决策树码率自适应算法的无数据蒸馏框架

黄天驰 李朝阳 张睿霄 李文哲 孙立峰
(清华大学计算机科学与技术系 北京 100084)

摘 要 码率自适应(Adaptive Bit-Rate,ABR)算法是流媒体视频传输中至关重要的技术.该算法根据当前网络情况和播放状态等因素,为下一个视频块选择合适的码率,以确保用户获得良好的体验质量(QoE).其中,基于学习的 ABR 算法因其不依赖传统模型和从头学习策略的特点,表现出良好的性能,并逐渐取代需要繁琐调优的启发式 ABR 算法,成为研究领域的热点.然而,这些算法使用神经网络推理,导致模型参数较多,整体计算量较大,使得在实际场景中难以部署.因此,以往的研究提出了决策树蒸馏方案,即使用轻量级的决策树来提取基于学习的 ABR 算法的专家策略,并在线上部署这些决策树.然而,本文的实验结果表明,过去的蒸馏框架忽略了训练环境对蒸馏后策略的影响,导致策略的泛化能力较差.因此,本文提出了一种名为 NIA(data-free Network-environmental Imitation-based rate Adaptation framework)的新型无数据蒸馏框架,用于生成具有更好泛化性能的决策树 ABR 算法.NIA 通过网络环境生成模块构建多个人工网络环境,并在每次迭代训练前使用环境选择模块来选择适合的网络场景,然后与该场景进行交互,利用基于学生驱动的模仿学习算法完成决策树的蒸馏过程.本文还设计了完整的评测平台测试 NIA 的性能.实验表明,NIA 在各种带宽数据集上展现出良好的 QoE 性能和泛化性能:(1)相较于启发式算法,在 QoE 指标上提升了 1%~46%;(2)与以往的决策树蒸馏方案相比,在低带宽场景下表现相当,但在高带宽场景下提升了近 1 倍;(3)总体性能接近甚至超过基于学习的算法(即专家策略)的表现.

关键词 流媒体;码率自适应算法;无数据蒸馏
中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2024.00113

A Data-Free Distillation Framework for Adaptive Bitrate Algorithms

HUANG Tian-Chi LI Chao-Yang ZHANG Rui-Xiao LI Wen-Zhe SUN Li-Feng
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Adaptive Bit-Rate (ABR) algorithm is a key technique in streaming video transmission. The algorithm selects an appropriate bit-rate for the next video chunk based on the current network conditions and playback status to ensure a high-quality user experience (QoE). Among them, learning-based ABR algorithms, due to their characteristic of bypassing traditional modeling and learning strategies from scratch, have achieved better performance and gradually replaced heuristic ABR algorithms that require careful tuning, becoming a research hotspot in the field. However, these algorithms use neural network inference, which results in a large number of model parameters and high overall computational overhead, making it difficult to deploy them in real-world scenarios. Therefore, previous works proposed decision tree distillation schemes, which utilize lightweight decision trees to distill expert policies from learning-based ABR algorithms and deploy them online. However, the experiments in the paper show that the previous distillation framework overlooked the influence of training environments on the distilled policies, resulting in poor generalization capability. The paper proposes NIA (data-free Network-environmental Imitation-based

收稿日期:2022-11-16;在线发布日期:2023-07-12.本课题得到国家自然科学基金(61936011)资助.**黄天驰**(通信作者),博士,主要研究方向为智能流媒体传输、CDN 调度、深度强化学习.E-mail: htc19@mails.tsinghua.edu.cn.**李朝阳**,硕士研究生,主要研究方向为流媒体传输.**张睿霄**,博士,主要研究方向为 CDN 调度、深度学习.**李文哲**,硕士,主要研究方向为联邦学习.**孙立峰**,博士,教授,主要研究领域为边缘计算、视频传输、CDN 调度、联邦学习、视频分析等.

