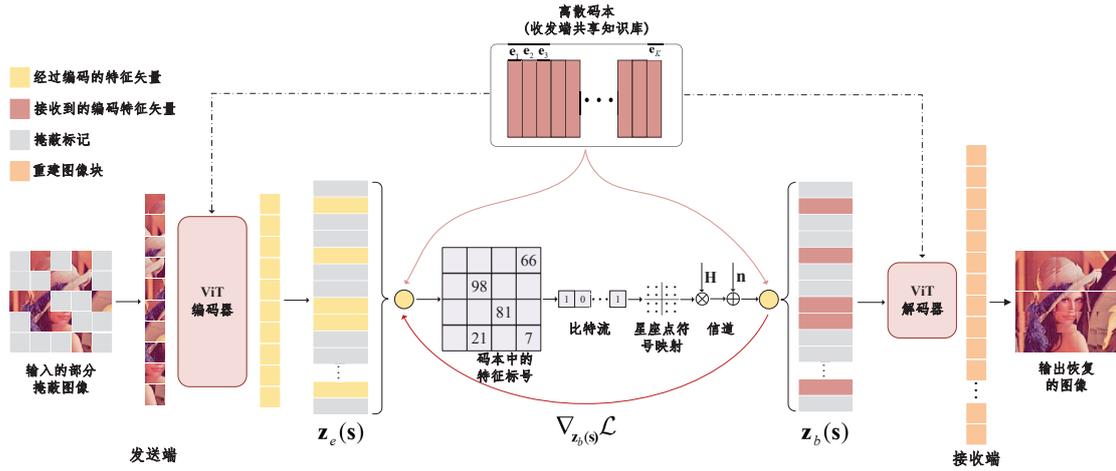


图 6.2 带有离散码本设计的掩蔽VQ-VAE的架构

棒性同时降低系统的传输开销。此外，提供了相应的性能分析，并基于语义相似度提出了一个新颖的损失函数，来提升系统鲁棒性。本章节设计的语义码本不同于信道反馈的码本^[1]：（1）本节提出的掩蔽VQ-VAE框架采用深度学习网络架构，联合训练发送端的编码网络、码本，以及接收端的解码网络，而信道反馈码本通常采用传统方法设计，比如：字典学习；（2）本节设计的码本用于信源压缩同时对抗语义通信系统中的语义噪声，而信道反馈码本用于传统通信系统中的信道压缩。

6.3.1 掩蔽矢量量化变分自编码器

信源中往往存在信息冗余，图像存在较大的空间冗余。一张图像可以被分割为若干块（Patch），而一张图像如果丢失一个或几个块，网络可基于对邻近块的信息、物体、场景的理解，将丢失的块恢复出来。因此，随机掩蔽局部块的策略是一种高效的方法，其可以用于创建一个具有挑战性的任务，要求模型全面地理解图像统计特性和语义信息，同时也减少图像信息冗余。此外，由于在图像的块中加入了语义噪声，掩蔽操作可以在一定程度上消除语义噪声的影响。

6.3.1.1 掩蔽矢量量化变分自编码器的架构

如图6.2所示，本章设计了带有ViT结构的掩蔽VQ-VAE模型。通过随机掩蔽输入图像的某些块，让训练好的网络模型能够恢复出这些图像块。掩蔽VQ-VAE属于变分自编码器的一种，但不同的是该网络结构可根据部分图像信息恢复出完整的图像。此外，不同于传统变分自编码器，这里采用非对称的编码-解码网络架构。

具体而言，编码器只需要处理一小部分未被掩蔽的块，而不需要处理被掩蔽的块，并将其映射为经过编码的图像特征进行传输，这大大减少了训练时间和内存消耗。该方法需要移除被掩蔽的块，并嵌入未被掩蔽的块及其在原始图像的位置信息，随后经过一系列ViT网络结构对其进行处理^[271]。与之对应的，如图6.2所示，解码网络的输入为：(1) 未被掩蔽的块编码特征和(2) 掩蔽标记组成的集合。每个掩蔽标记都是一个共享的通过网络训练得到的矢量，指示该位置存在待预测的被掩蔽的块。将位置信息嵌入到该集合的所有输入矢量中。如果没有该嵌入过程，掩码标记将丢失关于每个块在原始图像中的位置信息。此外，解码网络仅在预训练阶段执行图像重建任务，而编码网络用于提取输入图像的特征。因此，可以灵活地设计独立于编码网络的解码网络结构。与通信系统中传统的自编码器相比，掩蔽VQ-VAE具有以下优点：

- 采用这种不对称的结构设计，编码网络只处理未被掩蔽的块，轻量级解码网络根据经过编码的特征和掩蔽标记重建图像。这样可以显著降低计算复杂度和训练时间。
- 预先训练的掩蔽VQ-VAE可以用于不同的下游任务，例如图像分类，只需对轻量级解码网络的结构略加修改并用较短的时间微调掩蔽VQ-VAE的网络参数即可。
- 仅需传输未被掩蔽块对应的编码特征和掩蔽标记至接收端用于解码，可大大减少传输开销。
- 掩蔽操作可对抗语义噪声，因为在掩蔽部分块的同时，部分语义噪声也被掩蔽了。

6.3.1.2 对抗语义噪声的掩蔽策略

首先，将原始图像划分为一系列互相不重叠的图像块。随后，对块进行采样，保留一部分块，掩蔽并删除剩余的块。高掩蔽比，即掩蔽块的数量占所有块数量的比例较高，能够在很大程度上消除图像的空间信息冗余。文献[272]中的“随机采样”策略按照均匀分布对块进行随机掩蔽，即每个块的掩蔽概率相同。然而，语义噪声并不是随机出现的，它往往更频繁地出现在与图像中的目标主体相关的块中。因此，为了进一步降低语义噪声对系统的影响，根据语义噪声的统计特性，提高受语义噪声影响严重块的掩蔽概率。

6.3.2 离散码本设计

本章节旨在为经过编码网络编码的特征向量设计一个离散码本，并用码本中的基向量表征编码后的特征。考虑重要的任务相关特征，而忽略带有噪声和难以察觉细节的任务无

关特征。具体来说，将这些基向量设为训练参数，并将它们与编码网络和解码网络的参数一起训练。编码网络输出连续的编码特征，然后将这些连续特征映射为离散码本中对应的基向量，并用其在码本中的标号表征。该设计有如下优点：

- 训练码本较为容易，且基于码本的输出编码特征方差较小，这使得语义通信系统更稳定。
- 基于离散码本的编码特征表示可以对抗语义噪声。
- 发送端仅仅需要传输经过编码的特征对应码本中基向量的标号，显著降低了通信传输开销。

6.3.2.1 针对编码特征的离散码本设计

如图6.2所示，记针对经过网络编码特征的离散码本为 $\mathcal{E} \triangleq [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_J] \in \mathbb{R}^{J \times D}$ 。该码本包含 J 个基向量 $\{\mathbf{e}_j \in \mathbb{R}^D, j \in 1, 2, \dots, J\}$ ， D 是每个基向量 \mathbf{e}_j 的维度。掩蔽VQ-VAE模型的输入为一张图像 \mathbf{s} ，经过编码网络产生一系列编码特征向量 $\mathbf{z}_e(\mathbf{s})$ 。随后，这些编码特征向量被映射为码本中的基向量 $\mathbf{z}_b(\mathbf{s})$ ，并采用最邻近查找的映射方案

$$\mathbf{z}_b(\mathbf{s}) = \arg \min_{\mathbf{e}_j} \|\mathbf{z}_e(\mathbf{s}) - \mathbf{e}_j\|_2, \forall \mathbf{e}_j, \quad (6.12)$$

为表述清楚起见，略去编码特征的标号。接着， $\mathbf{z}_b(\mathbf{s})$ 被传输到接收端，并输入解码网络。可以将这一步前向计算过程视为网络的一层，且该层具有特殊的非线性函数，将编码特征向量 $\mathbf{z}_e(\mathbf{s})$ 映射为码本中的基向量 $\mathbf{z}_b(\mathbf{s})$ 。这些码本中的基向量 $\{\mathbf{e}_j, \forall j\}$ 被视为网络训练参数，和编码、解码网络的参数一起训练。然而，操作(6.12)是不可微分的。因此，在反向传播阶段，用直通式估计器近似梯度^[273]，即将梯度从解码网络的输入端 $\mathbf{z}_b(\mathbf{s})$ 拷贝到编码网络的输出端 $\mathbf{z}_e(\mathbf{s})$ 。总而言之，在前向传播阶段，码本中距离当前编码特征向量最近的基向量 $\mathbf{z}_b(\mathbf{s})$ 被传输到接收端，并输入解码网络，而在反向传播阶段，解码网络输入端的梯度 $\nabla_{\mathbf{z}_b(\mathbf{s})} \mathcal{L}_c$ 不做改动，直接拷贝到编码网络的输出端。注意到该网络的梯度包含以最小化损失函数为目标，来训练编码网络的有用信息，并可以推动编码网络的输出 $\mathbf{z}_e(\mathbf{s})$ ，更精确高效地进行离散化，从而实现更好的性能。

6.3.2.2 可微分的损失函数

本章节设计一个可微分的损失函数高效地联合训练编码、解码网络和码本，涉及网络各模块的参数

$$\mathcal{L}_c(\mathbf{s}, \mathbf{z}; \boldsymbol{\theta}, \mathbf{e}_j) = \|\hat{\mathbf{s}} - \mathbf{z}\|_2^2 + \|\text{ng}[\mathbf{z}_e(\mathbf{s})] - \mathbf{e}_j\|_2^2 + \beta \|\mathbf{z}_e(\mathbf{s}) - \text{ng}[\mathbf{e}_j]\|_2^2, \quad (6.13)$$

其中 \mathbf{s} , $\hat{\mathbf{s}}$ 和 \mathbf{z} 分别表示网络的输入、输出和该输入样本对应的真实标签。 $\boldsymbol{\theta}$ 表示编码、解码网络的训练参数, β 为给定的超参数。带有符号 $\text{ng}[\mathbf{z}_e(\mathbf{s})]$ 的项表示没有梯度传递到 $\mathbf{z}_e(\mathbf{s})$, 即其对应的梯度为零, 从而有效限制 $\mathbf{z}_e(\mathbf{s})$ 为不更新的常数。第一项为损失函数, 训练编码、解码网络的参数。由于直通式估计器的设计将梯度从 $\mathbf{z}_b(\mathbf{s})$ 拷贝到 $\mathbf{z}_e(\mathbf{s})$, 码本中的基向量 $\{\mathbf{e}_j, \forall j\}$ 不能通过 $\|\hat{\mathbf{s}} - \mathbf{z}\|_2^2$ 这一项得到反传的梯度。因此, 为了训练基向量, 采用 l_2 范数, 使得基向量尽可能逼近编码网络输出的编码向量 $\mathbf{z}_e(\mathbf{s})$, 即损失函数表达式(6.13)的第二项。由于编码特征向量空间的维度是无穷的, 码本理论上有可能变得无穷大, 导致网络训练过程难以收敛, 尤其当码本中的基向量 $\{\mathbf{e}_j, \forall j\}$ 的训练更新速度慢于编码网络参数时。为了解决该问题, 设计了损失函数(6.13)中的第三项。总而言之, 解码网络仅通过损失函数的第一项来训练, 编码网络通过损失函数的第一项和第三项来训练, 码本中的基向量通过损失函数的第二项来训练。

6.3.3 码本鲁棒性

6.3.3.1 语义相似度

码本中两个基向量 \mathbf{e}_1 和 \mathbf{e}_2 的语义相似度可以被定义为向量的乘积 $\mathbf{e}_1^T \cdot \mathbf{e}_2$, 余弦距离 $\frac{\mathbf{e}_1^T \cdot \mathbf{e}_2}{\|\mathbf{e}_1\| \cdot \|\mathbf{e}_2\|}$, 或者向量差的 l_2 范数 $\|\mathbf{e}_1 - \mathbf{e}_2\|$ 等。高语义相似度表示这两个基向量包含相似的语义信息。本章选择余弦距离作为两个基向量的语义相似度, 并计算码本中任意两个基向量的语义相似度。记包含所有归一化基向量的归一化码本为

$$\mathbf{E} \triangleq \left[\frac{\mathbf{e}_1}{\|\mathbf{e}_1\|}, \frac{\mathbf{e}_2}{\|\mathbf{e}_2\|}, \dots, \frac{\mathbf{e}_J}{\|\mathbf{e}_J\|} \right]. \quad (6.14)$$

由此, 矩阵 $\mathbf{E}^T \mathbf{E}$ 的第 (i, j) 个元素表示基向量 \mathbf{e}_i 和 \mathbf{e}_j 的语义相似度。

6.3.3.2 鲁棒码本设计

旨在降低基向量之间语义相似度(即增加基向量之间余弦距离)的损失函数被设计为

$$\mathcal{L}_s = \|\mathbf{E}^T \mathbf{E}\|_2. \quad (6.15)$$

基于语义相似度, 将 $\|\mathbf{E}^T \mathbf{E}\|_2$ 这一项加入损失函数(6.13), 希望尽可能使得码本中的基向量相互正交, 即尽可能降低基向量之间的语义相似度, 增大基向量之间的余弦距离。

6.3.3.3 码本鲁棒性分析

离散码本可以有效地降低语义噪声的影响。如图6.3所示, 语义噪声的存在使得提取的特征矢量向某一特定的方向移动。只要不远离不受语义噪声影响的原始提取特征对应的基

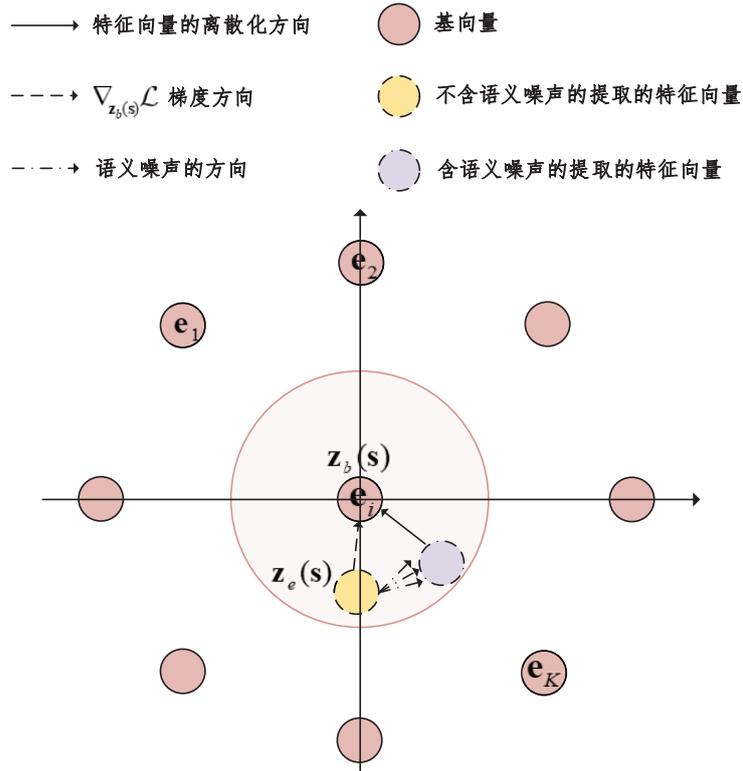


图 6.3 对抗语义噪声的离散码本设计

向量，该离散码本设计就可以消除语义噪声的影响。因此，增加两个基向量之间的距离可以提高码本对语义噪声的鲁棒性。基于语义相似度，提出的损失函数可以增加两个基向量之间的距离，并使它们尽可能相互正交。正交的基向量具有如下两个优点：（1）两个正交基向量之间的距离较大；（2）当基向量相互正交时，用于表示编码后的特征向量对应的特征空间所需的基向量数量最少，即可以用较低维度的码本表征编码特征向量，从而进一步降低网络训练开销和通信传输开销。

6.3.4 基于码本的高效传输通信系统

6.3.4.1 基于码本的星座点设计

现有的语义通信工作主要是将信源数据直接映射为传输符号，并假设通信系统为全分辨率星座图^[231-233]，即星座点可以出现在星座图的任何位置。然而，对于实际通信系统而言，全分辨率星座图是极其复杂且难以实现的。因此，为了使得系统更符合实际，需要限制星座图中星座点的数量。为了解决上述问题，提出一种基于离散码本的通信系统。该系统将编码特征映射为码本基向量对应的标号，再映射到现有的有限星座图中，从而能够更好地适配当前数字通信系统。一般情况下，离散化会导致数据传输精度的降低。

而VQ-VAE是一种有效的矢量量化方案,实现了较好的性能^[273]。由于码本与通信系统中的编码、解码网络联合训练,VQ-VAE的量化性能优于传统的均匀量化。

6.3.4.2 基于码本的高效传输语义通信系统

假设发送端和接收端共享一个码本 \mathcal{E} ,该码本由基向量 $\{\mathbf{e}_j, \forall j\}$ 组成,这些基向量在训练阶段完成后是固定的。因此,对于编码网络输出的每个编码特征,发送端只需要发送其对应基向量在码本中的标号,这显著降低了通信传输开销。如图6.2所示,在传输阶段,基向量的标号首先被映射成二进制比特。随后,这些二进制比特被映射为传输符号,并通过无线信道 \mathbf{H} 传输。接收端将接收到的符号映射回基向量在码本中的标号,并在码本中找到对应的基向量。这些基向量是解码网络的输入,用于后续进一步处理,并完成任务。信道和高斯白噪声可被视为整个系统中网络的一层,并与编码网络、解码网络的参数,以及码本联合训练。

6.4 特征重要性模块和训练方法

本章节通过设计FIM使语义通信系统更鲁棒和高效。此外,FIM针对信噪比进行了设计,确保所提出的系统可以在不同的信噪比水平下运行并达到良好的效果。此外,提出了一种新的损失函数和训练方法来训练FIM。

6.4.1 噪声相关的特征抑制

FIM根据不同样本动态地学习特征重要性,并将其纳入训练阶段来训练深度学习网络模型。该模型可抑制那些与噪声相关,而与任务无关的不重要的特征。

6.4.1.1 噪声相关的特征

不同的特征从不同的方面描述图像,且特征与系统针对语义噪声的鲁棒性之间存在很强的联系,而不同特征的鲁棒性不同。不同于现有的研究假设图像每个特征是同等重要的,本章关注特征、噪声和图像本身之间的关系,并赋予图像不同特征不同的重要性。直观来说,不同特征对任务结果的贡献不同,对语义噪声的鲁棒性也不同。语义通信系统应学习到不同特征各自的重要性,从而更好地理解输入图像背后的语义信息。在此基础上,发送端可以发送与任务相关且与噪声无关的重要特征,这大大提高了系统的鲁棒性,同时降低了通信传输开销。

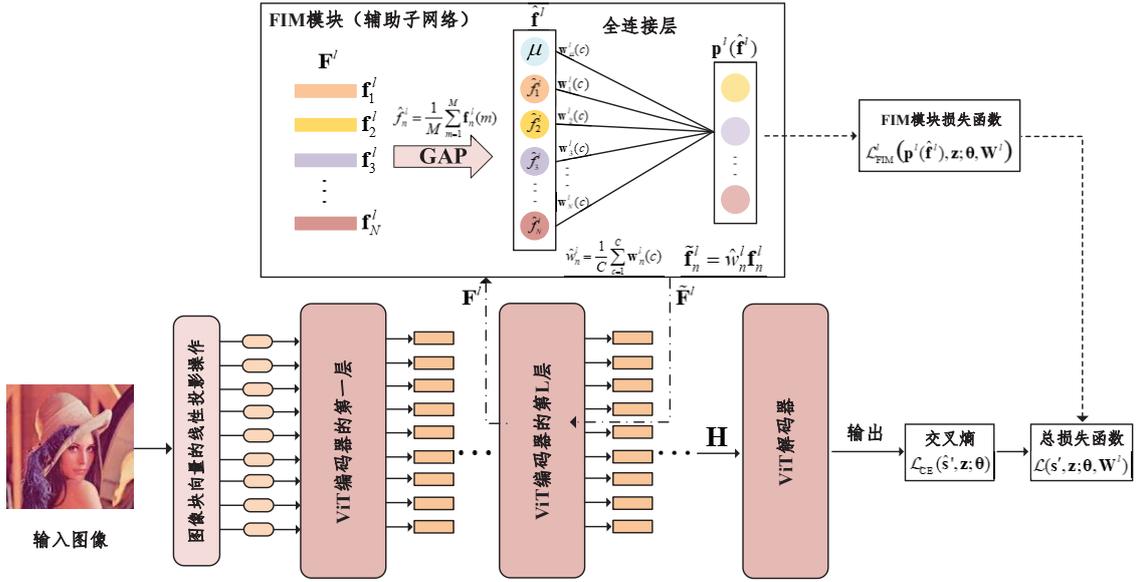


图 6.4 包含FIM模块的掩蔽VQ-VAE架构

6.4.1.2 特征激活

从特征激活的角度可以观察到语义噪声的两个特点：（1）带有语义噪声的样本被激活特征的幅值高于不带语义噪声的样本；（2）带有语义噪声的样本中某些噪声相关特征被激活得更为均匀和频繁。对抗训练可以解决第一个问题，即激活特征的高幅值特性。换句话说，即使经过对抗训练，一些和任务无关的低贡献特征，它们不会被不带有语义噪声的样本激活，但会被带有语义噪声的样本激活。这在一定程度上解释了为什么对抗训练是有效的，但其效果却难以令人满意。因此，本章节期望通过设计FIM，训练能为不同特征分配不同的重要性的模型，从而有效抑制与任务无关和与噪声相关的特征被语义噪声激活。