rate Adaptation framework), a novel data-free distillation framework for generating decision tree ABR algorithms with better generalization. NIA generates and selects suitable artificial network environments for distilling decision tree policies. Specifically, NIA uses a network environment generation module to construct multiple artificial network environments and, before each iteration of training, leverages an environment selection module to choose appropriate network scenarios. Prior to each training iteration, the module chooses suitable network environments for distilling teacher policies using a student model. This process is modeled as a "no-regret online learning" problem in reinforcement learning. In detail, the module utilizes the Upper Confidence Bound (UCB) algorithm with the Top-K approach to select network environments from the pool generated by the network environment generation module based on the current performance indicators of the student model and the environment. The selection process ensures performance improvement while maintaining lower bounds. NIA then interacts with the selected scenario to complete the decision tree distillation process based on a student-driven imitation learning algorithm. Based on the observed current state, the student model selects an appropriate bit-rate, while the state is input to the teacher model to obtain expert policies. Subsequently, the module distills the policies by reducing the distance between the student model's strategy and the expert strategy. The paper designs a comprehensive evaluation platform to test the performance of NIA. The data demonstrates that NIA exhibits good QoE performance and generalization performance on various bandwidth datasets: (1) Compared to heuristic algorithms, it improves QoE metrics by 1% to 46%; (2) Compared to previous decision tree distillation schemes, it performs equally well in low-bandwidth scenarios but achieves nearly twice the improvement in high-bandwidth scenarios; (3) Its overall performance is close to or even surpasses that of learning-based algorithms (i. e., expert policies). Finally, the paper carefully analyzes and compares the important parameter settings within the three modules of NIA, including the impact of student model and network environment parameters on NIA's performance, the number of environment generations, the Top-K ratio, the selection algorithm, and the exploration index in the algorithm. Through extensive experiments, each parameter of NIA is appropriately set within the correct range to achieve better results.

Keywords video streaming; adaptive bitrate algorithm; data-free distillation

1 引 言

随着网络的普及以及用户对视频内容的需求日益提升,流媒体视频服务比之前更加重要[1],流媒体视频产业迎来了巨大增长.据《2022 年全球互联网现象报告》[2]显示,在 2021 年 1 月到 6 月期间,全球流媒体流量占带宽总量的 53.72%,其中 YouTube[3]、Netflix[4]以及 Facebook[5]位居前三.无一例外,这三家著名的视频网站都使用自适应流媒体架构(Adaptive Video Streaming)向用户提供视频服务,使用户获得更高质量的用户体验(Quality of Experience,QoE).

自适应流媒体架构的核心是码率自适应(Adaptive BitRate,ABR)算法.ABR 算法通过选择不同码率的视频块适应变化的带宽网络[6].目前学术界已经提出了多种基于客户端的码率自适应(ABR)算法,主要分为启发式算法与基于学习的算法.具体来说,启发式 ABR 算法通常通过传统的预测加控制方案做决策,例如预测吞吐量[7]、利用调整缓冲区占用[8-9],或利用预先 ABR 流程建模在"虚拟播放器"中完成多步决策[10].但是,这些方案采用固定的参数或模型进行决策,使算法无法在所有考虑到的场景中顺畅工作[11].于是,研究者们借助深度学习无需手工设计,直接从数据中预测或泛化策略的特性,提出了基于学习的 ABR 算法,大幅度改善了用户体验.例如基于强化学习获取更卓越的 QoE 性能[11]、基于模仿学习快速训练神经网络策略[12]、与传统方案结合提升性能减少开销[13-14]以及自我强化学习满足多样化的 QoE 需求[15]等(第 2 章).

尽管基于学习的 ABR 算法获得了优越的性能，但是其部署困难，性能开销大等缺点制约了算法在真实场景落地. Meng 等人[16-17]提出了使用决策树算法(即学生模型)蒸馏已训练完成的神经网络模型(即教师模型)，并直接在真实场景中部署决策树模型. 然而，本文通过在各种网络环境及场景下运行训练完成的决策树模型策略，表明过去的这些工作[16]仅在接近训练集分布的网络场景下表现良好(即低带宽场景)，但在远离数据分布的场景下(即高带宽场景)表现较差. 其原因在于原先工作的训练流程并未考虑训练集的网络环境对蒸馏出的决策树算法造成的影响，导致训练的算法策略有可能生成了“在训练网络下特定的策略”，而非“所有网络环境下都适合的普适策略”. 针对过去的决策树蒸馏算法的缺陷，本文提出的核心思想为无数据(Data-free)蒸馏，即不使用真实网络数据，仅构建“人工”的网络环境，通过学生模型在虚拟环境中交互并请求教师模型的专家策略，最终得到泛化性较好的轻量级决策树 ABR 算法. 进一步分析表明，完成 Data-free 蒸馏不仅需要考虑生成多样网络场景，还要在每次训练前挑选合适的场景供学生模型训练，更要设计高效的蒸馏架构(第 3 章).

为此，本文提出了 NIA(data-free Network-environmental Imitation-based rate Adaptation framework)来解决以上挑战. NIA 是一个 Data-free 的轻量级码率自适应算法蒸馏框架. NIA 主要由三大模块组成，分别解决以上提到的三个挑战. 具体而言，这三模块为网络环境生成模块、环境选择模块和学生蒸馏模块. 首先，网络生成模块主要用于构建丰富多样的网络环境. 详细地说，模块通过随机设置一系列网络带宽的“关键参数”生成大量全面且贴近真实场景的网络环境——宏观看既有隐马尔科夫迁移性，局部看又有随机波动性. 其次，每次迭代训练之前，环境选择模块会选择适合的网络环境供学生模型蒸馏教师策略. 本模块将该过程建模为强化学习中的“无后悔的在线学习(no-regret online learning)”问题[18]，并采用了 Top-K 下的多臂赌博机(UCB)算法根据当前学生与环境的性能指标在网络生成模块生成的网络环境池中选择，在保证下限的情况下完成性能的提升. 最后，NIA 利用学生蒸馏模块训练决策树. 与以往决策树蒸馏工作[17]不同的是，该模块采用学生驱动的模仿学习算法，使用学生模型(即决策树)的策略与选择模块挑选的网络环境交互，根据观测到的当前状态，选择合适的码率. 与此同时，将状态输入教师模型获取专家策略. 随后，模块通过拉近学生模型策略与专家策略的距离完成策略蒸馏，这里可以使用任意的决策树优化算法，例如 CART[19]、ID3 以及 C4.5[20]等(第 4 章).

为了更好地分析 NIA 是否能够蒸馏出合适的轻量级策略，我们构建了一个完整的评测平台，包括虚拟播放器、视频测试集、网络环境测试集与多个对比算法. 通过一系列测试得出以下结论. 首先，NIA 的性能在低带宽与高带宽网络环境下均优于过去提出的启发式算法. 其中，在低带宽场景下，与最佳的启发式算法 RobustMPC[10]相比，NIA 的整体 QoE 性能在测试的 3 个网络数据集下各提升了 1.32%～6.81%；与 BBA 算法[8]相比，NIA 的 QoE 性能提升约 46%. 同时，在高带宽场景下，NIA 与决策树蒸馏算法 Metis[17]相比大幅度提升了策略的泛化能力，其 QoE 性能在 HSR 网络数据集下提升了近 1 倍. 此外，在该场景下，NIA 表现得与其他启发式算法相当. 其次，通过研究 NIA 在训练过程中的归一化 QoE 曲线，得出虽然 NIA 未“看见”过真实网络的场景，但是其在各种网络场景下都取得了接近教师模型策略的结果. 值得一提的是，NIA 甚至在部分网络数据集例如 HSDPA 与 FCC 上超越了教师模型，即基于学习的算法 Pensieve[11]提升了 0.91%～2.41%. 相反虽然 Metis 在低带宽场景能够蒸馏教师策略，但是其在高带宽场景下最终收敛到了次优策略. 最后，本文深入研究了每个 ABR 算法的平均视频码率与平均卡顿率. 实验结果表明 NIA 在考虑的所有场景中都获得了 Top-3 的性能表现且相对稳定. 与教师模型相比，NIA 学到的学生策略在相同平均视频码率的情况下获得了更低的平均卡顿率，并有适应多个场景的泛化能力(第 5 章).

在剩下的章节，本文仔细分析并比较了 NIA 的三个模块中的重要的参数设置，包括学生模型与网络环境的参数对 NIA 性能的影响、环境生成数、Top-K 的比例、选择器算法、算法中的探索指数等. 通过大量的实验，NIA 的每一个参数被设置在正确的范围，从而取得更佳的结果(第 6 章).