6.4.2 具有动态信噪比的特征重要性模块

6.4.2.1 特征重要性模块的结构

将由图像特征组成的特征矩阵 $\mathbf{F}^l \triangleq [\mathbf{f}_1^l, \mathbf{f}_2^l, \dots, \mathbf{f}_N^l] \in \mathbb{R}^{M \times N}$ ，作为网络第 l 层的输出，其中 M 和 N 分别表示特征向量的维度和数量。如图6.4所示，首先，针对特征矩阵 \mathbf{F}^l 采取全局平均池化操作，得到特征激活向量 $\hat{\mathbf{f}}^l \triangleq [\mu, \hat{f}_1^l, \hat{f}_2^l, \dots, \hat{f}_N^l] \in \mathbb{R}^{N+1}$ ，其中 μ 表示信噪比。对于第 n 个特征，进行如下处理

$$\hat{f}_n^l = \frac{1}{M} \sum_{m=1}^M \mathbf{f}_n^l(m). \quad (6.16)$$

注意到全局平均池化操作通过平均特征向量中每个元素的值来提取全局特征的信息^[233]。此外，不同的信噪比水平往往会导致特征的重要性不同。因此，为了确保所提出的语义通信系统可以在较为宽泛的信噪比水平范围运行，信噪比 μ 被设计为FIM输入 $\hat{\mathbf{f}}^l$ 的一部分。

特征激活向量 $\hat{\mathbf{f}}^l$ 传递到一个具有全连接结构和ReLU非线性函数的辅助神经网络。语义通信系统解码网络的输出为 $\mathbf{z} \in \mathbb{R}^C$ ，比如：图像分类问题则为所属类别标签，图像重建问题则为重建的图像。随后，该辅助神经网络的训练参数可以写成 $\mathbf{W}^l \triangleq [\mathbf{w}_\mu^l, \mathbf{w}_1^l, \mathbf{w}_2^l, \dots, \mathbf{w}_N^l] \in \mathbb{R}^{C \times (N+1)}$ ，表示每个特征相对于解码网络输出 \mathbf{z} 的重要性。参数 \mathbf{W}^l 将被用于重新加权特征矩阵 \mathbf{F}^l 。记权重变量 $\hat{\mathbf{w}}^l \triangleq [\hat{w}_1^l, \hat{w}_2^l, \dots, \hat{w}_N^l] \in \mathbb{R}^N$ ，其中对于第 n 个特征

$$\hat{w}_n^l = \frac{1}{C} \sum_{c=1}^C \mathbf{w}_n^l(c), \quad (6.17)$$

该权重变量被用于度量第 l 层第 n 个特征和输出 \mathbf{z} 相关的重要性权重。采用一个Softmax层将该权重变量缩放到 $[0, 1]$ 范围内。随后，用该权重对特征矩阵 \mathbf{F}^l 中的特征 $\{\mathbf{f}_n^l, \forall n\}$ 进行加权，得到 $\tilde{\mathbf{f}}_n^l = \hat{w}_n^l \mathbf{f}_n^l$ 。加权后的特征矩阵 $\tilde{\mathbf{F}}^l \triangleq [\tilde{\mathbf{f}}_1^l, \tilde{\mathbf{f}}_2^l, \dots, \tilde{\mathbf{f}}_N^l] \in \mathbb{R}^{M \times N}$ ，将通过前向传播传递到下一层网络。通过该方式，获得了特征之间的关系及特征的重要性，并为不同特征赋予了不同的权重，以增强或抑制其与下一层的连接强度，从而影响解码网络输出的结果。

6.4.2.2 具有标签信息的特征重要性模块

此外，为了充分利用标签信息，本小节针对图像分类任务的FIM进行了轻微的修改，并将标签信息考虑其中以提升性能。具体而言，在训练阶段，利用真实标签作为确定特征重要性的指标。在推理阶段，由于真实标签难以获得，将与预测类别相关联的权重作为特征重要性的指标。随后，使用特征重要性来重新加权原始特征矩阵为 $\tilde{\mathbf{f}}_n^l = \mathbf{w}_n^l(y) \mathbf{f}_n^l$ ，其中 y 表示训练阶段的真实标签，以及推理阶段预测的类别，由此可得重新加权后的特征矩阵 $\tilde{\mathbf{F}}^l \triangleq [\tilde{\mathbf{f}}_1^l, \tilde{\mathbf{f}}_2^l, \dots, \tilde{\mathbf{f}}_N^l] \in \mathbb{R}^{M \times N}$ 。接着，重新加权后的特征矩阵 $\tilde{\mathbf{F}}^l$ 通过前向传播传递到下一层网络。由此，特征重要性的学习过程考虑了标签信息。

6.4.3 模型训练

6.4.3.1 特征重要性模块的损失函数

将提出的FIM嵌入到掩蔽VQ-VAE网络的某些层中，将其视为辅助网络。可采用对抗训练方法，将其与掩蔽VQ-VAE网络的参数一起训练。接下来，设计损失函数来同时训练

掩蔽VQ-VAE网络和FIM。以在第 l 层后插入的一个FIM为例，损失函数设计为

$$\mathcal{L}_{\text{FIM}}^l(\mathbf{p}^l(\hat{\mathbf{f}}^l), \mathbf{z}; \boldsymbol{\theta}, \mathbf{W}^l) \triangleq \mathcal{L}_{\text{CE}}(\mathbf{p}^l(\hat{\mathbf{f}}^l), \mathbf{z}; \boldsymbol{\theta}, \mathbf{W}^l), \quad (6.18)$$

其中 $\mathbf{p}^l(\hat{\mathbf{f}}^l) \triangleq \text{ReLU}(\mathbf{W}^l \hat{\mathbf{f}}^l)$ 为FIM的输出， $\boldsymbol{\theta}$ 表示掩蔽VQ-VAE网络的训练参数， \mathbf{W}^l 为FIM的训练参数， $\mathcal{L}_{\text{CE}}(\mathbf{p}^l(\hat{\mathbf{f}}^l), \mathbf{z}; \boldsymbol{\theta}, \mathbf{W}^l)$ 为关于 $\mathbf{p}^l(\hat{\mathbf{f}}^l)$ 和标签 \mathbf{z} 的交叉熵损失函数。FIM的损失函数设计为表达式(6.18)，因为当 $\mathbf{p}^l(\hat{\mathbf{f}}^l)$ 更逼近标签 \mathbf{z} 时，表明FIM被训练得更好，即：由于噪声相关的特征以及和任务无关的特征被充分抑制，从而导致网络输出更逼近标签，系统实现了更好的性能。

上述设计很容易直接扩展到多FIM。包含FIM的对抗训练整体损失函数可以被写成

$$\mathcal{L}_f(\mathbf{s}', \mathbf{z}; \boldsymbol{\theta}, \mathbf{W}^l) = \mathcal{L}_{\text{CE}}(\hat{\mathbf{s}}', \mathbf{z}; \boldsymbol{\theta}) + \frac{\gamma}{L} \sum_{l=1}^L \mathcal{L}_{\text{FIM}}^l(\mathbf{p}^l(\hat{\mathbf{f}}^l), \mathbf{z}; \boldsymbol{\theta}, \mathbf{W}^l), \quad (6.19)$$

其中 \mathbf{s}' 表示被用于对抗训练的带有语义噪声的样本， $\hat{\mathbf{s}}'$ 为解码网络的输出， $\mathcal{L}_{\text{CE}}(\hat{\mathbf{s}}', \mathbf{z}; \boldsymbol{\theta})$ 表示关于 $\hat{\mathbf{s}}'$ 和 \mathbf{z} 的交叉熵损失函数， L 表示网络层数， γ 为用于控制FIM强度的可调超参数。

表 6.3 鲁棒语义通信系统中模型的训练方法

算法 6.3 鲁棒语义通信系统中模型的训练方法

- 1: **输入：** 包含输入图像 \mathbf{s}_i 及其对应标签 \mathbf{z}_i 的训练数据集，产生的信道样本 \mathbf{H}_i ，以及模型训练的周期数 N ；
 - 2: **输出：** 离散码本 \mathcal{E} ，包含FIM、编码网络和解码网络的鲁棒语义通信系统模型；
 - 3: **for** $n = 1 : N$ **do**
 - 4: 根据提出的掩蔽策略，掩蔽输入图像的一部分；
 - 5: 根据损失函数(6.13)计算 \mathcal{L}_c ，用于联合训练编码网络、解码网络和离散码本 \mathcal{E} ；
 - 6: 根据损失函数(6.19)计算 \mathcal{L}_f ，用于训练FIM；
 - 7: 根据损失函数(6.15)计算 \mathcal{L}_s ，用于微调编码网络和解码网络；
 - 8: 计算 \mathcal{L}_c ， \mathcal{L}_f 和 \mathcal{L}_s 的和，执行表6.2中的算法用于对抗训练。
 - 9: **end for**
-

6.4.3.2 模型的训练方案

提出了一种联合训练前述鲁棒语义通信系统中所有模块的方案，其中详细的训练细节请见算法表6.3。

6.5 仿真结果

本节通过数值仿真来验证鲁棒语义通信系统的性能。

表 6.4 模型的网络结构

	层名字	维度	激活函数
发送端	8× Transformer编码器	768 (12 heads)	线性函数
	全连接层	256	Sigmoid函数
	码本	128	无
	FIM	196	ReLU函数
信道	信道层	N_r	无
接收端	FIM	196	ReLU函数
	码本	128	无
	全连接层	768	Sigmoid函数
	4× Transformer编码器	768 (12 heads)	线性函数
	全连接层	10	ReLU函数

6.5.1 仿真设置

考虑发送端和接收端分别配置 $N_t = 4$ 根发射天线和 $N_r = 2$ 根接收天线的场景。在MIMO信道下，将本章所提出的掩蔽VQ-VAE与传统信源编码和信道编码方案进行比较^[1]。采用CIFAR-10（包含10个类别，60,000张图像）作为图像分类任务的数据集，Cars196（包含196个类别，16,185张图像）作为图像检索任务的数据集，ImageNet（包含14,197,122张图像）作为图像重建任务的数据集。数据集中平均每张JPEG图像的大小为5,108字节，每张图像的图像块数目为 14×14 。掩蔽VQ-VAE的掩蔽率为0.5，码本大小为256，其中每个特征的标号（索引）需要用8比特来传输。采用16-QAM调制方式和1/2码率的LDPC信道编码方式。生成语义噪声的迭代次数设为 $K = 5$ ，语义噪声功率为 $\epsilon = 0.016$ 。所提出模型的网络结构如表6.4所示。具体而言，“码本层”表示码本中的基向量被设置为网络的训练参数，“码本层”的“维度”表示基向量的数量。此外，其他层的“维度”表示该层的输出维度。若要将该模型用于处理不同的下游任务，例如：模型训练时的任务是图像重建任务，后续需要用于处理图像分类任务，则只需改变网络最后一层的维度，同时微调经过预训练的模型。本节比较了以下方案的性能：

- VQ-VAE+FIM+AT：本章提出的掩蔽VQ-VAE模型，且包含有FIM和对抗训练；
- VQ-VAE+AT：本章提出的掩蔽VQ-VAE模型，仅包含对抗训练；

表 6.5 完成不同任务传输一张图像需要的符号数

方案	JPEG+LDPC	掩蔽VQ-VAE (Patch= 16)	掩蔽VQ-VAE (Patch= 8)
图像分类	53,760	196	784
图像检索	249,900	490	1,960
图像重建	143,350	490	1,960

- VQ-VAE: 本章提出的掩蔽VQ-VAE模型;
- JSCC+AT: 文献[231]提出的针对不同任务的具有ViT架构的修正JSCC方案, 并添加了对抗训练;
- JSCC: 文献[231]提出的针对不同任务的具有ViT架构的修正JSCC方案;
- JPEG+LDPC+AT: 采用JPEG图像信源编码、LDPC信道编码的传统方案, 并采用ViT作为图像分类器进行对抗训练;
- JPEG+LDPC: 采用JPEG图像信源编码、LDPC信道编码的传统方案。

本章提出了两种语义噪声模型: (1) 添加在发送端的样本依赖的语义噪声; (2) 添加在接收端的非样本依赖的语义噪声。其中语义噪声模型 (1) 对语义通信系统有更为严重的影响。因此, 除非特殊说明, 本节都采用语义噪声模型 (1), 语义噪声模型 (2) 可以类似添加和处理。考虑以下的任务场景并生成相应的语义噪声:

- 对于图像分类任务, 添加了语义噪声的图像在视觉上与干净样本相似, 但该图像大概率将被模型错误分类;
- 对于图像检索任务, 添加了语义噪声的图像在视觉上与干净样本相似, 但该微小的语义噪声会导致图像检索结果出错;
- 对于图像重建任务, 语义噪声将导致图像重建失败, 例如: 重建图像中一些关键目标和信息丢失或模糊化。

6.5.2 传输开销分析

表6.5给出了传统JPEG+LDPC和本章提出的掩蔽VQ-VAE的传输开销, 即完成不同任务 (对应不同数据集), 传输一张图像所需要的符号数。其中, “Patch= 16”表示掩蔽VQ-VAE在预处理阶段的图像压缩率为16, 即一个 $16 \times 16 \times 3$ 的图像块会被压缩成一个标量。因此, 该值越大表示越高的图像压缩率和越低的传输开销。JPEG+LDPC方案对每

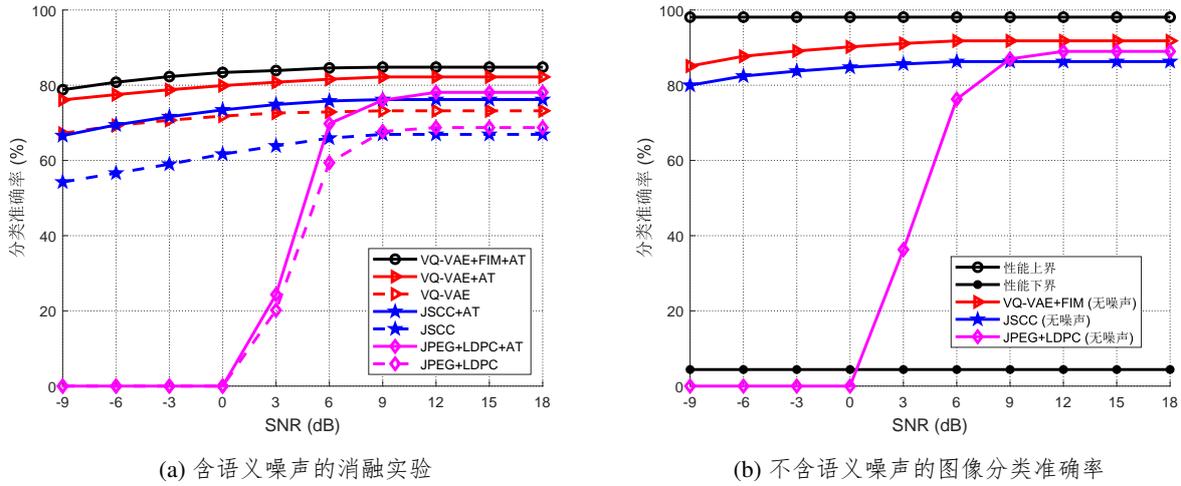


图 6.5 图像分类准确率随SNR的变化

张图像的传输符号数为： $\frac{LHC_n P_b R_c}{C_r B_s}$ ，其中 L 和 H 分别表示图像的长度和宽度， C_n 为通道数， P_b 表示每个图像像素所需的传输比特数， R_c 为码率， C_r 表示压缩比， B_s 表示一个符号对应的比特数，取决于调制方式。以图像分类为例：需要 $\frac{224 \times 224 \times 3 \times 8 \times 2}{11.2 \times 4} = 53,760$ 符号/图像。此外，所提出的掩蔽VQ-VAE对每张图像的传输符号数为： $\frac{LH J_c M_r}{P_a^2 B_s}$ ，其中， J_c 表示在码本中传输一个基向量对应的索引标号所需的比特数， M_r 是图像掩蔽比例， P_a 表示图像块的大小。以用于图像分类任务的掩蔽VQ-VAE（Patch= 16）为例： $\frac{224 \times 224 \times 8 \times 0.5}{16^2 \times 4} = 196$ 符号/图像。因此，所提出的掩蔽VQ-VAE仅需要传输传统方案JPEG+LDPC 所需符号的 $196/53760 = 0.36\%$ 。

6.5.3 图像分类准确率

图6.5描述了图像分类准确率与SNR的关系。在SNR= -3 ~ 12dB下训练模型，并在SNR= -9 ~ 18dB范围内对其进行测试。从图6.5(a)可以看出，所有方案的分类准确率都随着SNR的增加而提高。所提出的掩蔽VQ-VAE+FIM+AT方案显著优于JSCC+AT、JSCC、JPEG+LDPC+AT和JPEG+LDPC方案，并实现了最佳性能。此外，掩蔽VQ-VAE+FIM+AT方案实现的性能优于掩蔽VQ-VAE+AT和掩蔽VQ-VAE方案。这验证了本章的设计中每个模块的有效性，包括掩蔽VQ-VAE、FIM和对抗训练。另外，所提出的方案在低SNR场景下均显著优于传统的JPEG+LDPC方案。这主要是因为，在低SNR场景下系统BER往往较高，而在提出的方案中，发送端传输的是所提取任务相关特征对应码本中的索引标号，因此具有很强的鲁棒性，即使部分传输比特出错也不会影响系统的图像分类准确率。

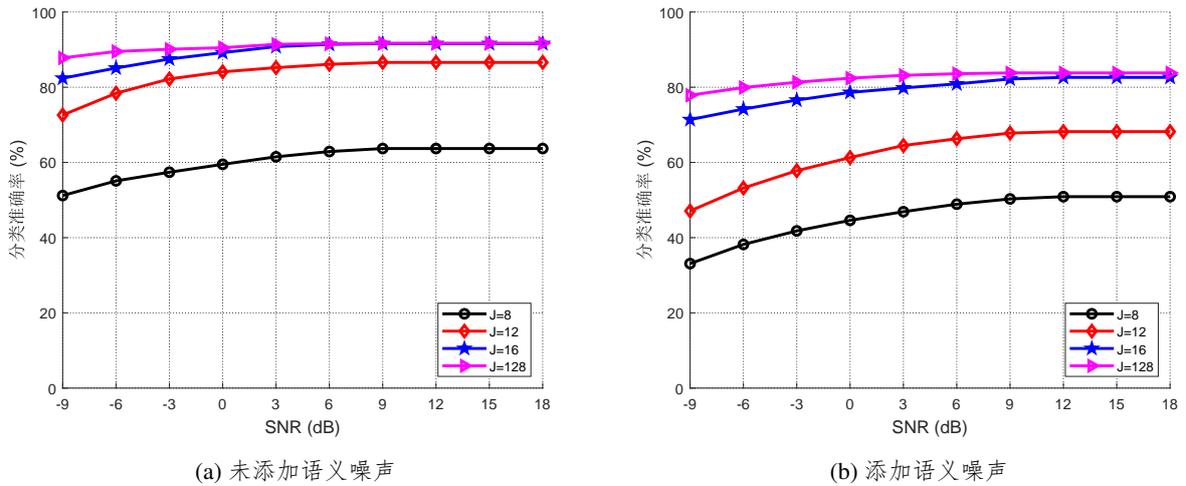


图 6.6 不同码本大小 J 的图像分类准确率随 SNR 的变化

图6.5(b)描述了没有添加语义噪声方案的分类准确率作为对比，图中“无噪声”表示没有添加语义噪声的方案。性能上界是由JSCC在不受信道噪声和语义噪声影响的场景下实现的分类准确率，而性能下界则是由添加了信道噪声和语义噪声的JSCC实现的。从图中可以看出，所提出的掩蔽VQ-VAE+FIM（无噪声）接近上界性能，且显著优于JSCC（无噪声）和JPEG+LDPC（无噪声）。此外，掩蔽VQ-VAE+FIM+AT方案接近未添加语义噪声的模型实现的性能，这表明所提出的模型可以通过降低语义噪声的影响来有效提高系统的鲁棒性。

图6.6(a)和图6.6(b)分别描述了未添加语义噪声和添加了语义噪声的场景下，采用不同码本大小 J 的模型，其实现的图像分类准确率与 SNR 的关系。由图可见，分类准确率随 SNR 的增加而提高。更大的码本尺寸对语义噪声具有更强的鲁棒性，并且对整个图像数据集具有更强的表示能力，这提高了分类准确率。此外，当码本尺寸 $J \geq 16$ ，则足以对图像数据集进行精确分类。