总结而言，本文的贡献如下：

(1) 本文首次指出了在 ABR 场景中决策树蒸馏任务下泛化指标的重要性，并通过实验发现过去的 ABR 蒸馏工作忽略的核心因素(第 3 章).

(2) 基于发现的问题，提出了 NIA，一种 Data-free 的轻量级 ABR 训练框架，解决了如何在多样的“人造网络场景”中学习到有用的策略的同时，提升其泛化性(第 4 章).

(3) 本文构建了完整评测平台，在低带宽和高

带宽网络场景全面测试了 NIA 蒸馏的策略性能. 实验结果展示了 NIA 在各种网络场景下的泛化能力(第 5 章、第 6 章).

2 相关工作

2.1 码率自适应相关算法

自适应码率算法(Adaptive BitRate,ABR)是一种通过网络信息和当前视频播放信息一起选择视频码率的流媒体传输算法. 传统的视频流媒体系统由一个固定最大缓冲区长度的视频客户端和内容分发网络(CDN)组成. 在视频正式被传输前,文件将被转码为不同的码率档位(即低清中清高清等)并分割成多个视频块存在 CDN 中. 二档流媒体服务启动后,客户端将通过 ABR 算法有序地从 CDN 获取视频块存入缓冲区内,播放器从缓冲区解码数据为画面并渲染到屏幕上. 一般来说,这些 ABR 算法通过吞吐量估计和当前缓冲区利用率为下一个块选择合适的视频码率块. 每次播放结束后,客户端将统计总视频、总缓冲时长和总体的码率变化等几个指标,并将其汇总为 QoE 指标来评估性能. 码率自适应算法根据实现原理可以分类为启发式和基于学习两种.

启发式码率自适应算法. 码率自适应算法(ABR)首先被设计为启发式的——基于启发式的 ABR 算法通常根据观察到的特征结合领域知识推测下一个视频块的码率,其使用的特征包括带宽或者缓冲大小. 例如,Festive[21] 和 PANDA[7] 等工作通过过去多个测量的带宽预测未来带宽,并根据预测的带宽大小直接为下一个视频块选择合适码率. BBA[8] 和 BOLA[22] 则根据当前的缓冲时长选择合适码率,旨在保持整体缓冲时长稳定在一定的范围. 然而,这些方法通常需要精确的带宽估计算法,否则会在长期带宽波动的网络情况下表现不佳. 故 MPC[10] 联合考虑了当前带宽情况和缓冲大小,并通过离线的 ABR 过程建模选择码率. 然而,MPC 对其参数的设置敏感,需要根据不同的网络条件的精心调整各个参数. 针对上述缺点,Oboe[23] 是一种自动调整方法,用于调整传统的启发式方法,以在不同的网络设置下实现更好的性能. 然而,如果当前的网络条件不满足所提出的 ABR 算法基本原理的假设,这种启发式算法将表现不稳定.

基于学习的码率自适应算法. 为了解决启发式算法的缺陷,基于学习的 ABR 算法直接将原始观测信息(例如带宽和缓冲等)作为输入,从零开始利用深度强化学习算法[11,24] 或模仿学习算法[12] 训练 ABR 策略,其中其策略载体通常是一个深度神经网络. Pensieve[11] 是一种基于强化学习算法训练的 ABR 算法,其奖励函数为码率,卡顿时长与码率变化的简单加权和. Stick[14] 融合了强化学习和缓冲区的 ABR 方法,不仅可以实现更高的性能,还可以减少计算开销. 此外,为了使基于学习的 ABR 方案更实用,Meng 等人[17] 提出 Pitree 将 ABR 策略提炼成基于决策树的模型. 同时,Lumos[25] 利用回归树根据网络的特征进行准确吞吐量预测,并使用 MPC 算法选择下一个码率,从而获得更好的 QoE 性能. 此类方法在离线设置中使用"固定"的网络分布进行训练或优化. 故如果真实网络分布与训练集中的网络分布不同,这些基于学习的 ABR 算法表现将不佳.

同时,近年来学界提出了多种基于在线学习的 ABR 算法. OnRL[26] 采用联邦学习不断更新其实时通信策略. Puffer[27] 根据当前带宽容量定期动态更新配置图或 NN 模型. 此外,领域的另一个大方向为偏好感知的码率自适应算法. DAVS[28] 是一种基于模仿学习的方法,它考虑了用户的观看偏好,以使该方法适应 QoE 的多样性. Zuo 等人[29] 提出了用户偏好驱动的强化学习的 ABR 系统 Ruyi,将偏好感知融入到 QoE 模型和 ABR 算法中.

2.2 网络优化场景下的蒸馏框架

Meng 等人[17] 提出了 Metis 框架,可以对深度神经网络的决策进行解释,使得基于深度神经网络的网络系统具有可部署性. Metis 框架提供了两种不同的解释方法,并在两个领域的最先进的基于深度神经网络的网络系统上进行了评估,结果表明 Metis 提供了易于理解的解释,同时几乎不会影响性能. Primo[30] 是一个用于设计实用学习增强系统的框架,可以通过提供可解释的模型、使用贝叶斯优化和工具支持等方法,解决机器学习中的实质性缺陷,提高模型的可解释性和性能. 经过评估,Primo 能够提供清晰的模型解释、更好的系统性能和更低的部署成本.

2.3 深度强化学习在网络下的鲁棒性与可解释性

Zheng 等人[31] 提出了一种基于深度强化学习的带有可信老师的统一框架,称为 Teacher-Student 学习框架. 该框架通过引入领域专业算法作为老师,提供关键状态下的建议来改善网络的健壮性和可解释性. 学生神经网络在最大化期望奖励的同时,也会学习老师的建议. Alves Esteves 等人[32] 则针对软件化网络中使用机器学习进行管理的影响进行了评估,并提出了一种针对在线学习网络切片优化的鲁

棒性评估方法.研究假设切片请求到达是非稳态的，模拟了不可预测的网络负载变化，并比较了两种深度强化学习(DRL)算法：一种基于纯 DRL 算法和一种启发式控制的 DRL 算法，以评估这些不可预测的流量变化对算法性能的影响. Dethise 等人[33]提出了可以对具有非平凡特性的神经代理进行形式化分析方法，旨在提高计算机网络中神经网络的行为正确性和可预测性.实验结果表明，可以在几分钟内建立神经网络对其输入的对抗攻击的弹性，并在以前依靠有教养的猜测的属性上进行正式证明.最后，形式化验证有助于创建代理行为的准确可视化，提高其可信度.与此同时，有一些算法解释了基于学习的 ABR 算法. Dethise 等人[34]对基于机器学习的视频码率自适应模型 Pensieve 进行了一项案例研究，并发现该模型的决策可以基于很少的输入，并且很少使用某些可用的比特率.该研究强调了理解机器学习模型的重要性，超越它们的底线性能，并建议这样的异常可能存在于其他基于机器学习的解决方案中.