图6.7展示了图像分类准确率与语义噪声的功率 ϵ 之间的关系。性能上界和性能下界分别由无语义噪声下的JSCC和最大语义噪声功率下的JSCC实现。由图可知，所有方案实现的图像分类准确率都随着 ϵ 的增加而降低。本章提出的掩蔽VQ-VAE+FIM+AT方案显著优于基线方案，并实现了最佳性能，尤其当 ϵ 较大时性能增益更为明显。这验证了所提出模型在对抗语义噪声时的优势。

图6.8描述了在不同语义噪声功率 ϵ 下，提出的掩蔽VQ-VAE+FIM+AT方案的图像分类准确率与图像保留率之间的关系。其中保留率=1-掩蔽率，表示图像的保留部分（未被掩蔽部分）占整个图像的比例。可以看出，掩蔽VQ-VAE+FIM+AT实现的图像分类准确率随

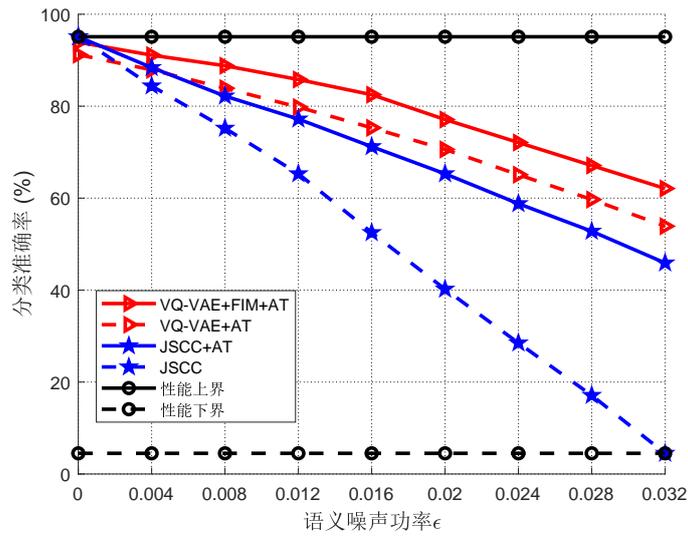


图 6.7 图像分类准确率随语义噪声功率 ϵ 的变化

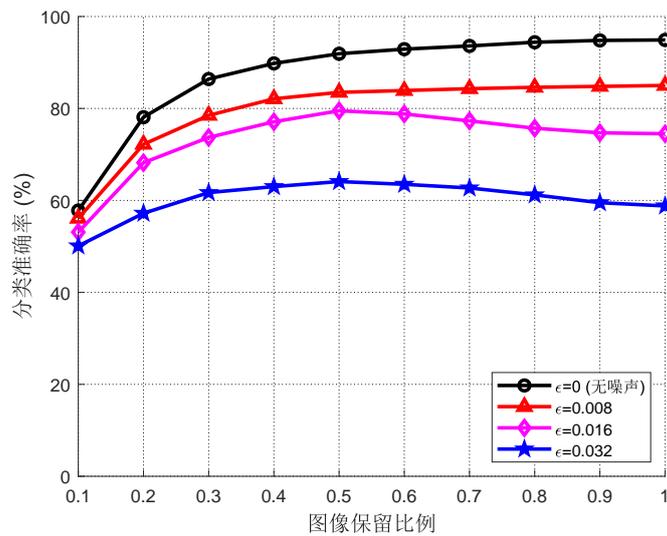


图 6.8 不同语义噪声功率 ϵ 下的图像分类准确率随图像保留率的变化

着 ϵ 增加而降低。此外，当 ϵ 较小时，例如 $\epsilon = 0.008$ ，分类准确率随着保留率的增加而提升，因为较高的保留率保留了图像的更多语义信息。而当 ϵ 较大时，例如 $\epsilon = 0.016$ ，分类准确率随着保留率的增加先提升后降低。这主要是因为尽管较高的保留率保留了图像更多的语义信息，但同时也保留了更多的语义噪声。因此，保留率0.5实现了语义信息和语义噪声之间的良好折中，达到了最高的分类准确率。

图6.9展示了不同方案的图像分类准确率与信号-语义噪声比例的关系，其中语义噪声为在接收端添加的非样本依赖的语义噪声。由图可见，所有方案实现的分类准确率都随着信号-语义噪声比例的增加而提升，而掩蔽VQ-VAE+FIM+AT实现的性能显著优于基线方

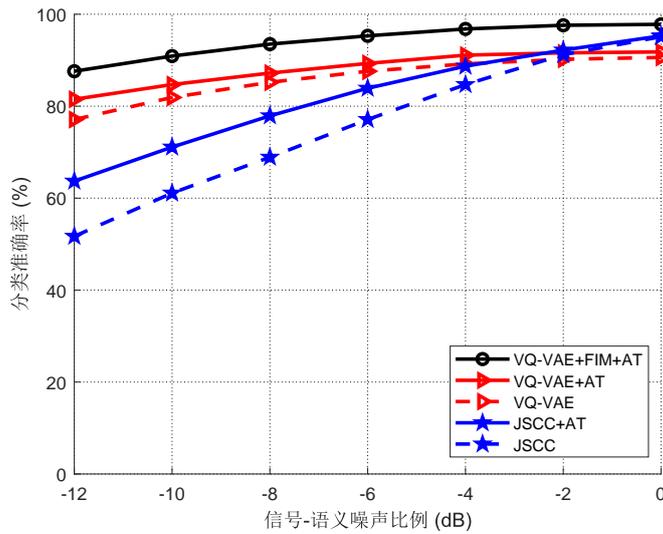


图 6.9 图像分类准确率随接收端信号-语义噪声比例的变化

表 6.6 白盒和黑盒语义噪声下的图像分类准确率

攻击模型	Proposed			Transformer		ResNet	
	P-P	T-P	R-P	T-T	R-T	R-R	T-R
分类准确率	68.7%	82.3%	85.3%	9.5%	75.3%	8.2%	67.2%

案，并实现了最佳性能。这说明相比于JSCC，所提出的模型对非样本依赖的语义噪声具有更强的鲁棒性。与样本依赖的语义噪声相比，非样本依赖的语义噪声对图像分类准确率的影响相对较小，对抗训练对其的效果不显著。

6.5.4 白盒和黑盒语义噪声

一般认为，攻击者需要知道收发端模型的网络结构、训练参数和损失函数等模型所有的信息，才能攻击语义通信系统。事实上，对于一个语义通信系统而言，这几乎是一种最坏的情况，并且通过该种方式产生的语义噪声会显著影响系统的性能。将这种攻击者已知所有收发端模型的信息时产生的语义噪声称为白盒语义噪声。然而，当攻击者只知道部分模型信息时，也可以产生语义噪声。特别地，当攻击者只知道网络结构等部分模型信息时，可采用针对其他具有相似网络结构的模型所产生的语义噪声，来攻击当前的语义通信系统。该语义噪声称为黑盒语义噪声。

在表6.6中，首先为所提出的模型（即掩蔽VQ-VAE+FIM+AT）、基于Transformer的模型和基于ResNet的模型生成语义噪声。随后，使用这些不同类型的语义噪声来评估不同模型的鲁棒性。具体而言，“语义噪声类型”这一行中的“T-P”指的是采用基于Transformer的模

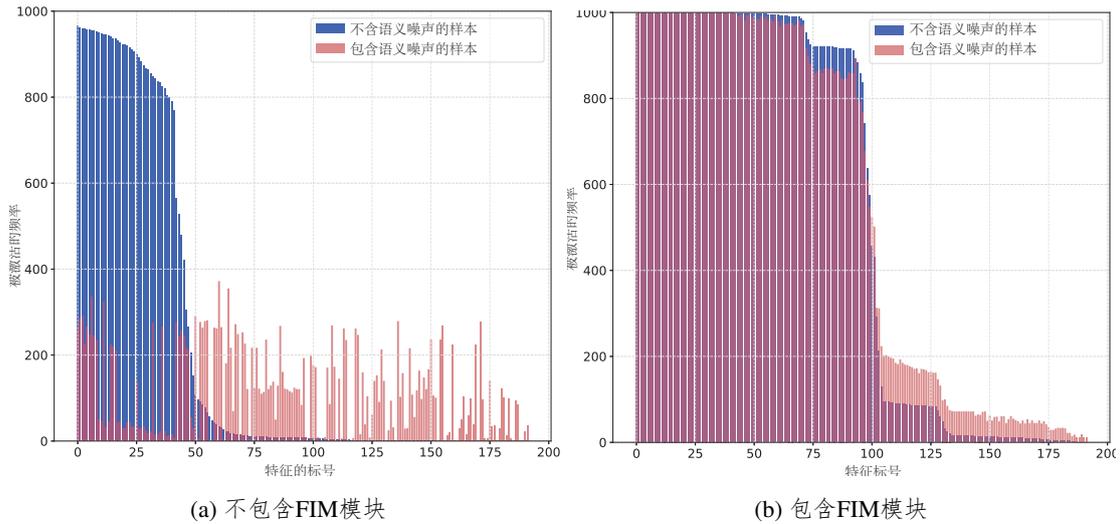


图 6.10 不同特征的激活频率

型产生的语义噪声来攻击所提出的模型，即黑盒语义噪声。而“P-P”则表示采用所提出的模型产生的语义噪声来攻击所提出的模型，即白盒语义噪声。可以看到，白盒语义噪声对模型的影响比黑盒噪声更严重。根据基于Transformer的模型产生的黑盒语义噪声会影响所提出模型的性能，且比基于ResNet的模型产生的黑盒语义噪声更具有攻击性，造成的影响更严重。这主要是因为所提出的模型也是基于Transformer的结构设计的。该结果表明，由一个更相似的模型产生的黑盒语义噪声往往更有效，因为具有相似架构的模型通常具有相似的参数和网络层结构。此外，所提出的模型比基于Transformer和基于ResNet的模型对白盒和黑盒语义噪声具有更强的鲁棒性。需要强调的是，本章的目标是设计一个针对语义噪声的鲁棒语义通信系统。而从表6.6可以看到，所提出的模型也能有效地防御黑盒语义噪声。

6.5.5 特征激活

图6.10(a)和图6.10(b)分别展示了不包含FIM的模型和包含FIM的模型，其对应的图像特征激活频率。当特征的激活值大于阈值，则将其判定为激活。从图6.10(a)可以看出，添加了语义噪声的样本，其特征被激活地更均匀，且它们倾向于频繁地激活那些很少被无噪声样本激活的特征，即特征125 – 200。这些低频特征与噪声相关，是对于图像分类任务不太重要的冗余特征激活。此外，语义噪声还会抑制某些被无噪声样本频繁激活的特征，即特征1 – 75。图6.10(b)说明FIM提高了一些任务相关重要特征的激活频率，例如特征75 – 100，并抑制带有语义噪声的样本去激活那些任务不相关的冗余特征，例如特征125 – 200。这一结果证明了所提出FIM的有效性，它可以抑制任务不相关的、噪声相关

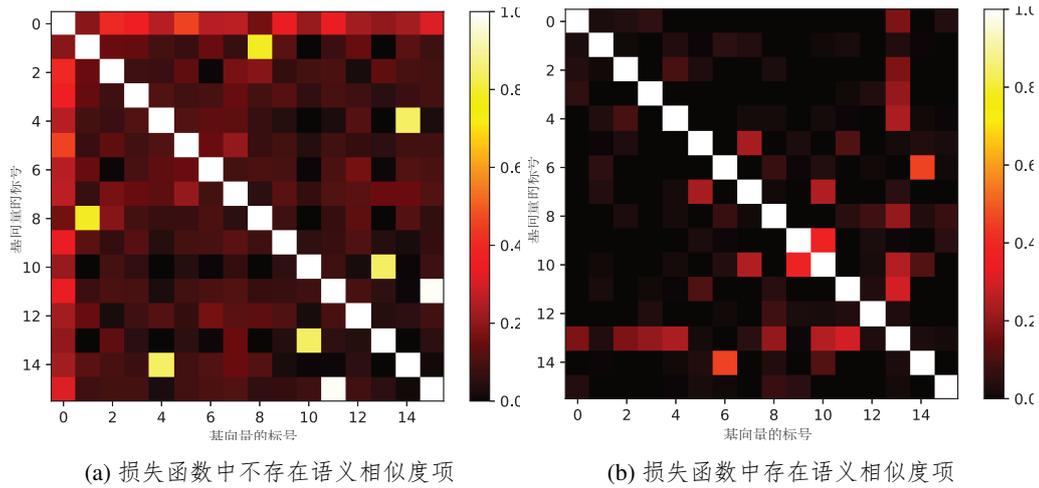


图 6.11 码本中基向量的语义相似性

的特征被激活。

6.5.6 语义相似度

图6.11(a)和图6.11(b)分别展示了在损失函数中包含和不包含语义相似度项的情况下，码本中基向量的语义相似度。图中颜色越亮表示对应的值越大，即表示两个基向量之间的语义相似度越高。可以看到，图6.11(a)中基向量之间的语义相似度普遍大于图6.11(b)。这证实了所提出损失函数的语义相似度项 $\|\mathbf{E}^T \mathbf{E}\|_2$ ，可以使码本中的基向量几乎相互正交，从而增强码本的鲁棒性和表示能力。

6.5.7 其他任务的性能

图6.12(a)描述了图像检索任务的recall@1性能与SNR的关系，其中recall@1表示图像检索任务在第一次查询时的成功率。性能上界由JPEG+LDPC在没有语义噪声的高SNR场景下实现，因为它实现了几乎无损的图像传输。由图可知，Patch 8的性能优于Patch 16，但Patch 16具有更高的图像压缩率，从而传输开销更低。此外，Patch 8显著优于传统的JPEG+LDPC方案，尤其是在低SNR场景下。这是因为在低SNR场景下，采用JPEG+LDPC不能正确解码所传输的图像，而所提出的方案同时对语义噪声和信道噪声都更为鲁棒，可以较好地传输图像进行解码。

图6.12(b)展示了语义噪声功率 $\epsilon = 0.012$ 时的图像重建质量。图中的左、中、右列分别表示原始图像、用Patch 16重建的图像和用Patch 8重建的图像。由图可知，较低的图像压缩率往往具有更好的重建质量，且Patch 8针对添加了语义噪声的图像，实现了较好的重建

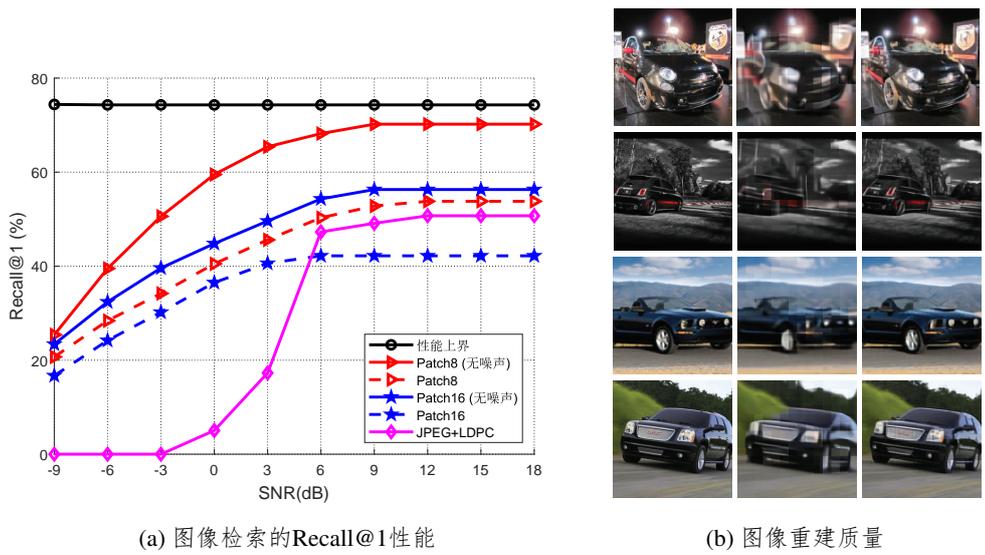


图 6.12 图像检索与图像重建的性能

质量。

6.6 本章小结

本章分析了语义噪声的产生原因，并提出了产生样本依赖语义噪声和非样本依赖语义噪声的方法。随后，针对语义噪声，搭建了鲁棒语义通信系统框架，并提出了基于权值扰动对抗训练。设计了一种带有噪声相关掩蔽策略的掩蔽VQ-VAE作为模型架构，并采用了一种用于表征编码特征的离散码本。为了提升系统的鲁棒性，提出了一种用于抑制噪声相关、任务无关特征被激活的FIM结构。仿真结果表明，该方法可以应用于许多下游任务，显著提升了语义通信系统对抗语义噪声的鲁棒性，同时大大降低了传输开销。未来的工作可以探索其他类型语义噪声的影响，并扩展所提出的方法，以提高语义通信系统在文本和语音等其他领域对语义噪声的鲁棒性。

7 工作总结和展望

本章将对论文工作进行总结，并对未来的研究工作进行展望。

7.1 工作总结

随着无线通信技术的演进和社会的高速发展，未来的6G网络将变得更为庞大复杂，无线业务数据量呈指数增长，现有的通信技术将面临前所未有的挑战。近年来，深度学习迎来了新一轮研究热潮和技术变革。利用其强大的学习能力和处理大数据的优势，解决通信中物理层收发机的复杂问题，赋能下一代通信系统已成必然趋势。面向6G系统的需求，本论文结合了毫米波通信、大规模MU-MIMO、离散透镜天线和语义通信等热点通信技术，并运用模型驱动网络、DRL网络、自编码器网络等深度学习方法对物理层收发机的关键模块进行设计，包括数字模拟预编码设计、波束选择、信道估计、信道反馈、鲁棒传输、星座点映射等模块。旨在提升物理层收发机性能，同时降低通信开销和计算复杂度。本论文的贡献可概括为：第二章和第三章设计了模型驱动网络分别用于物理层收发机中的预编码矩阵设计和信道估计。第四章至第六章设计了端到端深度学习网络架构，基于模型驱动网络、DRL网络、自编码器网络等网络结构联合设计了多个物理层收发机模块。具体而言，论文的主要研究成果总结如下：

1. 针对现有的无线通信系统中迭代优化算法复杂度高的问题，本文提出了一种矩阵形式的模型驱动网络框架。具体而言，基于优化算法的迭代结构，在前向传播中引入训练参数替代其中原有的高复杂度操作。针对网络训练的反向传播，提出了GCR，用于推导训练参数在训练过程中的闭式梯度，从而提升网络训练效率。随后，将所提出的模型驱动网络框架用于解决MU-MIMO系统中预编码设计的和速率最大化问题。在迭代WMMSE算法的基础上，设计了一种高效的模型驱动网络。具体而言，迭代WMMSE算法被展开为类似于DNN的逐层结构，并引入训练参数来替代其中的高复杂度矩阵求逆操作。为了更高效地训练网络，采用了GCR来描述反向传播过程中，网络相邻两层参数梯度的递推关系，用于推导每层网络参数的梯度闭式表达式。此

外，还讨论了该方案的计算复杂度和泛化能力。仿真结果表明，所提出的IAIDNN在降低计算复杂度的前提下，能够实现逼近迭代WMMSE算法的性能。

2. 针对模型驱动网络对不同输入样本的层数相同的问题，本文提出了一种基于DRL的模型驱动网络框架。该框架实现了对不同输入样本的可变层模型驱动网络，其中模型驱动网络的训练参数由DRL网络学习得到。具体而言，模型驱动网络的优化变量、训练参数和层结构被分别设计为深度强化学习的状态、动作和状态转移。随后，将该框架应用于解决MU-MIMO系统的信道估计问题。具体来说，首先建模了稀疏信道估计问题并提出使用带有离网参数的基于SBL的迭代算法求解。随后，在该迭代算法中引入训练参数，并将其展开为具有逐层结构的模型驱动网络。根据该得到的网络结构，采用上述可变层模型驱动网络框架解决该信道估计问题。为了更好地实现针对信道估计问题的可变层网络并使其估计精度更为可控，设计了停止分数来指示何时网络停止运行并输出最终结果。该算法在显著减少一批样本的网络平均层数的前提下，实现的性能优于固定层网络。
3. 针对无线通信中具有离散变量和耦合约束的MINLP问题，本文提出了一种高效的联合DNN框架。具体而言，该框架由DRL网络和模型驱动网络组成，分别用于优化离散变量和连续变量。随后，将该框架用于设计带有DLA的毫米波MU-MIMO系统中的收发机，以在发射功率约束和波束选择矩阵结构约束下使得系统和速率最大化。具体来说，DRL网络和模型驱动网络分别用于设计波束选择矩阵和数字预编码矩阵。对于DRL网络，波束选择问题被建模为MDP，并采用DDQN架构进行求解。基站被设计为智能体，其中的状态、动作和奖励函数的设计，使得满足约束的前提下尽可能选择使得用户性能好的波束。针对数字预编码矩阵的设计，提出了一种基于迭代WMMSE算法的模型驱动网络，并引入训练参数替换高复杂度操作。该联合训练网络具有较低的计算复杂度和较强的鲁棒性，在求解MINLP问题时，显著优于现有迭代算法。
4. 针对毫米波大规模MU-MIMO系统的物理层收发机模块的联合设计，本文提出了一种基于端到端深度学习的联合收发机网络框架。其针对信道估计、信道反馈和混合预编码等模块设计了DNN。该框架在接收端将收到的导频信号映射为信道反馈的信息比特，在发送端进一步将收到的信息比特映射为混合预编码器矩阵。为了进一步降低信令开销和由传输延迟导致的信道失配，基于上述框架设计了一种双时间尺度网络架构，其由长时间尺度网络和短时间尺度网络组成。长时间尺度网络估计并反馈高维原始信道，并在每帧更新一次收发端模拟预编码矩阵。短时间尺度网络估计

并反馈低维等效信道，并在每个时隙更新一次收发端数字预编码矩阵。该网络架构由于存在信道反馈模块，因此包含二值层，为了避免梯度中断从而更高效地训练网络，同时提升网络的泛化性能，提出了一种两阶段训练方法。随后，分析了网络的泛化能力和信令开销。该方案在显著降低信令开销和缩短导频序列的情况下，使系统的BER性能显著优于传统方案。

5. 针对端到端图像传输通信系统存在的鲁棒性问题，本文提出了一种鲁棒的端到端语义通信系统框架。具体而言，对语义通信系统中的样本依赖和非样本依赖语义噪声进行了模型。为对抗语义噪声，提出了基于权值扰动对抗训练方式，将含有语义噪声的样本纳入训练数据集。随后，提出对语义噪声出现概率较高的输入图像块进行掩蔽，并根据该掩蔽策略设计掩蔽VQ-VAE。此外，为了更好地适配实际数字通信系统同时提升系统传输效率，设计由发送端和接收端共享的离散码本来对图像特征进行编码，并将其和星座图建立映射关系。为了进一步提升系统的鲁棒性，提出了FIM来抑制噪声相关的且和图像任务无关的特征。因此，发送端只需要传输码本中与图像任务相关的重要特征的标号索引。该方法可以应用于多种不同的图像任务，显著提高了系统的鲁棒性，同时降低了传输开销。

7.2 进一步展望

目前，针对基于端到端模型驱动深度神经网络的物理层收发机设计的研究仍处于起步阶段。本文选取了五个具有代表性的核心技术问题进行研究，针对无线通信系统中的一些关键环节，例如物理层预编码、信道估计、波束选择、端到端学习、语义通信等，分别设计了基于深度学习的物理层通信算法并获得了性能增益，一定程度上推动了该领域的发展。但仍存在诸多问题有待进一步的深入研究。具体而言，未来拟进一步研究的内容包括：

1. 新型复杂通信场景下的基于深度学习的物理层收发机设计。本文研究了大规模MU-MIMO系统、DLA以及语义通信等场景中的若干物理层收发机相关问题，并设计了基于深度学习的相应解决方案。在未来的6G系统中，有更多新颖、复杂的通信场景，例如：智能反射面、联合通信感知一体化、卫星和无人机通信等。这些通信场景中，捕捉场景信息和设计通信算法较为困难，对复杂度、时延、性能等要求更为严格，传统算法遇到了技术瓶颈。因此，新型复杂通信场景下的深度学习算法有着广阔的应用前景和学术价值。

2. 基于深度学习的物理层收发机算法的工程化问题。本文的研究成果均为物理层收发机的仿真研究结果，还不曾有机会应用于实际通信系统中。在实际应用阶段，深度学习技术可能会面临硬件实现困难、模型不匹配、由于实际场景复杂所导致的难以训练等问题。需从硬件成本、实现复杂度等多方面深入研究深度学习在6G场景中的实现，辅助相关产品的落地，应用到民用或军用的各个领域。因此，如何将基于深度学习的物理层收发机设计与工程实践结合，是一个值得深入研究和探讨的问题。
3. 基于深度学习的物理层收发机算法的理论研究。尽管深度学习表现出优异的性能，但缺乏系统的理论基础和框架，阻碍了它的应用和推广。绝大多数基于深度学习的物理层收发机算法都缺乏深入理论支持，难以解释和证明其性能的优越性，缺少理论界限的推导。这使得研究者难以通过理论分析来深入理解其背后的机制和原理，从而需要大量的仿真试验和调参试错来实现高性能的深度学习算法设计。因此，如何运用数学体系构建理论框架，提供基于深度学习的物理层收发机算法相应的理论分析，并推导性能界限是一个至关重要的问题。
4. 物理层收发机模块中DNN的在线训练。本论文所研究的物理层收发机中的DNN是通过离线训练后在线部署使用的，即网络参数训练后需保持不变，不能在线调整来适应通信环境的变化。这是因为假设了信道的统计特性在相干时间内不发生改变。但在实际通信系统中，存在通信环境变化较快的场景，而线下训练的方式限制了算法的应用场景，当通信环境快速变化时，会面临训练和预测阶段的模型失配。相较于线下训练，基于在线训练DNN的物理层收发机设计具有更广阔的应用前景，但会导致较高的计算复杂度和通信时延。因此，如何将在线训练深度学习方法应用于快变通信场景是一个极具挑战性的问题。
5. 物理层收发机模块的专用神经网络架构设计。本论文所研究的物理层收发机模块中的DNN是基于迭代算法的模型驱动网络，或者针对CV和NLP领域针对图像和文本设计的DNN架构，尚未研发出适用于通信模块的专用网络架构。而物理层收发机有其特有的、有别于其它领域的系统需求。因此，为物理层收发机模块设计专用的高效神经网络架构，从而提升通信系统运行效率，显得尤为重要。

参考文献

- [1] D. Tse, P. Viswanath. Fundamentals Wireless Communication[M]. Cambridge University Press, 2005.
- [2] F. Boccardi, R. W. Heath, A. Lozano, et al. Five disruptive technology directions for 5G[J]. IEEE Communications Magazine, Feb. 2014. 52(2):74–80.
- [3] IMT-2020(5G)推进组. 5G愿景与需求[R]. 2014.
- [4] S. Rangan, T. S. Rappaport, E. Erkip. Millimeter-wave cellular wireless networks: Potentials and challenges[J]. Proceedings of the IEEE, Mar. 2014. 102(3):366–385.
- [5] 尤力, 高西奇. 大规模MIMO无线通信关键技术[J]. 中兴通讯技术, Feb. 2014. 20(2):26–28.
- [6] T. L. Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas[J]. IEEE Transactions on Wireless Communications, Nov. 2010. 9(11):3590–3600.
- [7] E. G. Larsson, O. Edfors, F. Tufvesson, T. L. Marzetta. Massive MIMO for next generation wireless systems[J]. IEEE Communications Magazine, Feb. 2014. 52(2):186–195.
- [8] I. F. Akyildiz, A. Kak, S. Nie. 6G and beyond: The future of wireless communications systems[J]. IEEE Access, Jul. 2020. 8:133995–134030.
- [9] W. Saad, M. Bennis, M. Chen. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems[J]. IEEE Network, May 2020. 34(3):134–142.
- [10] M. Giordani, M. Polese, M. Mezzavilla, et al. Toward 6G networks: Use cases and technologies[J]. IEEE Communications Magazine, Mar. 2020. 58(3):55–61.
- [11] 冯伟, 韦舒婷, 曹俊诚. 6G技术发展愿景与太赫兹通信[J]. 物理学报, 2021. 70(24):175–189.
- [12] 林德平, 彭涛, 刘春平. 6G愿景需求, 网络架构和关键技术展望[J]. 信息通信技术与政策, 2021. 47(1):82–89.
- [13] 闫实, 彭木根, 王文博. 通信-感知-计算融合: 6G愿景与关键技术[J]. 北京邮电大学学报, 2021. 44(4):1–11.
- [14] IMT-2030推进组. 6G总体愿景与潜在关键技术白皮书[R]. 2021.
- [15] Nokia Bell Labs. Communications in the 6G Era[R]. 2020.
- [16] 华为技术有限公司. 6G: 无线通信新征程[R]. 2022.
- [17] University of Oulu. 6G white paper on edge intelligence[R]. 2022.
- [18] Samsung. The next hyper: Connected experience for all[R]. 2020.

- [19] ATIS. Next G alliance, the roadmap[R]. 2022.
- [20] 赵亚军, 郁光辉, 徐汉青. 6G移动通信网络: 愿景, 挑战与关键技术[J]. 中国科学: 信息科学, 2019. 49(8):963–987.
- [21] 魏克军. 全球6G研究进展综述[J]. 移动通信, 2020. 44(3):34–36.
- [22] P. Yang, Y. Xiao, M. Xiao, S. Li. 6G wireless communications: Vision and potential techniques[J]. IEEE Network, Jul. 2019. 33(4):70–75.
- [23] 张贤达. 矩阵分析与应用. 第二版[M]. 北京:清华大学出版社, 2013.
- [24] T. O’Shea, J. Hoydis. An introduction to deep learning for the physical layer[J]. IEEE Transactions on Cognitive Communications and Networking, Dec. 2017. 3(4):563–575.
- [25] Y. S. Jeon, S. N. Hong, N. Lee. Supervised-learning-aided communication framework for MIMO systems with low-resolution ADCs[J]. IEEE Transactions on Vehicular Technology, Aug 2018. 67(8):7299–7313.
- [26] N. Samuel, T. Diskin, A. Wiesel. Learning to detect[J]. IEEE Transactions on Signal Processing, May 2019. 67(10):2554–2564.
- [27] Z. Qin, H. Ye, G. Y. Li, B. F. Juang. Deep learning in physical layer communications[J]. IEEE Wireless Communications, Apr. 2019. 26(2):93–99.
- [28] A. Balatsoukas-Stimming, C. Studer. Deep unfolding for communications systems: A survey and some new directions[C]//IEEE International Workshop on Signal Processing Systems (SiPS’2019). Nanjing, China, Oct. 2019:266–271.
- [29] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain.[J]. Psychological Review, 1958. 65(6):386.
- [30] J. L. Elman. Distributed representations, simple recurrent networks, and grammatical structure[J]. Machine Learning, Sep. 1991. 7(2):195–225.
- [31] S. Hochreiter, J. Schmidhuber. Long short-term memory[J]. Neural Computation, Nov. 1997. 9(8):1735–1780.
- [32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, Nov. 1998. 86(11):2278–2324.
- [33] A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017. 60(6):84–90.
- [34] C. Tian, L. Fei, W. Zheng, et al. Deep learning on image denoising: An overview[J]. Neural Networks, Jul. 2020. 131:251–275.
- [35] J. Long, E. Shelhamer, T. Darrell. Fully convolutional networks for semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2015). Boston, MA, USA, Jun. 2015:3431–3440.
- [36] S. P. Singh, A. Kumar, H. Darbari, et al. Machine translation using deep learning: An

- overview[C]//International Conference on Computer, Communications and Electronics (Comptelix). Jaipur, India, Jul. 2017:162–167.
- [37] D. Xiong, D. Zhang, X. Zhao, Y. Zhao. Deep learning for EMG-based human-machine interaction: A review[J]. *IEEE/CAA Journal of Automatica Sinica*, Mar. 2021. 8(3):512–533.
- [38] S. Liu, S. Zhang, X. Zhang, H. Wang. R-Trans: RNN transformer network for chinese machine reading comprehension[J]. *IEEE Access*, 2019. 7:27736–27745.
- [39] W. Zhang, X. Cui, U. Finkler, et al. Distributed deep learning strategies for automatic speech recognition[C]//IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2019). Brighton, UK, May 2019:5706–5710.
- [40] K. Choi, G. Fazekas, K. Cho, M. Sandler. A tutorial on deep learning for music information retrieval[J]. arXiv preprint arXiv:1709.04396, 2017.
- [41] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [42] C. Szegedy, L. Wei, J. Yangqing, et al. Going deeper with convolutions[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015). Boston, MA, USA, Jun. 2015:1–9.
- [43] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2016). Las Vegas, NV, USA, Jun. 2016:770–778.
- [44] Y. LeCun, Y. Bengio, G. Hinton. Deep learning[J]. *Nature*, May 2015. 521(7553):436–444.
- [45] D. P. Kingma, M. Welling. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- [46] J. T. Chien, C. H. Lee. Deep unfolding for topic models[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Feb. 2018. 40(2):318–331.
- [47] R. Liu, S. Cheng, L. Ma, et al. Deep proximal unrolling: Algorithmic framework, convergence analysis and applications[J]. *IEEE Transactions on Image Processing*, Oct. 2019. 28(10):5013–5026.
- [48] V. Monga, Y. Li, Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing[J]. *IEEE Signal Processing Magazine*, Mar. 2021. 38(2):18–44.
- [49] L. Zhang, G. Wang, G. B. Giannakis. Real-time power system state estimation and forecasting via deep unrolled neural networks[J]. *IEEE Transactions on Signal Processing*, Aug. 2019. 67(15):4069–4077.
- [50] H. He, S. Jin, C. K. Wen, et al. Model-driven deep learning for physical layer communications[J]. *IEEE Wireless Communications*, Oct. 2019. 26(5):77–83.
- [51] J. R. Hershey, J. Le Roux, F. Weninger. Deep unfolding: Model-based inspiration of novel deep architectures[C]//Advances in Neural Information Processing Systems (NIPS'2014). Montreal, Quebec, Canada, Dec. 2014:1–27.
- [52] L. Bottou. Large-scale machine learning with stochastic gradient descent[C]//International Conference on Computational Statistics (COMPSTAT'2010). Paris, France, Aug. 2010:177–186.

- [53] N. Qian. On the momentum term in gradient descent learning algorithms[J]. *Neural Networks*, Aug. 1999. 12(1):145–151.
- [54] M. C. Mukkamala, M. Hein. Variants of rmsprop and adagrad with logarithmic regret bounds[C]//*International Conference on Machine Learning (ICML'2017)*. PMLR, Sydney, Australia, Aug. 2017:2545–2553.
- [55] D. P. Kingma, J. Ba. Adam: A method for stochastic optimization[C]//*International Conference on Learning Representations (ICLR'2015)*. San Diego, CA, USA, Apr. 2015:1–15.
- [56] K. Hornik, M. Stinchcombe, H. White. Multilayer feedforward networks are universal approximators[J]. *Neural networks*, Mar. 1989. 2(5):359–366.
- [57] A. Zappone, M. Di Renzo, M. Debbah. Wireless networks design in the era of deep learning: Model-based, AI-based, or both?[J]. *IEEE Transactions on Communications*, Oct 2019. 67(10):7331–7376.
- [58] K. B. Letaief, W. Chen, Y. Shi, et al. The roadmap to 6G: AI empowered wireless networks[J]. *IEEE Communications Magazine*, Aug. 2019. 57(8):84–90.
- [59] 张静, 金石, 温朝凯, 高飞飞, 江涛. 基于人工智能的无线传输技术最新研究进展[J]. *电信科学*, 2018. 34(8):46–55.
- [60] 何恒涛. 面向智能通信的先进收发机理论与关键技术[D]. 江苏:东南大学, 2020.
- [61] 华郁秀. 无线网络智能感知与切片资源管理研究[D]. 浙江:浙江大学, 2021.
- [62] 韦逸. 基于深度学习的物理层通信技术研究[D]. 浙江:浙江大学, 2022.
- [63] 李润东. 基于深度学习的通信信号智能盲检测与识别技术研究[D]. 四川:电子科技大学, 2021.
- [64] 汪周飞, 袁伟娜. 基于深度学习的多载波系统信道估计与检测[J]. *浙江大学学报(工学版)*, 2020. 54(4):732–738.
- [65] K. Hornik, M. Stinchcombe, H. White. Multilayer feedforward networks are universal approximators[J]. *Neural Networks*, 1989. 2(5):359–366.
- [66] H. Ye, G. Y. Li, B. H. Juang. Deep learning based end-to-end wireless communication systems without pilots[J]. *IEEE Transactions on Cognitive Communications and Networking*, Sep. 2021. 7(3):702–714.
- [67] R. Collobert, K. Kavukcuoglu, C. Farabet. Torch7: A matlab-like environment for machine learning[C]//*BigLearn, NIPS workshop*. 2011.
- [68] M. Abadi, A. Agarwal, P. Barham, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems[J]. *arXiv preprint arXiv:1603.04467*, 2016.
- [69] Y. Jia, E. Shelhamer, J. Donahue, et al. Caffe: Convolutional architecture for fast feature embedding[C]//*Proceedings of ACM International Conference on Multimedia*. 2014:675–678.
- [70] J. Bergstra, O. Breuleux, F. Bastien, et al. Theano: a CPU and GPU math expression compiler[C]//*Proceedings of the Python for Scientific Computing Conference (SciPy)*. volume 4. 2010:1–7.
- [71] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, F. Capolino. Millimeter-wave massive MIMO: the next

- wireless revolution?[J]. *IEEE Communications Magazine*, Sep. 2014. 52(9):56–62.
- [72] A. Ghosh, T. A. Thomas, M. C. Cudak, et al. Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks[J]. *IEEE Journal on Selected Areas in Communications*, Jun. 2014. 32(6):1152–1163.
- [73] F. Rusek, D. Persson, B. K. Lau, et al. Scaling up MIMO: Opportunities and challenges with very large arrays[J]. *IEEE Signal Processing Magazine*, Jan. 2013. 30(1):40–60.
- [74] E. Bjornson, L. Sanguinetti, J. Hoydis, M. Debbah. Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?[J]. *IEEE Transactions on Wireless Communications*, Jun. 2015. 14(6):3059–3075.
- [75] L. Lu, G. Y. Li, A. L. Swindlehurst, et al. An overview of massive MIMO: Benefits and challenges[J]. *IEEE Journal of Selected Topics in Signal Processing*, Oct. 2014. 8(5):742–758.
- [76] H. Q. Ngo, E. G. Larsson, T. L. Marzetta. Energy and spectral efficiency of very large multiuser MIMO systems[J]. *IEEE Transactions on Communications*, Apr. 2013. 61(4):1436–1449.
- [77] S. Ye, R. S. Blum. Optimized signaling for MIMO interference systems with feedback[J]. *IEEE Transactions on Signal Processing*, Nov. 2003. 51(11):2839–2848.
- [78] G. Scutari, D. P. Palomar, S. Barbarossa. The MIMO iterative waterfilling algorithm[J]. *IEEE Transactions on Signal Processing*, May 2009. 57(5):1917–1935.
- [79] J. Brady, N. Behdad, A. M. Sayeed. Beam-space MIMO for millimeter-wave communications: System architecture, modeling, analysis, and measurements[J]. *IEEE Transactions on Antennas and Propagation*, Jul. 2013. 61(7):3814–3827.
- [80] H. Yang, T. L. Marzetta. Performance of conjugate and zero-forcing beamforming in large-scale antenna systems[J]. *IEEE Journal on Selected Areas in Communications*, Feb. 2013. 31(2):172–179.
- [81] S. Atapattu, P. Dharmawansa, C. Tellambura, J. Evans. Exact outage analysis of multiple-user downlink with MIMO matched-filter precoding[J]. *IEEE Communications Letters*, Dec. 2017. 21(12):2754–2757.
- [82] Schmidt, Joham, Utschick. Minimum mean square error vector precoding[C]//*IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2005)*. Berlin, Germany, Sep. 2005:107–111.
- [83] S. S. Christensen, R. Agarwal, E. De Carvalho, J. M. Cioffi. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design[J]. *IEEE Transactions on Wireless Communications*, Dec. 2008. 7(12):4792–4799.
- [84] W. Yu, J. M. Cioffi. FDMA capacity of gaussian multiple-access channels with ISI[J]. *IEEE Transactions on Communications*, Jan. 2002. 50(1):102–111.
- [85] N. D. Sidiropoulos, T. N. Davidson, Z. Q. Luo. Transmit beamforming for physical-layer multicasting[J]. *IEEE Transactions on Signal Processing*, Jun. 2006. 54(6):2239–2251.

- [86] Q. Shi, M. Razaviyayn, Z. Q. Luo, C. He. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel[J]. *IEEE Transactions on Signal Processing*, Sep. 2011. 59(9):4331–4340.
- [87] S. K. Mohammed, E. G. Larsson. Per-antenna constant envelope precoding for large multi-user MIMO systems[J]. *IEEE Transactions on Communications*, Mar. 2013. 61(3):1059–1071.
- [88] M. Costa. Writing on dirty paper (corresp.)[J]. *IEEE Transactions on Information Theory*, May 1983. 29(3):439–441.
- [89] R. D. Wesel, J. M. Cioffi. Achievable rates for tomlinson-harashima precoding[J]. *IEEE Transactions on Information Theory*, Mar. 1998. 44(2):824–831.
- [90] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, N. D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management[J]. *IEEE Transactions on Signal Processing*, Oct. 2018. 66(20):5438–5453.
- [91] W. Lee, M. Kim, D. H. Cho. Deep power control: Transmit power control scheme based on convolutional neural network[J]. *IEEE Communications Letters*, Jun. 2018. 22(6):1276–1279.
- [92] F. Liang, C. Shen, W. Yu, F. Wu. Towards optimal power control via ensembling deep neural networks[J]. *IEEE Transactions on Communications*, Mar. 2020. 68(3):1760–1776.
- [93] A. M. Elbir. CNN-based precoder and combiner design in mmwave MIMO systems[J]. *IEEE Communications Letters*, Jul. 2019. 23(7):1240–1243.
- [94] Z. Bo, R. Liu, M. Li, Q. Liu. Deep learning based efficient symbol-level precoding design for MU-MISO systems[J]. *IEEE Transactions on Vehicular Technology*, Aug. 2021. 70(8):8309–8313.
- [95] F. Sotrabadi, Z. Chen, W. Yu. Deep active learning approach to adaptive beamforming for mmwave initial alignment[J]. *IEEE Journal on Selected Areas in Communications*, Aug. 2021. 39(8):2347–2360.
- [96] Y. Yuan, G. Zheng, K. K. Wong, B. Ottersten, Z. Q. Luo. Transfer learning and meta learning-based fast downlink beamforming adaptation[J]. *IEEE Transactions on Wireless Communications*, Mar. 2021. 20(3):1742–1755.
- [97] Z. Lei, X. Liao, Z. Gao, A. Li. CI-NN: A model-driven deep learning-based constructive interference precoding scheme[J]. *IEEE Communications Letters*, Jun. 2021. 25(6):1896–1900.
- [98] J. Zhang, M. You, G. Zheng, I. Krikidis, L. Zhao. Model-driven learning for generic MIMO downlink beamforming with uplink channel information[J]. *IEEE Transactions on Wireless Communications*, Apr. 2022. 21(4):2368–2382.
- [99] W. Xia, G. Zheng, K. K. Wong, H. Zhu. Model-driven beamforming neural networks[J]. *IEEE Wireless Communications*, Feb. 2020. 27(1):68–75.
- [100] Y. He, H. He, C. K. Wen, S. Jin. Model-driven deep learning for massive multiuser MIMO constant envelope precoding[J]. *IEEE Wireless Communications Letters*, Nov. 2020. 9(11):1835–1839.

- [101] H. He, M. Zhang, S. Jin, C. K. Wen, G. Y. Li. Model-driven deep learning for massive MU-MIMO with finite-alphabet precoding[J]. *IEEE Communications Letters*, Oct. 2020. 24(10):2216–2220.
- [102] S. Liu, Z. Gao, C. Hu, et al. Model-driven deep learning based precoding for FDD cell-free massive MIMO with imperfect CSI[C]//*International Wireless Communications and Mobile Computing (IWCMC'2022)*. Dubrovnik, Croatia, May 2022:696–701.
- [103] L. Liang, W. Xu, X. Dong. Low-complexity hybrid precoding in massive multiuser MIMO systems[J]. *IEEE Wireless Communications Letters*, Dec. 2014. 3(6):653–656.
- [104] W. Ni, X. Dong, W. S. Lu. Near-optimal hybrid processing for massive MIMO systems via matrix decomposition[J]. *IEEE Transactions on Signal Processing*, Aug. 2017. 65(15):3922–3933.
- [105] Q. Shi, M. Hong. Spectral efficiency optimization for millimeter wave multiuser MIMO systems[J]. *IEEE Journal of Selected Topics in Signal Processing*, Jun. 2018. 12(3):455–468.
- [106] F. Sofrabi, W. Yu. Hybrid digital and analog beamforming design for large-scale antenna arrays[J]. *IEEE Journal of Selected Topics in Signal Processing*, Apr. 2016. 10(3):501–513.
- [107] X. Yu, J. C. Shen, J. Zhang, K. B. Letaief. Alternating minimization algorithms for hybrid precoding in millimeter wave MIMO systems[J]. *IEEE Journal of Selected Topics in Signal Processing*, Apr. 2016. 10(3):485–500.
- [108] S. He, J. Wang, Y. Huang, et al. Codebook-based hybrid precoding for millimeter wave multiuser systems[J]. *IEEE Transactions on Signal Processing*, Oct. 2017. 65(20):5289–5304.
- [109] S. S. Ioushua, Y. C. Eldar. A family of hybrid analog-digital beamforming methods for massive MIMO systems[J]. *IEEE Transactions on Signal Processing*, Jun. 2019. 67(12):3243–3257.
- [110] A. Liu, V. K. N. Lau, M. J. Zhao. Online successive convex approximation for two-stage stochastic nonconvex optimization[J]. *IEEE Transactions on Signal Processing*, Nov. 2018. 66(22):5941–5955.
- [111] R. Mai, T. Le-Ngoc, D. H. N. Nguyen. Two-timescale hybrid RF-baseband precoding with MMSE-VP for multi-user massive MIMO broadcast channels[J]. *IEEE Transactions on Wireless Communications*, Jul. 2018. 17(7):4462–4476.
- [112] Y. Cai, K. Xu, A. Liu, et al. Two-timescale hybrid analog-digital beamforming for mmWave full-duplex MIMO multiple-relay aided systems[J]. *IEEE Journal on Selected Areas in Communications*, Sep. 2020. 38(9):2086–2103.
- [113] P. Dong, H. Zhang, G. Y. Li. Framework on deep learning-based joint hybrid processing for mmWave massive MIMO systems[J]. *IEEE Access*, Jun. 2020. 8:106023–106035.
- [114] H. Huang, Y. Song, J. Yang, et al. Deep-learning-based millimeter-wave massive MIMO for hybrid precoding[J]. *IEEE Transactions on Vehicular Technology*, Mar. 2019. 68(3):3027–3032.
- [115] A. M. Elbir, A. K. Papazafeiropoulos. Hybrid precoding for multiuser millimeter wave massive MIMO systems: A deep learning approach[J]. *IEEE Transactions on Vehicular Technology*, Jan. 2020. 69(1):552–

- 563.
- [116] P. Zhang, L. Pan, T. Laohapensaeng, M. Chongcheawchamnan. Hybrid beamforming based on an unsupervised deep learning network for downlink channels with imperfect CSI[J]. *IEEE Wireless Communications Letters*, Jul. 2022. 11(7):1543–1547.
- [117] H. Hojatian, J. Nadal, J. F. Frigon, F. Leduc-Primeau. Unsupervised deep learning for massive MIMO hybrid beamforming[J]. *IEEE Transactions on Wireless Communications*, Nov. 2021. 20(11):7086–7099.
- [118] D. Hu, Y. Zhang, L. He, J. Wu. Low-complexity deep-learning-based DOA estimation for hybrid massive MIMO systems with uniform circular arrays[J]. *IEEE Wireless Communications Letters*, Jan. 2020. 9(1):83–86.
- [119] A. M. Elbir. A deep learning framework for hybrid beamforming without instantaneous CSI feedback[J]. *IEEE Transactions on Vehicular Technology*, Oct. 2020. 69(10):11743–11755.
- [120] T. Peken, S. Adiga, R. Tandon, T. Bose. Deep learning for SVD and hybrid beamforming[J]. *IEEE Transactions on Wireless Communications*, Oct. 2020. 19(10):6621–6642.
- [121] M. Chai, S. Tang, M. Zhao, W. Zhou. HPNet: A compressed neural network for robust hybrid precoding in multi-user massive MIMO systems[C]//*IEEE Global Communications Conference (GlobeCom'2020)*. Taipei, Taiwan, Dec. 2020:1–7.
- [122] E. Balevi, J. G. Andrews. Unfolded hybrid beamforming with GAN compressed ultra-low feedback overhead[J]. *IEEE Transactions on Wireless Communications*, Dec. 2021. 20(12):8381–8392.
- [123] K. Y. Chen, H. Y. Chang, R. Y. Chang, W. H. Chung. Hybrid beamforming in mmwave MIMO-OFDM systems via deep unfolding[C]//*IEEE Vehicular Technology Conference (VTC'2022-Spring)*. Helsinki, Finland, Jun. 2022:1–7.
- [124] Y. Zeng, R. Zhang. Millimeter wave MIMO with lens antenna array: A new path division multiplexing paradigm[J]. *IEEE Transactions on Communications*, Apr. 2016. 64(4):1557–1571.
- [125] H. Liu, X. Yuan, Y. J. Zhang. Statistical beamforming for FDD downlink massive MIMO via spatial information extraction and beam selection[J]. *IEEE Transactions on Wireless Communications*, Jul. 2020. 19(7):4617–4631.
- [126] X. Gao, L. Dai, Z. Chen, et al. Near-optimal beam selection for beamspace mmWave massive MIMO systems[J]. *IEEE Communications Letters*, May 2016. 20(5):1054–1057.
- [127] R. Guo, Y. Cai, M. Zhao, et al. Joint design of beam selection and precoding matrices for mmWave MU-MIMO systems relying on lens antenna arrays[J]. *IEEE Journal of Selected Topics in Signal Processing*, May 2018. 12(2):313–325.
- [128] A. Sayeed, J. Brady. Beamspace MIMO for high-dimensional multiuser communication at millimeter-wave frequencies[C]//*IEEE Global Communications Conference (GlobeCom'2013)*. Atlanta, GA, USA, Dec. 2013:3679–3684.

- [129] P. V. Amadori, C. Masouros. Low RF-complexity millimeter-wave beamspace-MIMO systems by beam selection[J]. *IEEE Transactions on Communications*, Jun. 2015. 63(6):2212–2223.
- [130] W. Shen, X. Bu, X. Gao, et al. Beamspace precoding and beam selection for wideband millimeter-wave MIMO relying on lens antenna arrays[J]. *IEEE Transactions on Signal Processing*, Dec. 2019. 67(24):6301–6313.
- [131] N. J. Myers, A. Mezghani, R. W. Heath. FALP: Fast beam alignment in mmWave systems with low-resolution phase shifters[J]. *IEEE Transactions on Communications*, Dec. 2019. 67(12):8739–8753.
- [132] R. Pal, A. K. Chaitanya, K. V. Srinivas. Low-complexity beam selection algorithms for millimeter wave beamspace MIMO systems[J]. *IEEE Communications Letters*, Apr. 2019. 23(4):768–771.
- [133] Y. Long, Z. Chen, J. Fang, C. Tellambura. Data-driven-based analog beam selection for hybrid beam-forming under mmWave channels[J]. *IEEE Journal of Selected Topics in Signal Processing*, May 2018. 12(2):340–352.
- [134] A. Klautau, P. Batista, N. Gonzalez-Prelcic, et al. 5G MIMO data for machine learning: Application to beam-selection using deep learning[C]//*Information Theory and Applications Workshop (ITA'2018)*. San Diego, CA, USA, Feb. 2018:1–9.
- [135] C. Anton-Haro, X. Mestre. Learning and data-driven beam selection for mmWave communications: An angle of arrival-based approach[J]. *IEEE Access*, Feb. 2019. 7:20404–20415.
- [136] H. Echigo, Y. Cao, M. Bouazizi, T. Ohtsuki. A deep learning-based low overhead beam selection in mmwave communications[J]. *IEEE Transactions on Vehicular Technology*, Jan. 2021. 70(1):682–691.
- [137] S. Rezaie, A. Amiri, E. de Carvalho, C. N. Manchon. Deep transfer learning for location-aware millimeter wave beam selection[J]. *IEEE Communications Letters*, Sep. 2021. 25(9):2963–2967.
- [138] S. Rezaie, E. de Carvalho, C. N. Manchon. A deep learning approach to location- and orientation-aided 3D beam selection for mmwave communications[J]. *IEEE Transactions on Wireless Communications*, Dec. 2022. 21(12):11110–11124.
- [139] H. Ahn, I. Orikumhi, J. Kang, et al. Machine learning-based vision-aided beam selection for mmwave multiuser MISO system[J]. *IEEE Wireless Communications Letters*, Jun. 2022. 11(6):1263–1267.
- [140] S. He, S. Xiong, W. Zhang, et al. GBLinks: GNN-based beam selection and link activation for ultra-dense D2D mmwave networks[J]. *IEEE Transactions on Communications*, May 2022. 70(5):3451–3466.
- [141] M. Tian, Z. Zhang, Q. Xu, L. Yang. A privacy-preserved split learning solution for deep learning-based mmwave beam selection[J]. *IEEE Communications Letters*, Jul. 2022. 26(7):1474–1478.
- [142] J. C. Shen, J. Zhang, K. C. Chen, K. B. Letaief. High-dimensional CSI acquisition in massive MIMO: Sparsity-inspired approaches[J]. *IEEE Systems Journal*, Mar. 2017. 11(1):32–40.
- [143] O. Edfors, M. Sandell, J. J. van de Beek, et al. OFDM channel estimation by singular value decomposition[J]. *IEEE Transactions on Communications*, Jul. 1998. 46(7):931–939.

- [144] Y. Li, N. Seshadri, S. Ariyavisitakul. Channel estimation for OFDM systems with transmitter diversity in mobile wireless channels[J]. *IEEE Journal on Selected Areas in Communications*, Mar. 1999. 17(3):461–471.
- [145] H. Yin, D. Gesbert, M. Filippou, Y. Liu. A coordinated approach to channel estimation in large-scale multiple-antenna systems[J]. *IEEE Journal on Selected Areas in Communications*, Feb. 2013. 31(2):264–273.
- [146] D. D. Souza, M. M. M. Freitas, G. S. Borges, et al. Effective channel blind estimation in cell-free massive MIMO networks[J]. *IEEE Wireless Communications Letters*, Mar. 2022. 11(3):468–472.
- [147] F. Gao, A. Nallanathan. Blind channel estimation for MIMO OFDM systems via nonredundant linear precoding[J]. *IEEE Transactions on Signal Processing*, Jan. 2007. 55(2):784–789.
- [148] C. Shin, R. W. Heath, E. J. Powers. Blind channel estimation for MIMO-OFDM systems[J]. *IEEE Transactions on Vehicular Technology*, Mar. 2007. 56(2):670–685.
- [149] E. Nayebi, B. D. Rao. Semi-blind channel estimation for multiuser massive MIMO systems[J]. *IEEE Transactions on Signal Processing*, Jan. 2018. 66(2):540–553.
- [150] B. Muquet, M. de Courville, P. Duhamel. Subspace-based blind and semi-blind channel estimation for OFDM systems[J]. *IEEE Transactions on Signal Processing*, Jul. 2002. 50(7):1699–1712.
- [151] K. Mawatwal, D. Sen, R. Roy. A semi-blind channel estimation algorithm for massive MIMO systems[J]. *IEEE Wireless Communications Letters*, Feb. 2017. 6(1):70–73.
- [152] F. Fernandes, A. Ashikhmin, T. L. Marzetta. Inter-cell interference in noncooperative TDD large scale antenna systems[J]. *IEEE Journal on Selected Areas in Communications*, Feb. 2013. 31(2):192–201.
- [153] J. Choi, D. J. Love, P. Bidigare. Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory[J]. *IEEE Journal of Selected Topics in Signal Processing*, Oct. 2014. 8(5):802–814.
- [154] E. Bjornson, J. Hoydis, M. Kountouris, M. Debbah. Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits[J]. *IEEE Transactions on Information Theory*, Nov. 2014. 60(11):7112–7139.
- [155] X. Rao, V. K. N. Lau. Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems[J]. *IEEE Transactions on Signal Processing*, Jun. 2014. 62(12):3261–3271.
- [156] J. C. Shen, J. Zhang, E. Alsusa, K. B. Letaief. Compressed CSI acquisition in FDD massive MIMO: How much training is needed?[J]. *IEEE Transactions on Wireless Communications*, Jun. 2016. 15(6):4145–4156.
- [157] J. A. Tropp, A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit[J]. *IEEE Transactions on Information Theory*, Dec. 2007. 53(12):4655–4666.
- [158] A. Liu, L. Lian, V. K. N. Lau, X. Yuan. Downlink channel estimation in multiuser massive MIMO with

- hidden markovian sparsity[J]. *IEEE Transactions on Signal Processing*, Sep. 2018. 66(18):4796–4810.
- [159] Y. Han, J. Lee. Two-stage compressed sensing for millimeter wave channel estimation[C]//*IEEE International Symposium on Information Theory (ISIT'2016)*. Barcelona, Spain, Jul. 2016:860–864.
- [160] J. Choi, D. J. Love, P. Bidigare. Downlink training techniques for FDD massive MIMO systems: Open-loop and closed-loop training with memory[J]. *IEEE Journal of Selected Topics in Signal Processing*, Oct. 2014. 8(5):802–814.
- [161] X. Wu, L. Gu, W. Wang, X. Gao. Pilot design and AMP-based channel estimation for massive MIMO-OFDM uplink transmission[C]//*IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC'2016)*. Valencia, Spain, Sep. 2016:1–7.
- [162] D. Lee. MIMO OFDM channel estimation via block stagewise orthogonal matching pursuit[J]. *IEEE Communications Letters*, Oct. 2016. 20(10):2115–2118.
- [163] Y. Ding, B. D. Rao. Dictionary learning-based sparse channel representation and estimation for FDD massive MIMO systems[J]. *IEEE Transactions on Wireless Communications*, Aug. 2018. 17(8):5437–5451.
- [164] J. Dai, A. Liu, V. K. N. Lau. FDD massive MIMO channel estimation with arbitrary 2D-array geometry[J]. *IEEE Transactions on Signal Processing*, May 2018. 66(10):2584–2599.
- [165] S. Ji, Y. Xue, L. Carin. Bayesian compressive sensing[J]. *IEEE Transactions on Signal Processing*, Jun. 2008. 56(6):2346–2356.
- [166] H. Ye, G. Y. Li, B. H. Juang. Power of deep learning for channel estimation and signal detection in OFDM systems[J]. *IEEE Wireless Communications Letters*, Feb. 2018. 7(1):114–117.
- [167] P. Dong, H. Zhang, G. Y. Li, et al. Deep CNN-based channel estimation for mmWave massive MIMO systems[J]. *IEEE Journal of Selected Topics in Signal Processing*, Sep. 2019. 13(5):989–1000.
- [168] H. Hirose, T. Ohtsuki, G. Gui. Deep learning-based channel estimation for massive mimo systems with pilot contamination[J]. *IEEE Open Journal of Vehicular Technology*, 2021. 2:67–77.
- [169] H. Huang, J. Yang, H. Huang, Y. Song, G. Gui. Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system[J]. *IEEE Transactions on Vehicular Technology*, Sep. 2018. 67(9):8549–8560.
- [170] Y. Jin, J. Zhang, S. Jin, B. Ai. Channel estimation for cell-free mmwave massive MIMO through deep learning[J]. *IEEE Transactions on Vehicular Technology*, Oct. 2019. 68(10):10325–10329.
- [171] C. Liu, X. Liu, D. W. K. Ng, J. Yuan. Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications[J]. *IEEE Transactions on Wireless Communications*, Feb. 2022. 21(2):898–912.
- [172] Y. Wei, M. M. Zhao, M. Zhao, et al. An AMP-based network with deep residual learning for mmWave beamspace channel estimation[J]. *IEEE Wireless Communications Letters*, Aug. 2019. 8(4):1289–1292.

- [173] H. He, C. K. Wen, S. Jin, G. Y. Li. Deep learning-based channel estimation for beamspace mmWave massive MIMO systems[J]. *IEEE Wireless Communications Letters*, Oct. 2018. 7(5):852–855.
- [174] Z. Mao, X. Liu, M. Peng, et al. Joint channel estimation and active-user detection for massive access in internet of things-a deep learning approach[J]. *IEEE Internet of Things Journal*, Feb. 2022. 9(4):2870–2881.
- [175] A. Ghazanfari, T. Van Chien, E. Bjornson, E. G. Larsson. Model-based and data-driven approaches for downlink massive MIMO channel estimation[J]. *IEEE Transactions on Communications*, Mar. 2022. 70(3):2085–2101.
- [176] X. Ma, Z. Gao, F. Gao, M. Di Renzo. Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems[J]. *IEEE Journal on Selected Areas in Communications*, Aug. 2021. 39(8):2388–2406.
- [177] Z. Qin, J. Fan, Y. Liu, Y. Gao, G. Y. Li. Sparse representation for wireless communications: A compressive sensing approach[J]. *IEEE Signal Processing Magazine*, May 2018. 35(3):40–58.
- [178] Z. Gao, L. Dai, Z. Wang, S. Chen. Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO[J]. *IEEE Transactions on Signal Processing*, Dec. 2015. 63(23):6169–6183.
- [179] X. Rao, V. K. N. Lau. Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems[J]. *IEEE Transactions on Signal Processing*, Jun. 2014. 62(12):3261–3271.
- [180] A. Alkhateeb, G. Leus, R. W. Heath. Limited feedback hybrid precoding for multi-user millimeter wave systems[J]. *IEEE Transactions on Wireless Communications*, Nov. 2015. 14(11):6481–6494.
- [181] M. R. Castellanos, V. Raghavan, J. H. Ryu, et al. Channel-reconstruction-based hybrid precoding for millimeter-wave multi-user MIMO systems[J]. *IEEE Journal of Selected Topics in Signal Processing*, May 2018. 12(2):383–398.
- [182] D. J. Love, R. W. Heath, V. K. N. Lau, et al. An overview of limited feedback in wireless communication systems[J]. *IEEE Journal on Selected Areas in Communications*, Oct. 2008. 26(8):1341–1365.
- [183] N. Jindal. MIMO broadcast channels with finite-rate feedback[J]. *IEEE Transactions on Information Theory*, Nov. 2006. 52(11):5045–5060.
- [184] P. Liang, J. Fan, W. Shen, Z. Qin, G. Y. Li. Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems[J]. *IEEE Transactions on Vehicular Technology*, Aug. 2020. 69(8):9217–9222.
- [185] Z. Hu, J. Guo, G. Liu, H. Zheng, J. Xue. MRFNet: A deep learning-based CSI feedback approach of massive MIMO systems[J]. *IEEE Communications Letters*, Oct. 2021. 25(10):3310–3314.
- [186] B. Cao, Y. Yang, P. Ran, D. He, G. He. ACCsiNet: Asymmetric convolution-based autoencoder framework for massive MIMO CSI feedback[J]. *IEEE Communications Letters*, Dec. 2021. 25(12):3873–3877.

- [187] Y. Cui, A. Guo, C. Song. TransNet: Full attention network for CSI feedback in FDD massive MIMO system[J]. *IEEE Wireless Communications Letters*, May 2022. 11(5):903–907.
- [188] S. Ji, M. Li. CLNet: Complex input lightweight neural network designed for massive MIMO CSI feedback[J]. *IEEE Wireless Communications Letters*, Oct. 2021. 10(10):2318–2322.
- [189] B. Tolba, M. Elsabrouty, M. G. Abdu-Aguye, et al. Massive MIMO CSI feedback based on generative adversarial network[J]. *IEEE Communications Letters*, Dec. 2020. 24(12):2805–2808.
- [190] M. Hussien, K. K. Nguyen, M. Cheriet. PRVNet: A novel partially-regularized variational autoencoders for massive MIMO CSI feedback[C]//*IEEE Wireless Communications and Networking Conference (WCNC' 2022)*. Austin, TX, USA, Apr. 2022:2286–2291.
- [191] C. K. Wen, W. T. Shih, S. Jin. Deep learning for massive MIMO CSI feedback[J]. *IEEE Wireless Communications Letters*, Oct. 2018. 7(5):748–751.
- [192] T. Wang, C. K. Wen, S. Jin, G. Y. Li. Deep learning-based CSI feedback approach for time-varying massive MIMO channels[J]. *IEEE Wireless Communications Letters*, Apr. 2019. 8(2):416–419.
- [193] C. Lu, W. Xu, H. Shen, et al. MIMO channel information feedback using deep recurrent network[J]. *IEEE Communications Letters*, Jan. 2019. 23(1):188–191.
- [194] Z. Liu, L. Zhang, Z. Ding. Exploiting Bi-directional channel reciprocity in deep learning for low rate massive MIMO CSI feedback[J]. *IEEE Wireless Communications Letters*, Jun. 2019. 8(3):889–892.
- [195] Y. Liu, O. Simeone. Hyperrnn: Deep learning-aided downlink CSI acquisition via partial channel reciprocity for FDD massive MIMO[C]//*IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC'2021)*. Lucca, Italy, Sep. 2021:31–35.
- [196] C. Qing, Q. Ye, W. Liu, J. Wang. Fusion learning for 1-bit CS-based superimposed CSI feedback with Bi-directional channel reciprocity[J]. *IEEE Communications Letters*, Apr. 2022. 26(4):813–817.
- [197] J. Wang, G. Gui, T. Ohtsuki, et al. Compressive sampled CSI feedback method based on deep learning for FDD massive MIMO systems[J]. *IEEE Transactions on Communications*, Sep. 2021. 69(9):5873–5885.
- [198] Q. Cai, C. Dong, K. Niu. Attention model for massive MIMO CSI compression feedback and recovery[J]. Apr. 2019:1–5.
- [199] M. B. Mashhadi, Q. Yang, D. Gunduz. CNN-based analog csi feedback in FDD MIMO-OFDM systems[C]//*IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2020)*. Barcelona, Spain, May 2020:8579–8583.
- [200] M. B. Mashhadi, Q. Yang, D. Gunduz. Distributed deep convolutional compression for massive MIMO CSI feedback[J]. *IEEE Transactions on Wireless Communications*, Apr. 2021. 20(4):2621–2633.
- [201] J. Guo, L. Wang, F. Li, J. Xue. Csi feedback with model-driven deep learning of massive MIMO systems[J]. *IEEE Communications Letters*, Mar. 2022. 26(3):547–551.
- [202] Z. Liu, M. del Rosario, Z. Ding. A Markovian model-driven deep learning framework for massive MIMO

- CSI feedback[J]. *IEEE Transactions on Wireless Communications*, Feb. 2022. 21(2):1214–1228.
- [203] Z. Yin, W. Xu, R. Xie, et al. Deep CSI compression for massive MIMO: A self-information model-driven neural network[J]. *IEEE Transactions on Wireless Communications*, Oct. 2022. 21(10):8872–8886.
- [204] H. He, C. K. Wen, S. Jin, G. Y. Li. Model-driven deep learning for MIMO detection[J]. *IEEE Transactions on Signal Processing*, Mar. 2020. 68:1702–1715.
- [205] M. Borgerding, P. Schniter, S. Rangan. AMP-inspired deep networks for sparse linear inverse problems[J]. *IEEE Transactions on Signal Processing*, Aug. 2017. 65(16):4293–4308.
- [206] N. Shlezinger, N. Farsad, Y. C. Eldar, A. J. Goldsmith. ViterbiNet: A deep learning based Viterbi algorithm for symbol detection[J]. *IEEE Transactions on Wireless Communications*, May 2020. 19(5):3319–3331.
- [207] N. Shlezinger, R. Fu, Y. C. Eldar. DeepSIC: Deep soft interference cancellation for multiuser MIMO detection[J]. *IEEE Transactions on Wireless Communications*, Feb. 2021. 20(2):1349–1362.
- [208] S. Cammerer, T. Gruber, J. Hoydis, S. ten Brink. Scaling deep learning-based decoding of polar codes via partitioning[C]//*IEEE Global Communications Conference (Globecom'2017)*. Singapore, Dec. 2017:1–6.
- [209] Y. Wei, M. M. Zhao, M. J. Zhao. Channel distribution learning: Model-driven GAN-based channel modeling for IRS-aided wireless communication[J]. *IEEE Transactions on Communications*, Jul. 2022. 70(7):4482–4497.
- [210] N. Larrieu. How can model driven development approaches improve the certification process for UAS?[C]//*International Conference on Unmanned Aircraft Systems (ICUAS'2014)*. Orlando, FL, US-A, May 2014:253–260.
- [211] A. Liu, Z. Huang, M. Li, et al. A survey on fundamental limits of integrated sensing and communication[J]. *IEEE Communications Surveys & Tutorials*, Apr. 2022. 24(2):994–1034.
- [212] S. Wang, H. Liu, P. H. Gomes, B. Krishnamachari. Deep reinforcement learning for dynamic multichannel access in wireless networks[J]. *IEEE Transactions on Cognitive Communications and Networking*, Jun. 2018. 4(2):257–265.
- [213] Y. Sun, M. Peng, S. Mao. Deep reinforcement learning-based mode selection and resource management for green fog radio access networks[J]. *IEEE Internet of Things Journal*, Apr. 2019. 6(2):1960–1971.
- [214] J. Cui, Y. Liu, A. Nallanathan. Multi-agent reinforcement learning-based resource allocation for UAV networks[J]. *IEEE Transactions on Wireless Communications*, Feb. 2020. 19(2):729–743.
- [215] X. Liu, Y. Liu, Y. Chen, H. V. Poor. RIS enhanced massive non-orthogonal multiple access networks: Deployment and passive beamforming design[J]. *IEEE Journal on Selected Areas in Communications*, Apr. 2021. 39(4):1057–1071.
- [216] U. Challita, L. Dong, W. Saad. Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective[J]. *IEEE Transactions on Wireless Communications*, Jul. 2018. 17(7):4674–4689.
- [217] N. Zhao, Y. C. Liang, D. Niyato, et al. Deep reinforcement learning for user association and resource

- allocation in heterogeneous cellular networks[J]. *IEEE Transactions on Wireless Communications*, Nov. 2019. 18(11):5141–5152.
- [218] L. Huang, S. Bi, Y. J. A. Zhang. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks[J]. *IEEE Transactions on Mobile Computing*, Nov. 2020. 19(11):2581–2593.
- [219] F. Sotiraki, K. M. Attiah, W. Yu. Deep learning for distributed channel feedback and multiuser precoding in FDD massive MIMO[J]. *IEEE Transactions on Wireless Communications*, Jul. 2021. 20(7):4044–4057.
- [220] S. Cammerer, F. A. Aoudia, S. Dörner, et al. Trainable communication systems: Concepts and prototype[J]. *IEEE Transactions on Communications*, Sep. 2020. 68(9):5489–5503.
- [221] H. Ye, L. Liang, G. Y. Li, B. H. Juang. Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels[J]. *IEEE Transactions on Wireless Communications*, May 2020. 19(5):3133–3143.
- [222] J. Guo, C. K. Wen, S. Jin. CAnet: Uplink-aided downlink channel acquisition in FDD massive MIMO using deep learning[J]. *IEEE Transactions on Communications*, Jan. 2022. 70(1):199–214.
- [223] Y. Sun, W. Xu, L. Fan, G. Y. Li, G. K. Karagiannidis. AnciNet: An efficient deep learning approach for feedback compression of estimated CSI in massive MIMO systems[J]. *IEEE Wireless Communications Letters*, Dec. 2020. 9(12):2192–2196.
- [224] J. Guo, C. K. Wen, S. Jin. Deep learning-based CSI feedback for beamforming in single- and multi-cell massive MIMO systems[J]. *IEEE Journal on Selected Areas in Communications*, Jul. 2021. 39(7):1872–1884.
- [225] A. A. Lee, Y. S. Wang, Y. W. P. Hong. Deep CSI compression and coordinated precoding for multicell downlink systems[C]//*IEEE Global Communications Conference (GLOBECOM'2020)*. Taipei, Taiwan, Dec 2020:1–6.
- [226] K. Kong, W. J. Song, M. Min. Knowledge distillation-aided end-to-end learning for linear precoding in multiuser MIMO downlink systems with finite-rate feedback[J]. *IEEE Transactions on Vehicular Technology*, Oct. 2021. 70(10):11095–11100.
- [227] J. Jang, H. Lee, S. Hwang, H. Ren, I. Lee. Deep learning-based limited feedback designs for MIMO systems[J]. *IEEE Wireless Communications Letters*, Apr. 2020. 9(4):558–561.
- [228] Z. Qin, X. Tao, J. Lu, G. Y. Li. Semantic communications: Principles and challenges[J]. *arXiv preprint arXiv:2201.01389*, 2021.
- [229] M. Kountouris, N. Pappas. Semantics-empowered communication for networked intelligent systems[J]. *IEEE Communications Magazine*, Jun. 2021. 59(6):96–102.
- [230] G. Shi, Y. Xiao, Y. Li, X. Xie. From semantic communication to semantic-aware networking: Model, architecture, and open problems[J]. *IEEE Communications Magazine*, Aug. 2021. 59(8):44–50.

- [231] E. Boursoulatzé, D. Burth Kurka, D. Gunduz. Deep joint source-channel coding for wireless image transmission[J]. *IEEE Transactions on Cognitive Communications and Networking*, Sep. 2019. 5(3):567–579.
- [232] H. Xie, Z. Qin, G. Y. Li, B. H. Juang. Deep learning enabled semantic communication systems[J]. *IEEE Transactions on Signal Processing*, Apr. 2021. 69:2663–2675.
- [233] J. Xu, B. Ai, W. Chen, et al. Wireless image transmission using deep source channel coding with attention modules[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, Apr. 2022. 32(4):2315–2328.
- [234] K. Lu, R. Li, X. Chen, et al. Reinforcement learning-powered semantic communication via semantic similarity[J]. *arXiv preprint arXiv:2108.12121*, 2021.
- [235] D. B. Kurka, D. Gunduz. DeepJSCC-f: Deep joint source-channel coding of images with feedback[J]. *IEEE Journal on Selected Areas in Information Theory*, May 2020. 1(1):178–193.
- [236] Z. Weng, Z. Qin. Semantic communication systems for speech transmission[J]. *IEEE Journal on Selected Areas in Communications*, Aug. 2021. 39(8):2434–2444.
- [237] P. Jiang, C. K. Wen, S. Jin, G. Y. Li. Deep source-channel coding for sentence semantic transmission with HARQ[J]. *IEEE Transactions on Communications*, Aug. 2022. 70(8):5225–5240.
- [238] C. H. Lee, J. W. Lin, P. H. Chen, Y. C. Chang. Deep learning-constructed joint transmission-recognition for Internet of Things[J]. *IEEE Access*, Jun. 2019. 7:76547–76561.
- [239] M. Jankowski, D. Gunduz, K. Mikołajczyk. Wireless image retrieval at the edge[J]. *IEEE Journal on Selected Areas in Communications*, Jan. 2021. 39(1):89–100.
- [240] J. Shao, Y. Mao, J. Zhang. Learning task-oriented communication for edge inference: An information bottleneck approach[J]. *IEEE Journal on Selected Areas in Communications*, Jan. 2022. 40(1):197–211.
- [241] H. Xie, Z. Qin, X. Tao, K. B. Letaief. Task-oriented multi-user semantic communications[J]. *IEEE Journal on Selected Areas in Communications*, Sep. 2022. 40(9):2584–2597.
- [242] G. Zhang, Q. Hu, Z. Qin, et al. A unified multi-task semantic communication system for multimodal data[J]. *arXiv preprint arXiv:2209.07689*, 2022.
- [243] I. J. Goodfellow, J. Shlens, C. Szegedy. Explaining and harnessing adversarial examples[C]//*International Conference on Learning Representations (ICLR'2015)*. San Diego, CA, USA, Apr. 2015:1–11.
- [244] A. Madry, A. Makelov, L. Schmidt, et al. Towards deep learning models resistant to adversarial attacks[C]//*International Conference on Learning Representations (ICLR'2018)*. Vancouver, BC, Canada, Apr. 2018:1–10.
- [245] N. Papernot, P. McDaniel, S. Jha, et al. The limitations of deep learning in adversarial settings[C]//*IEEE European Symposium on Security and Privacy (EuroS&P'2016)*. Saarbruecken, Germany, Mar. 2016:372–387.

- [246] S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard. DeepFool: a simple and accurate method to fool deep neural networks[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2016). Las Vegas, NV, USA, Jun. 2016:2574–2582.
- [247] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard. Universal adversarial perturbations[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2017). Honolulu, HI, USA, Jul. 2017:86–94.
- [248] B. Kim, Y. E. Sagduyu, K. Davaslioglu, et al. Channel-aware adversarial attacks against deep learning-based wireless signal classifiers[J]. IEEE Transactions on Wireless Communications, Jun. 2022. 21(6):3868–3880.
- [249] A. Kurakin, I. J. Goodfellow, S. Bengio. Adversarial examples in the physical world[C]//International Conference on Learning Representations (ICLR'2017). Toulon, France, Apr. 2017:1–13.
- [250] F. Liao, M. Liang, Y. Dong, et al. Defense against adversarial attacks using high-level representation guided denoiser[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2018). Salt Lake City, UT, USA, Jun. 2018:1778–1787.
- [251] N. Papernot, P. McDaniel, X. Wu, et al. Distillation as a defense to adversarial perturbations against deep neural networks[C]//IEEE Symposium on Security and Privacy (SP'2016). San Jose, CA, USA, May 2016:582–597.
- [252] S. Gu, L. Rigazio. Towards deep neural network architectures robust to adversarial examples[C]//International Conference on Learning Representations (ICLR'2015). San Diego, CA, USA, Apr. 2015:1–9.
- [253] D. Wu, S. Xia, Y. Wang. Adversarial weight perturbation helps robust generalization[C]//Advances in Neural Information Processing Systems (NIPS'2020). Vancouver, Canada, Dec. 2020:2958–2969.
- [254] Y. Bai, Y. Zeng, Y. Jiang, et al. Improving adversarial robustness via channel-wise activation suppressing[C]//International Conference on Learning Representations (ICLR'2021). Singapore, Mar. 2021:1–19.
- [255] W. Chen, B. Zhang, S. Jin, et al. Solving sparse linear inverse problems in communication systems: A deep learning approach with adaptive depth[J]. IEEE Journal on Selected Areas in Communications, Jan. 2021. 39(1):4–17.
- [256] S. Oymak, B. Recht, M. Soltanolkotabi. Sharp time-data tradeoffs for linear inverse problems[J]. IEEE Transactions on Information Theory, Jun. 2018. 64(6):4129–4158.
- [257] A. Graves. Adaptive computation time for recurrent neural networks[J]. arXiv preprint arXiv:1603.08983, 2016.
- [258] K. Wei, A. Aviles-Rivero, J. Liang, et al. Tuning-free plug-and-play proximal algorithm for inverse imaging problems[C]//International Conference on Machine Learning (ICML'2020). PMLR, Jul. 2020:10158–10169.

- [259] T. P. Lillicrap, J. J. Hunt, A. Pritzel, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
- [260] K. K. Nguyen, T. Q. Duong, N. A. Vien, et al. Distributed deep deterministic policy gradient for power allocation control in D2D-based V2V communications[J]. IEEE Access, Nov. 2019. 7:164533–164543.
- [261] C. Qiu, Y. Hu, Y. Chen, B. Zeng. Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications[J]. IEEE Internet of Things Journal, Oct. 2019. 6(5):8577–8588.
- [262] Y. Sun, P. Babu, D. P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning[J]. IEEE Transactions on Signal Processing, Feb. 2017. 65(3):794–816.
- [263] I. Loshchilov, F. Hutter. SGDR: Stochastic gradient descent with warm restarts[J]. arXiv preprint arXiv:1608.03983, 2016.
- [264] J. Tao, J. Chen, J. Xing, et al. Autoencoder neural network based intelligent hybrid beamforming design for mmWave massive MIMO systems[J]. IEEE Transactions on Cognitive Communications and Networking, Sep. 2020. 6(3):1019–1030.
- [265] J. Chung, S. Ahn, Y. Bengio. Hierarchical multiscale recurrent neural networks[C]//International Conference on Learning Representations (ICLR'2017). Toulon, France, Apr. 2017:1–13.
- [266] J. Zhang, M. Haardt, I. Soloveychik, A. Wiesel. A channel matching based hybrid analog-digital strategy for massive multi-user MIMO downlink systems[C]//IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM'2016). Rio de Janeiro, Brazil, Jul. 2016:1–5.
- [267] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani. Deep learning for IoT big data and streaming analytics: A survey[J]. IEEE Communications Surveys & Tutorials, Jun. 2018. 20(4):2923–2960.
- [268] C. Bonchelet. The essential guide to image processing[M]. Academic Press, 2009.
- [269] W. Wang, R. Wang, L. Wang, et al. Towards a robust deep neural network in texts: A survey[J]. arXiv preprint arXiv:1902.07285, 2019.
- [270] C. Szegedy, W. Zaremba, I. Sutskever, et al. Intriguing properties of neural networks[C]//International Conference on Learning Representations (ICLR'2014). Banff, AB, Canada, Apr. 2014:1–10.
- [271] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems (NIPS'2017). Long Beach, USA, Dec. 2017:1–11.
- [272] K. He, X. Chen, S. Xie, et al. Masked autoencoders are scalable vision learners[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2022). New Orleans, LA, USA, Jun. 2022:15979–15988.
- [273] A. van den Oord, O. Vinyals, K. Kavukcuoglu. Neural discrete representation learning[C]//Advances in Neural Information Processing Systems (NIPS'2017). Long Beach, USA, Dec. 2017:1–10.

附录

A.1 矩阵形式模型驱动网络框架：大规模MU-MIMO系统预编码设计

A.1.1 定理2.1的证明

首先，根据文献[23]，给出如下定理A.1用于计算矩阵变量的梯度，引理A.1展示了矩阵微分的部分性质。

定理 A.1. 如果函数 f 关于矩阵变量 \mathbf{X} 的微分有如下形式：

$$df = \text{Tr}(\mathbf{A}d\mathbf{X}^H), \quad (\text{A.1})$$

则函数 f 关于矩阵变量 \mathbf{X}^* 的偏导为：

$$\frac{\partial f}{\partial \mathbf{X}^*} = \mathbf{A}, \quad (\text{A.2})$$

其中 $f: \mathbb{C}^{m \times n} \mapsto \mathbb{R}$ 为变量 $\mathbf{X} \in \mathbb{C}^{m \times n}$ 的函数。

引理 A.1. 矩阵微分的性质：

$$\begin{aligned} d\text{Tr}(\mathbf{X}) &= \text{Tr}(d\mathbf{X}), & d\text{Tr}(\mathbf{X}\mathbf{X}^H) &= \text{Tr}(\mathbf{X}d\mathbf{X}^H + \mathbf{X}^H d\mathbf{X}), \\ d\log \det(\mathbf{X}) &= \text{Tr}(\mathbf{X}^{-1}d\mathbf{X}), & d\text{Tr}(\mathbf{A}\mathbf{X}^{-1}) &= -\text{Tr}(\mathbf{X}^{-1}\mathbf{A}\mathbf{X}^{-1}d\mathbf{X}), \\ d(\mathbf{X} + \mathbf{Y}) &= d\mathbf{X} + d\mathbf{Y}, & d(\mathbf{X}\mathbf{Y}) &= (d\mathbf{X})\mathbf{Y} + \mathbf{X}(d\mathbf{Y}), \end{aligned} \quad (\text{A.3})$$

其中 \mathbf{A} 为常数矩阵， \mathbf{X} 和 \mathbf{Y} 为矩阵变量。

回顾矩阵迹的性质：

$$\text{Tr}(\mathbf{A}\mathbf{B}) = \text{Tr}(\mathbf{B}\mathbf{A}), \quad \text{Tr}(\mathbf{A}^T(\mathbf{B} \circ \mathbf{C})) = \text{Tr}((\mathbf{A}^T \circ \mathbf{B}^T)\mathbf{C}). \quad (\text{A.4})$$

根据定理A.1，引理A.1和矩阵迹的性质(A.4)，可推导出定理2.1中矩阵形式的链式法则。证毕。

A.1.2 输出层矩阵变量的梯度推导

根据定理A.1和引理A.1, 目标函数(2.15)关于变量 \mathbf{W}_k^L 在最后一层的梯度如下所示。为表述清楚起见, 忽略层的脚标 l , 下式中的变量都来自最后一层, 即 $l = L$ 。

$$\begin{aligned}
\frac{\partial f}{\partial \mathbf{W}_k^L} = & - \sum_{m \neq k} \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \mathbf{H}_m^H \tilde{\mathbf{E}}_m \mathbf{H}_m \mathbf{V}_k - \sum_{m=1}^K \text{Tr}(\tilde{\mathbf{A}}_m^{-1} \mathbf{H}_m \tilde{\mathbf{D}}_m \tilde{\mathbf{C}}^{-H} \mathbf{H}_m^H \tilde{\mathbf{B}}_m^{-1}) \frac{\sigma_k^2}{P_T} \mathbf{U}_k^H \mathbf{U}_k \\
& - \sum_{m=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\tilde{\mathbf{E}}_m) \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \mathbf{V}_k - \sum_{m=1}^K \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \mathbf{H}_m^H \tilde{\mathbf{B}}_m^{-1} \tilde{\mathbf{A}}_m^{-1} \mathbf{H}_m \tilde{\mathbf{D}}_m \mathbf{H}_k^H \mathbf{U}_k \\
& + \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \mathbf{H}_k^H \tilde{\mathbf{B}}_k^{-1} \tilde{\mathbf{A}}_k^{-1} \mathbf{H}_k \mathbf{V}_k + \sum_{n=1}^K \sum_{m \neq k} \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \mathbf{H}_n^H \tilde{\mathbf{E}}_n \mathbf{H}_n \tilde{\mathbf{D}}_m \mathbf{H}_k^H \mathbf{U}_k \\
& + \sum_{n=1}^K \sum_{m=1}^K \frac{\sigma_n^2 \sigma_k^2}{P_T^2} \text{Tr}(\tilde{\mathbf{E}}_m) \text{Tr}(\tilde{\mathbf{D}}_n \tilde{\mathbf{C}}^{-H}) \mathbf{U}_k^H \mathbf{U}_k + \sum_{n=1}^K \sum_{m \neq k} \text{Tr}(\tilde{\mathbf{E}}_n \mathbf{H}_n \tilde{\mathbf{D}}_m \tilde{\mathbf{C}}^{-H} \mathbf{H}_n^H) \frac{\sigma_k^2}{P_T} \mathbf{U}_k^H \mathbf{U}_k \\
& + \sum_{n=1}^K \sum_{m=1}^K \frac{\sigma_n^2}{P_T} \text{Tr}(\tilde{\mathbf{E}}_m) \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H} \tilde{\mathbf{D}}_n \mathbf{H}_k^H \mathbf{U}_k,
\end{aligned} \tag{A.5}$$

其中

$$\begin{aligned}
\tilde{\mathbf{A}}_k & \triangleq \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \tilde{\mathbf{B}}_k^{-1}, \\
\tilde{\mathbf{B}}_k & \triangleq \sum_{m \neq k} \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H + \frac{\sigma_k^2}{P_T} \sum_{k=1}^K \text{Tr}(\mathbf{V}_k \mathbf{V}_k^H) \mathbf{I}, \\
\tilde{\mathbf{C}} & \triangleq \sum_{k=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\mathbf{U}_k \mathbf{W}_k \mathbf{U}_k^H) \mathbf{I} + \sum_{m=1}^K \mathbf{H}_m^H \mathbf{U}_m \mathbf{W}_m \mathbf{U}_m^H \mathbf{H}_m, \\
\tilde{\mathbf{D}}_k & \triangleq \mathbf{V}_k \mathbf{W}_k^H \mathbf{U}_k^H \mathbf{H}_k \tilde{\mathbf{C}}^{-H}, \quad \tilde{\mathbf{E}}_k \triangleq \tilde{\mathbf{B}}_k^{-1} \tilde{\mathbf{A}}_k^{-1} \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \tilde{\mathbf{B}}_k^{-1}.
\end{aligned}$$

目标函数关于变量 \mathbf{U}_k^L 在最后一层的梯度可以类似得到。

A.1.3 中间层矩阵变量的梯度推导

第 $(l+1)$ 层矩阵变量的全微分可表示为

$$\sum_{k=1}^K \text{Tr} \left(\mathbf{G}_k^{u,l+1} d\mathbf{U}_k^{l+1} + \mathbf{G}_k^{w,l+1} d\mathbf{W}_k^{l+1} + \mathbf{G}_k^{v,l+1} d\mathbf{V}_k^{l+1} \right). \tag{A.7}$$

以 $\mathbf{G}_n^{w,l}$ 为例， $\mathbf{G}_n^{u,l}$ 和 $\mathbf{G}_n^{v,l}$ 可以类似得到。为了计算 $\mathbf{G}_n^{w,l}$ ，首先将(2.19)代入(A.7)，详细推导请见附录A.1.4。保留与 $d\mathbf{W}_k^l$ 相关的项，基于定理2.1和推论2.1，可以得到如下结果

$$\begin{aligned} & \text{Tr} \left\{ \mathbf{G}_n^{v,l+1} \left(-(\mathbf{B}^{l+1})^+ \circ (\mathbf{B}^{l+1})^+ \circ d(\mathbf{B}^{l+1}) \right) \mathbf{X}_n^{v,l+1} \mathbf{H}_n^H \mathbf{U}_n^{l+1} \mathbf{W}_n^{l+1} \right. \\ & \quad \left. + \mathbf{G}_n^{v,l+1} d(\mathbf{B}^{l+1}) \mathbf{Y}_n^{v,l+1} \mathbf{H}_n^H \mathbf{U}_n^{l+1} \mathbf{W}_n^{l+1} + \mathbf{G}_n^{v,l+1} \mathbf{M}_n^{v,l+1} \mathbf{U}_n^{l+1} d\mathbf{W}_n^{l+1} \right\} \\ & \stackrel{(2.5)}{=} \text{Tr} \left\{ \sum_{k=1}^K \text{Tr}(\mathbf{J}_k^{v,l+1} + \mathbf{L}_k^{v,l+1}) \frac{\sigma_k^2}{P_T} (\mathbf{U}_n^{l+1})^H \mathbf{U}_n^{l+1} d\mathbf{W}_n^{l+1} \right. \\ & \quad \left. + \sum_{k=1}^K (\mathbf{U}_n^{l+1})^H \mathbf{H}_n (\mathbf{J}_k^{v,l+1} + \mathbf{L}_k^{v,l+1}) \mathbf{H}_n^H \mathbf{U}_n^{l+1} d\mathbf{W}_n^{l+1} + \mathbf{G}_n^{v,l+1} \mathbf{M}_n^{v,l+1} \mathbf{U}_n^{l+1} d\mathbf{W}_n^{l+1} \right\}, \end{aligned} \quad (\text{A.8})$$

其中

$$\begin{aligned} d(\mathbf{B}^{l+1}) &= \sum_{k=1}^K \frac{\sigma_k^2}{P_T} \text{Tr}(\omega_k \mathbf{U}_k^{l+1} d\mathbf{W}_k^{l+1} (\mathbf{U}_k^{l+1})^H) \mathbf{I} + \sum_{m=1}^K \omega_m \mathbf{H}_m^H \mathbf{U}_m^{l+1} d\mathbf{W}_m^{l+1} (\mathbf{U}_m^{l+1})^H \mathbf{H}_m, \\ \mathbf{J}_k^{v,l+1} &\triangleq \left(\mathbf{X}_k^{v,l+1} \mathbf{H}_k^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} \mathbf{G}_k^{v,l+1} \right) \circ \left(-(\mathbf{B}^{l+1})^+ \circ (\mathbf{B}^{l+1})^+ \right)^T, \\ \mathbf{L}_k^{v,l+1} &\triangleq \mathbf{Y}_k^{v,l+1} \mathbf{H}_k^H \mathbf{U}_k^{l+1} \mathbf{W}_k^{l+1} \mathbf{G}_k^{v,l+1}, \\ \mathbf{M}_k^{v,l+1} &\triangleq \left((\mathbf{B}^{l+1})^+ \mathbf{X}_k^{v,l+1} + \mathbf{B}^{l+1} \mathbf{Y}_k^{v,l+1} + \mathbf{Z}_k^{v,l+1} \right) \mathbf{H}_k^H. \end{aligned} \quad (\text{A.9})$$

由此可得关于相邻层之间的变量 $\{\mathbf{W}_n\}$ 的递推表达式

$$\mathbf{G}_n^{w,l} = \sum_{k=1}^K \text{Tr}(\mathbf{J}_k^{v,l} + \mathbf{L}_k^{v,l}) \frac{\sigma_k^2}{P_T} (\mathbf{U}_n^l)^H \mathbf{U}_n^l + \sum_{k=1}^K (\mathbf{U}_n^l)^H \mathbf{H}_n (\mathbf{J}_k^{v,l} + \mathbf{L}_k^{v,l}) \mathbf{H}_n^H \mathbf{U}_n^l + \mathbf{G}_n^{v,l} \mathbf{M}_n^{v,l} \mathbf{U}_n^l. \quad (\text{A.10})$$

类似地，可得相邻层之间的变量 $\{\mathbf{U}_n, \mathbf{V}_n\}$ 的递推表达式

$$\begin{aligned} \tilde{\mathbf{G}}_n^{v,l} &= -(\mathbf{J}_n^{w,l+1} + \mathbf{Y}_n^{w,l+1} \mathbf{G}_n^{w,l+1}) (\mathbf{U}_n^{l+1})^H \mathbf{H}_n + \mathbf{M}_n^{u,l+1} \\ & \quad + \sum_{k=1}^K \text{Tr}(\mathbf{N}_k^{u,l+1}) \frac{\sigma_k^2}{P_T} (\mathbf{V}_n^l)^H + \sum_{k=1}^K (\mathbf{V}_n^l)^H \mathbf{H}_k^H (\mathbf{N}_k^{u,l+1}) \mathbf{H}_k, \end{aligned} \quad (\text{A.11a})$$

$$\begin{aligned} \mathbf{G}_n^{u,l} &= -((\mathbf{J}_n^{w,l})^H + (\mathbf{G}_n^{w,l})^H (\mathbf{Y}_n^{w,l})^H) (\mathbf{V}_n^{l-1})^H \mathbf{H}_n^H + \sum_{k=1}^K \mathbf{W}_n^l (\mathbf{U}_n^l)^H \mathbf{H}_n (\mathbf{J}_k^{v,l} + \mathbf{L}_k^{v,l}) \mathbf{H}_n^H \\ & \quad + \sum_{k=1}^K \text{Tr}(\mathbf{J}_k^{v,l} + \mathbf{L}_k^{v,l}) \frac{\sigma_k^2}{P_T} \mathbf{W}_n^l (\mathbf{U}_n^l)^H + \sum_{k=1}^K (\mathbf{W}_n^l)^H (\mathbf{U}_n^l)^H \mathbf{H}_n ((\mathbf{J}_k^{v,l})^H + (\mathbf{L}_k^{v,l})^H) \mathbf{H}_n^H \\ & \quad + \sum_{k=1}^K \text{Tr}((\mathbf{J}_k^{v,l})^H + (\mathbf{L}_k^{v,l})^H) \frac{\sigma_k^2}{P_T} (\mathbf{W}_n^l)^H (\mathbf{U}_n^l)^H + \mathbf{W}_n^l \mathbf{G}_n^{v,l} \mathbf{M}_n^{v,l}, \end{aligned} \quad (\text{A.11b})$$

其中

$$\begin{aligned}
\mathbf{J}_k^{u,l+1} &\triangleq \left(\mathbf{X}_k^{u,l+1} \mathbf{H}_k \mathbf{V}_k^l \mathbf{G}_k^{u,l+1} \right) \circ \left(-(\mathbf{A}_k^l)^+ \circ (\mathbf{A}_k^l)^+ \right)^T, \\
\mathbf{L}_k^{u,l+1} &\triangleq \mathbf{Y}_k^{u,l+1} \mathbf{H}_k \mathbf{V}_k^l \mathbf{G}_k^{u,l+1}, \\
\mathbf{M}_k^{u,l+1} &\triangleq \mathbf{G}_k^{u,l+1} \left((\mathbf{A}_k^l)^+ \mathbf{X}_k^{u,l+1} + \mathbf{A}_k^l \mathbf{Y}_k^{u,l+1} + \mathbf{Z}_k^{u,l+1} \right) \mathbf{H}_k, \\
\mathbf{N}_k^{u,l+1} &\triangleq \mathbf{J}_k^{u,l+1} + \mathbf{L}_k^{u,l+1} + (\mathbf{J}_k^{u,l+1})^H + (\mathbf{L}_k^{u,l+1})^H, \\
\mathbf{J}_k^{w,l+1} &\triangleq \left(\mathbf{X}_k^{w,l+1} \mathbf{G}_k^{w,l+1} \right) \circ \left(-(\mathbf{E}_k^{l+1})^+ \circ (\mathbf{E}_k^{l+1})^+ \right)^T.
\end{aligned} \tag{A.12}$$

在每一层的最后，归一化每个预编码矩阵 \mathbf{V}_k 的功率 P_T 。变量 \mathbf{V} 的梯度为

$$\mathbf{G}_n^{v,l} = \tilde{\mathbf{G}}_n^{v,l} + \tilde{\mathbf{G}}_n^{v,l} \sqrt{P_t} a^{-\frac{1}{2}} - \sum_{m=1}^K \frac{1}{2} \text{Tr} \left(\tilde{\mathbf{G}}_m^{v,l} \mathbf{V}_m^l + (\tilde{\mathbf{G}}_m^{v,l})^H (\mathbf{V}_m^l)^H \right) \sqrt{P_t} a^{-\frac{3}{2}} (\mathbf{V}_n^l)^H, \tag{A.13}$$

其中 $a \triangleq \sum_{k=1}^K \text{Tr}(\mathbf{V}_k^l (\mathbf{V}_k^l)^H)$ 。

A.1.4 关于公式(A.7)-(A.8)的详细推导

本附录展示关于表达式(A.7)-(A.8)的推导细节。回顾 \mathbf{W}_k^{l+1} , \mathbf{E}_k^{l+1} 和 \mathbf{J}_k^{l+1} 的表达式(2.19), (2.20)和(A.12), 以及基于定理2.1和引理A.1推得的结果, 可以得到从变量 \mathbf{W}_k^{l+1} 到变量 \mathbf{V}_k^l 和变量 $(\mathbf{U}_k^{l+1})^H$ 的递推关系式,

$$\begin{aligned}
\text{Tr}(\mathbf{G}_k^{w,l+1} d\mathbf{W}_k^{l+1}) &= \text{Tr} \left\{ \mathbf{G}_k^{w,l+1} \left(-(\mathbf{E}_k^{l+1})^+ \circ (\mathbf{E}_k^{l+1})^+ \circ d\mathbf{E}_k^{l+1} \right) \mathbf{X}_k^{w,l+1} + \mathbf{G}_k^{w,l+1} d\mathbf{E}_k^{l+1} \mathbf{Y}_k^{w,l+1} \right\} \\
&= \text{Tr} \left\{ (\mathbf{J}_k^{w,l+1} + \mathbf{Y}_k^{w,l+1} \mathbf{G}_k^{w,l+1}) d\mathbf{E}_k^{l+1} \right\} \\
&= -\text{Tr} \left\{ (\mathbf{J}_k^{w,l+1} + \mathbf{Y}_k^{w,l+1} \mathbf{G}_k^{w,l+1}) (\mathbf{U}_k^{l+1})^H \mathbf{H}_k d\mathbf{V}_k^l \right. \\
&\quad \left. + \mathbf{H}_k \mathbf{V}_k^l (\mathbf{J}_k^{w,l+1} + \mathbf{Y}_k^{w,l+1} \mathbf{G}_k^{w,l+1}) d(\mathbf{U}_k^{l+1})^H \right\}.
\end{aligned} \tag{A.14}$$

表达式 $\text{Tr}(\mathbf{G}_k^{u,l+1} d\mathbf{U}_k^{l+1})$ 和 $\text{Tr}(\mathbf{G}_k^{v,l+1} d\mathbf{V}_k^{l+1})$ 可以类似得到。随后, 对这些项求和, 并写成(A.7)的形式。最终, 保留与 $d\mathbf{W}_k^l$ 有关的项, 得到(A.8)。

A.2 基于深度强化学习的可变层模型驱动网络：稀疏信道估计

A.2.1 定理3.1的证明

本附录证明定理3.1。以变量 α 的更新公式(3.24)为例，其中 γ 和 β 在变量 α 的更新过程中被视为常数。变量 γ 和 β 的更新可以类似分析。回顾

$$\eta(\alpha, \gamma, \beta) = \text{Tr}(\Phi(\beta)\Sigma(\alpha, \gamma, \beta)\Phi^H(\beta)) + \|\mathbf{y} - \Phi(\beta)\boldsymbol{\mu}(\alpha, \gamma, \beta)\|^2, \quad (\text{A.15})$$

其中

$$\begin{aligned} \boldsymbol{\mu}(\alpha, \gamma, \beta) &= \alpha\Sigma(\alpha, \gamma, \beta)\Phi^H(\beta)\mathbf{y}, \\ \Sigma(\alpha, \gamma, \beta) &= (\alpha\Phi^H(\beta)\Phi(\beta) + \text{diag}(\gamma))^{-1}. \end{aligned} \quad (\text{A.16})$$

为表述清楚起见，后文将 $\Phi(\beta)$ 记作 Φ ，将(A.15)中的 $\eta(\alpha, \gamma, \beta)$ 记作 $\eta(\alpha)$ 。

回顾变量 α 的闭式解

$$\alpha^{t+1} = \frac{T+a}{b+\eta(\alpha^t)}. \quad (\text{A.17})$$

将(A.15)和(A.16)代入(A.17)，获得如下从 α^t 到 α^{t+1} 的映射：

$$\begin{aligned} \alpha^{t+1} &= (T+a) \left(b + \text{Tr}(\Phi(\alpha^t\Phi^H\Phi + \text{diag}(\gamma))^{-1}\Phi^H) \right. \\ &\quad \left. + \|\mathbf{y} - \alpha^t\Phi(\alpha^t\Phi^H\Phi + \text{diag}(\gamma))^{-1}\Phi^H\mathbf{y}\|^2 \right)^{-1}, \end{aligned} \quad (\text{A.18})$$

其中 t 表示迭代的次数。根据(A.18)和矩阵迹的性质： $\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA})$ ，可以得到如下从 α^t 到 α^{t+2} 的映射：

$$\begin{aligned} \alpha^{t+2} &= (T+a) \left(b + \text{Tr}(\Phi^H\Phi(\frac{T+a}{b+\eta(\alpha^t)}\Phi^H\Phi + \text{diag}(\gamma))^{-1}) \right. \\ &\quad \left. + \|\mathbf{y} - \frac{T+a}{b+\eta(\alpha^t)}\Phi(\frac{T+a}{b+\eta(\alpha^t)}\Phi^H\Phi + \text{diag}(\gamma))^{-1}\Phi^H\mathbf{y}\|^2 \right)^{-1}. \end{aligned} \quad (\text{A.19})$$

回顾第 l 层关于变量 α 的模型驱动网络结构：

$$\alpha^{l+1} = (T+a) \left(b + \text{Tr}(\Phi(\Sigma(\alpha, \gamma, \beta) + \mathbf{O}_1)\Phi^H) + \|\mathbf{y} - \Phi(\boldsymbol{\mu}(\alpha, \gamma, \beta) + \mathbf{o}_2)\|_2^2 \right)^{-1}. \quad (\text{A.20})$$

将(A.16)代入(A.20)，得到从 α^l 到 α^{l+1} 的映射：

$$\begin{aligned} \alpha^{l+1} &= (T+a) \left(b + \text{Tr}(\Phi^H\Phi(\alpha^l\Phi^H\Phi + \text{diag}(\gamma))^{-1} + \Phi^H\Phi\mathbf{O}_1) \right. \\ &\quad \left. + \|\mathbf{y} - \alpha^l\Phi((\alpha^l\Phi^H\Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1)\Phi^H\mathbf{y} - \Phi\mathbf{o}_2\|^2 \right)^{-1}. \end{aligned} \quad (\text{A.21})$$

A.2.1.1 确定性信道

对于信道固定的场景，比如在信道相干时间内变化很慢的场景，可以证明存在训练参数 \mathbf{O}_1 和 \mathbf{o}_2 确保下式成立： $\alpha^{t+2} = \alpha^{t+1}$ 。令(A.19)的右侧表达式等于(A.21)的右侧表达式，求解该关于变量 \mathbf{O}_1 和 \mathbf{o}_2 的等式，得到

$$\begin{aligned}\mathbf{O}_1 &= \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1}, \\ \mathbf{o}_2 &= \frac{T+a}{b+\eta(\alpha^t)} \Phi^\dagger \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \\ &\quad - \alpha^l \Phi^\dagger \Phi \left((\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1 \right) \Phi^H \mathbf{y}.\end{aligned}\tag{A.22}$$

A.2.1.2 衰落信道

对于服从某一特定分布的信道，需要证明存在 \mathbf{O}_1 和 \mathbf{o}_2 使得 $\mathbb{E}_{\mathbf{h}}\{\|\alpha^{t+2} - \alpha^{t+1}\|^2\} < \delta$ 满足。根据(A.19)和(A.21)，可以得到

$$\begin{aligned}\mathbb{E}_{\mathbf{h}}\{\|\alpha^{t+2} - \alpha^{t+1}\|\} &= \mathbb{E}_{\mathbf{h}}\{\|(T+a)A^{-1} - (T+a)B^{-1}\|\} \\ &= \mathbb{E}_{\mathbf{h}}\left\{\left\| \frac{(T+a)}{AB} (A-B) \right\|\right\} \\ &\stackrel{(a)}{\leq} \left(\mathbb{E}_{\mathbf{h}}\left\{\left\| \frac{(T+a)}{AB} \right\|^2\right\} \right)^{\frac{1}{2}} \left(\mathbb{E}_{\mathbf{h}}\{\|A-B\|^2\} \right)^{\frac{1}{2}},\end{aligned}\tag{A.23}$$

其中不等式(a)成立是因为著名的Cauchy-Schwarz不等式，令

$$\begin{aligned}A &\triangleq b + \text{Tr}\left(\Phi^H \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1}\right) \\ &\quad + \left\| \mathbf{y} - \frac{T+a}{b+\eta(\alpha^t)} \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right\|^2, \\ B &\triangleq b + \text{Tr}\left(\Phi^H \Phi (\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \Phi^H \Phi \mathbf{O}_1\right) \\ &\quad + \left\| \mathbf{y} - \alpha^l \Phi \left((\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1 \right) \Phi^H \mathbf{y} - \Phi \mathbf{o}_2 \right\|^2.\end{aligned}\tag{A.24}$$

关注 $\mathbb{E}_{\mathbf{h}}\{\|A-B\|^2\}$ ，可以得到下式(A.25)

$$\begin{aligned}\mathbb{E}_{\mathbf{h}}\{\|A-B\|^2\} &= \mathbb{E}_{\mathbf{h}}\left\{ \left\| \text{Tr}\left(\Phi^H \Phi \left(\left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} \right. \right. \right. \\ &\quad \left. \left. \left. - \mathbf{O}_1 \right) \right) + \left\| \mathbf{y} - \frac{T+a}{b+\eta(\alpha^t)} \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right\|^2 \right. \\ &\quad \left. \left. - \left\| \mathbf{y} - \alpha^l \Phi \left((\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1 \right) \Phi^H \mathbf{y} - \Phi \mathbf{o}_2 \right\|^2 \right\|^2 \right\}\end{aligned}\tag{A.25a}$$

$$\begin{aligned}
&\stackrel{(b)}{\leq} \mathbb{E}_{\mathbf{h}} \left\{ \left\| \text{Tr} \left(\left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} - \mathbf{O}_1 \right) \right\|^2 \right\} \\
&\quad + \mathbb{E}_{\mathbf{h}} \left\{ \left\| \frac{T+a}{b+\eta(\alpha^t)} \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right. \right. \\
&\quad \left. \left. - \alpha^l \Phi \left((\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1 \right) \Phi^H \mathbf{y} - \Phi \mathbf{o}_2 \right\|^2 \right\} \tag{A.25b}
\end{aligned}$$

不等式(b)成立是因为Cauchy-Schwarz不等式，三角不等式和绝对值不等式。随后，令

$$\mathbf{O}_1 = \mathbb{E}_{\mathbf{h}} \left\{ \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} \right\}. \tag{A.26}$$

接着，提出如下引理A.2

引理 A.2. 根据大数定律，当信道样本数量足够大，公式(A.25b)的第一项将会收敛到0。

引理A.2的详细证明请见附录A.2.2。因此， δ 仅取决于(A.25b)的第二项

$$\begin{aligned}
&\mathbb{E}_{\mathbf{h}} \left\{ \left\| \frac{T+a}{b+\eta(\alpha^t)} \Phi \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right. \right. \\
&\quad \left. \left. - \alpha^l \Phi \left((\alpha^l \Phi^H \Phi + \text{diag}(\gamma))^{-1} + \mathbf{O}_1 \right) \Phi^H \mathbf{y} - \Phi \mathbf{o}_2 \right\|^2 \right\} \\
&\stackrel{(c)}{\leq} \mathbb{E}_{\mathbf{h}} \left\{ \left\| \frac{T+a}{b+\eta(\alpha^t)} \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right. \right. \\
&\quad \left. \left. - \alpha^l \mathbb{E}_{\mathbf{h}} \left\{ \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \right\} \Phi^H \mathbf{y} - \mathbf{o}_2 \right\|^2 \right\}, \tag{A.27}
\end{aligned}$$

其中不等式(c)成立是通过代入(A.26)中关于 \mathbf{O}_1 的表达式，并根据绝对值不等式。随后，令

$$\begin{aligned}
\mathbf{o}_2 = \mathbb{E}_{\mathbf{h}} \left\{ \left\| \frac{T+a}{b+\eta(\alpha^t)} \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \Phi^H \mathbf{y} \right. \right. \\
\left. \left. - \alpha^l \mathbb{E}_{\mathbf{h}} \left\{ \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} \right\} \Phi^H \mathbf{y} \right\|^2 \right\}. \tag{A.28}
\end{aligned}$$

根据大数定律，当信道样本足够多时，(A.27)收敛到0，具体证明可参照引理A.2的证明。因此，存在训练参数确保 α^{t+2} 和 α^{l+1} 的差距小于一个足够小的数 δ ，即 $\mathbb{E}_{\mathbf{h}} \{ \|\alpha^{t+2} - \alpha^{l+1}\| \} < \delta$ 。证毕。

A.2.2 引理A.2的证明

本附录证明引理A.2。回顾大数定律

定理 A.2. 记 X 为服从某一特定分布的随机变量, $x_n, n = 1, 2, \dots, N$ 是 N 个 X 的样本。数量足够多的样本的平均, 即 $\frac{1}{N} \sum_{n=1}^N x_n$, 无限逼近随机变量 X 的期望, 即 $\mathbb{E}(X)$, 并随着样本数量 N 的增加越趋近于 $\mathbb{E}(X)$, 可以被描述为

$$\lim_{N \rightarrow \infty} P \left(\left\| \frac{1}{N} \sum_{n=1}^N x_n - \mathbb{E}(X) \right\| < \varepsilon \right) = 1, \quad (\text{A.29})$$

其中 $\varepsilon > 0$ 是一个足够小的数。

回顾(A.25b)的第一项

$$\mathbb{E}_{\mathbf{h}} \left\{ \left\| \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^t \Phi^H \Phi + \text{diag}(\gamma))^{-1} - \mathbf{O}_1 \right\|^2 \right\}. \quad (\text{A.30})$$

记

$$F(\mathbf{h}) \triangleq \left(\frac{T+a}{b+\eta(\alpha^t)} \Phi^H \Phi + \text{diag}(\gamma) \right)^{-1} - (\alpha^t \Phi^H \Phi + \text{diag}(\gamma))^{-1}, \quad (\text{A.31})$$

是输入信道样本 \mathbf{h} 的函数。随后, 将训练参数 \mathbf{O}_1 设为 $\mathbf{O}_1 = \mathbb{E}_{\mathbf{h}}\{F(\mathbf{h})\}$ 。接着, (A.30)可被写成

$$\mathbb{E}_{\mathbf{h}} \left\{ \left\| F(\mathbf{h}) - \mathbb{E}_{\mathbf{h}}\{F(\mathbf{h})\} \right\|^2 \right\}. \quad (\text{A.32})$$

使用足够数量的训练样本训练模型驱动网络, 其性能由一批测试样本的平均性能进行评估。因此, 模型驱动网络中的 $F(\mathbf{h})$ 可由一批样本对应函数值的平均值近似, 即 $\frac{1}{N} \sum_{n=1}^N F(\mathbf{h}_n)$, 其中 N 表示足够多样本的数量。由此, 可将公式(A.32)写作

$$\mathbb{E}_{\mathbf{h}} \left\{ \left\| \frac{1}{N} \sum_{n=1}^N F(\mathbf{h}_n) - \mathbb{E}_{\mathbf{h}}\{F(\mathbf{h})\} \right\|^2 \right\}. \quad (\text{A.33})$$

根据大数定律(A.29), 可以得到

$$\lim_{N \rightarrow \infty} P \left(\left\| \frac{1}{N} \sum_{n=1}^N F(\mathbf{h}_n) - \mathbb{E}_{\mathbf{h}}\{F(\mathbf{h})\} \right\|^2 < \varepsilon^2 \right) = 1. \quad (\text{A.34})$$

因此, (A.32)将依概率1收敛到0。证毕。

A.3 基于联合强化学习和模型驱动网络的波束选择和预编码设计

A.3.1 强化学习的策略梯度

强化学习中的策略梯度可以用于高效求解具有高维度状态 \mathcal{S} 和复杂奖励函数 \mathcal{R} 的问题。首先, 定义一类参数化的策略: $\Lambda = \{\pi_{\theta}, \theta \in \mathbb{R}^m\}$, 该随机策略服从某一分布。对于每一

个策略，定义奖励函数

$$J(\boldsymbol{\theta}) = \mathbb{E} \left[\sum_{t \geq 0} \gamma^t r_t | \pi_{\boldsymbol{\theta}} \right]. \quad (\text{A.35})$$

旨在找到最优策略 $\boldsymbol{\theta}^* = \arg \max_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ 。可以将公式(A.35)写成数学形式

$$J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau}; \boldsymbol{\theta})} [r(\boldsymbol{\tau})] = \int_{\boldsymbol{\tau}} r(\boldsymbol{\tau}) p(\boldsymbol{\tau}; \boldsymbol{\theta}) d\boldsymbol{\tau}, \quad (\text{A.36})$$

其中 $r(\boldsymbol{\tau})$ 表示对一系列状态和动作轨迹 $\boldsymbol{\tau} = (s_0, a_0, s_1, a_1, \dots)$ 的奖励函数。 $J(\boldsymbol{\theta})$ 关于变量 $\boldsymbol{\theta}$ 的梯度为

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \int_{\boldsymbol{\tau}} (r(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}; \boldsymbol{\theta})) p(\boldsymbol{\tau}; \boldsymbol{\theta}) d\boldsymbol{\tau} = \mathbb{E}_{\boldsymbol{\tau} \sim p(\boldsymbol{\tau}; \boldsymbol{\theta})} [r(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}; \boldsymbol{\theta})], \quad (\text{A.37})$$

其中

$$p(\boldsymbol{\tau}; \boldsymbol{\theta}) = \prod_{t \geq 0} p(s_{t+1} | s_t, a_t) \pi_{\boldsymbol{\theta}}(a_t | s_t). \quad (\text{A.38})$$

对上式做等价变换并求导可得： $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\tau}; \boldsymbol{\theta}) = \sum_{t \geq 0} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t | s_t)$ 。随后，通过对状态动作轨迹 $\boldsymbol{\tau}$ 的采样估计 $J(\boldsymbol{\theta})$ 的梯度

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \approx \sum_{t \geq 0} r(\boldsymbol{\tau}) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t | s_t). \quad (\text{A.39})$$

因此，(A.37)可以通过蒙特卡洛采样(A.39)进行估计。为了降低(A.39)的估计方差，采用如下对(A.37)改进的估计方法

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \approx \sum_{t \geq 0} \left(\sum_{t' \geq t} \gamma^{t'-t} r_{t'} - b(s_t) \right) \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_t | s_t), \quad (\text{A.40})$$

其中 $b(s_t)$ 为依赖于状态 s_t 的价值函数。

A.3.2 启发式算法

本附录简要介绍两种启发式波束选择算法：IA算法和MS算法。

A.3.2.1 IA算法

首先，将基站到用户 k 的波束空间信道矢量 \mathbf{h}_k 每个元素的模值进行排序，记模值最大的波束为 $b_k^* \in \{1, 2, \dots, M_s\}$ 。当该用户 k 模值最大的波束对应的标号 b_k^* 不同于其他用户模值最大的波束对应的标号，则该用户 k 被定义为非干扰用户。对于非干扰用户，选择其模值最大的波束 b_k^* 。此外，当该用户 k 模值最大的波束对应的标号 b_k^* 和其他某个用户模

值最大的波束对应的标号相同，则该用户被定义为干扰用户。对于干扰用户，搜索波束集合中除了在前述步骤中被非干扰用户已经选择的波束。具体而言，选择对系统和速率性能贡献最大，即可以使得系统性能增益最明显的波束。关于该算法每一步骤详细的细节请参考文献[126]。图4.3提供了一个关于该算法的可视化示例。需要针对4个用户从其包含8个波束的波束空间信道中选择4个波束。用户1和2为非干扰用户，选择其模值最大的波束，即 $b_1^* = 1$ 和 $b_2^* = 7$ 。用户3和4为干扰用户，其模值最大的波束对应的标号相同，即 $b_3^* = b_4^* = 4$ 。为用户3选择其模值最大的波束，为用户4选择其模值第二大的波束。

A.3.2.2 MS算法

文献[129]提出MS-S波束选择算法，旨在通过搜索的方式选择能够最大化SINR的波束。此外，采用低复杂度迫零算法 $\mathbf{P} = \mathbf{H}(\mathbf{H}^H\mathbf{H})^{-1}$ 计算预编码矩阵，从而得到SINR。

作者简介

教育经历

2018.09 - 至今 博士研究生，信息与通信工程，信息与电子工程学院，浙江大学

2014.09 - 2018.06 工学学士，信息工程，信息与电子工程学院，浙江大学

主要研究方向

端到端深度学习在物理层通信的应用，语义通信，无人机通信，模型驱动网络，MIMO通信信号处理，优化理论

参与的研究项目

1. 国家基金委：面向高效能毫米波通信系统的多尺度混合收发机方法研究。
2. 浙江省杰青：毫米波MIMO系统中高效混合收发机优化算法研究。
3. XX公司：基于波形/预编码技术的认知频谱重用；多频段协同定位研究。

项目资助

1. 国家留学基金委公派留学奖学金（项目编号/资助序号：202006320382）。
2. 浙江大学博士研究生学术新星（项目编号/资助序号：2022068）。

奖励荣誉

2023年 浙江省优秀毕业生、浙江大学优秀毕业生
2022年 博士研究生国家奖学金、浙江大学优秀研究生
2021年 浙江大学中电莱斯奖学金、月轮奖学金、优秀研究生
2020年 研电赛全国三等奖、浙江大学优秀研究生
2019年 浙江大学国睿奖学金、三好研究生、优秀研究生

攻读博士学位期间的研究成果

已发表的期刊论文：

1. **Q. Hu**, Y. Cai, G. Yu, Z. Qin, M. Zhao, G. Y. Li, “Joint offloading and trajectory design for UAV-enabled mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879-1892, Apr. 2019. (ESI高被引论文)
2. **Q. Hu**, Y. Cai, A. Liu, G. Yu, G. Y. Li, “Low-complexity joint resource allocation and trajectory design for UAV-aided relay networks with the segmented ray-tracing channel model,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6179-6195, Sep. 2020.
3. **Q. Hu**, Y. Cai, Q. Shi, K. Xu, G. Yu, Z. Ding, “Iterative algorithm induced deep-unfolding neural networks: precoding design for multiuser MIMO systems,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1394-1410, Feb. 2021. (正文第二章)
4. **Q. Hu**, Y. Liu, Y. Cai, G. Yu, Z. Ding, “Joint deep reinforcement learning and unfolding: Beam selection and precoding for mmWave multiuser MIMO with lens arrays,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2289-2304, Aug. 2021. (正文第四章)
5. **Q. Hu**, Y. Cai, K. Kang, G. Yu, J. Hoydis, Y. C. Eldar, “Two-timescale end-to-end learning for channel acquisition and hybrid precoding,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 163-181, Jan. 2022. (正文第五章)
6. **Q. Hu**, S. Shi, Y. Cai, G. Yu, “DDPG-driven deep-unfolding with adaptive depth for channel estimation with sparse Bayesian learning,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 4665-4680, Sep. 2022. (正文第三章)
7. **Q. Hu**, G. Zhang, Z. Qin, Y. Cai, G. Yu, G. Y. Li, “Robust semantic communications with masked VQ-VAE enabled codebook,” *IEEE Transactions on Wireless Communications*, early access, 2023. (正文第六章)

8. Y. Liu, **Q. Hu**, Y. Cai, G. Yu, G. Y. Li, “Deep-unfolding beamforming for intelligent reflecting surface assisted full-duplex systems,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 7, pp. 4784-4800, Jul. 2022.
9. K. Kang, **Q. Hu**, Y. Cai, G. Yu, J. Hoydis, Y. C. Eldar, “Mixed-timescale deep-unfolding for joint channel estimation and hybrid beamforming,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2510-2528, Sep. 2022.
10. Y. Liu, **Q. Hu**, Y. Cai, M. Juntti, “Latency minimization in intelligent reflecting surface assisted D2D offloading systems,” *IEEE Communications Letters*, vol. 25, no. 9, pp. 3046-3050, Sep. 2021.

已发表的会议论文：

1. **Q. Hu**, Y. Cai, A. Liu, G. Yu, “Joint resource allocation and trajectory optimization for UAV-aided relay networks,” *IEEE Global Communications Conference (Globecom'2019)*, Waikoloa, HI, USA, Dec. 2019, pp. 1-6.
2. **Q. Hu**, Y. Liu, Y. Cai, G. Yu, “Deep learning based joint beam selection and precoding design for mmWave systems with lens arrays,” *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'2021)*, Helsinki, Finland, Sep. 2021, pp. 591-597. (正文第四章)
3. **Q. Hu**, S. Shi, Y. Cai, G. Yu, “Sparse Bayesian learning for channel estimation: A DDPG-driven deep-unfolding approach with adaptive depth,” *IEEE International Conference on Communications (ICC'2022)*, Seoul, Korea, May 2022, pp. 1282-1287. (正文第三章)
4. **Q. Hu**, G. Zhang, Z. Qin, Y. Cai, G. Yu, G. Y. Li, “Robust semantic communications against semantic noise,” *IEEE Vehicular Technology Conference (VTC-Fall'2022)*, London, United Kingdom, 2022, pp. 1-6. (正文第六章)
5. K. Kang, **Q. Hu**, Y. Cai, G. Yu, J. Hoydis, Y. C. Eldar, “Joint channel estimation and hybrid beamforming via deep-unfolding,” *IEEE European Signal Processing Conference (EUSIP-CO'2022)*, Belgrade, Serbia, Aug. 2022, pp. 658-662.
6. G. Zhang, **Q. Hu**, Z. Qin, Y. Cai, G. Yu, “A unified multi-task semantic communication system with domain adaptation,” *IEEE Global Communications Conference (Globecom'2022)*, Rio de Janeiro, Brazil, Dec. 2022, pp. 3971-3976.

已授权/受理的发明专利：

1. **胡棋昱**，蔡云龙，康凯，基于双时间尺度和深度学习的天线系统预编码方法和装置：中国，已授权，ZL202111056767.2，2021-09-09.
2. 康凯，**胡棋昱**，蔡云龙，一种基于混合预编码和信道估计的联合设计方法和装置：中国，已受理，202210167814.9，2022-02-24.
3. 张光义，**胡棋昱**，蔡云龙，赵民建，基于图像预测性能的自适应信道反馈方法和装置：中国，已受理，202310110039.8，2023-02-14.