3　核心动机

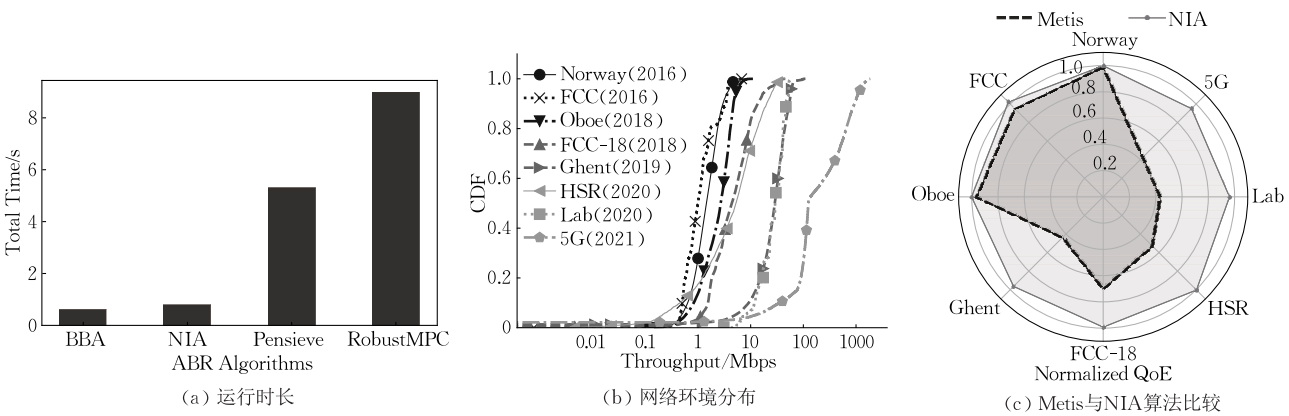
3.1　研究动机

基于强化学习的 ABR 算法在各种网络场景中都取得了优异的性能[11]，但是由于神经网络模型通常体积过大[14]以及整体运算消耗过多[35]，很难将模型直接部署在真实场景中.图 1(a)展示了各类不同的 ABR 算法在模拟器中运行测试环境的总时长，每一个 ABR 算法的详细介绍请参考第 5.1.5 节.可以看到，Pensieve 与 RobustMPC 的运行消耗都较大，与真正在线上部署并服务的 BBA 算法相比他们的整体开销多了 9～15 倍.于是，研究者[16-17]尝试将神经网络模型策略蒸馏到轻量级的决策树模型中，使 ABR 算法既能保持原有性能又能提升执行效率.图 1(a)中 NIA 算法即为决策树模型的 ABR 算法，其能够大幅度降低深度神经网络的运行开销到启发式算法的开销(BBA)，从而顺利部署上线.简要地说，模型蒸馏是将训练完成的神经网络模型看作教师模型 π^*，将待训练的决策树模型看作学生模型 π.学生模型的整体学习方式是一个监督学习，即在一个训练的网络数据集下通过教师模型标注“状态与动作”数据优化损失函数(通常被定义为平均预测准确率)：

$$\hat{\pi}=\arg\max_{\pi\in\Pi}(\mathbb{E}_{s\sim d_\pi}[\mathbb{1}_{\pi(s)=\pi^*(s)}]) \tag{1}$$

其中 s 为码率自适应过程中遇到的状态(state)，d_π 为使用学生策略 π 时的状态的平均分布，$\mathbb{E}$ 为策略 π 在所有元策略 Π 中的期望.在相同的状态 s 下，若学生策略选择的动作与教师策略执行的动作相同，则 $\mathbb{1}(\cdot)$ 等于 1，反之为 0.综上，本文的优化目标是最大化学生策略与教师策略相同的概率，从而完成行为克隆[36].

然而，过去的 ABR 蒸馏工作 Metis[17]忽略了决策树泛化性较差的问题，导致生成的决策树 ABR 算法只能在训练集的网络分布下工作，在训练集网络分布外的网络情况执行性能较差.为了验证这个结论，在 Metis 原论文提供的训练集上重新训练了决策树 ABR 算法，并且在 8 个现有的网络数据集上测试其性能.这 8 个网络数据集的吞吐量分布如图 1(c)所示，可以看到随着年数的增加，数据集的平均吞吐量明显提升.其中，FCC 数据集，Norway 数据集(即 HSDPA 数据集，下同)与 Oboe 数据集[23]具有相似的网络带宽分布，即属于低带宽网络环境；其余的网络带宽相对偏高，属于高带宽网络场景.本文将在第 5.1.3 节中展示部分数据集的详细统计信息.



(a) 运行时长　(b) 网络环境分布　(c) Metis与NIA算法比较

图 1　本图展示了本文的研究动机(与启发式算法相比，基于神经网络推理的 ABR 算法整体开销过大，故需要决策树蒸馏框架将策略迁移至轻量级网络结构中.然而，现实场景的网络环境分布多样，以往的蒸馏算法下的策略无法满足泛化性)

在这些网络场景下，现有的决策树蒸馏算法性能表现如何呢？图 1(b)汇报了 Metis 在以上八个数据集上的归一化 QoE 指标，其指标计算方法为 $\frac{QoE_{\text{metis}}}{QoE^*}$，其中 QoE^* 是教师模型在相同网络数据下得到的 QoE 分数. 如预料到的情况相似，Metis 仅仅在以上 3 个网络数据集中表现良好，反而在其余 5 个网络数据集上表现不佳. 这里值得注意的是，FCC、HSDPA 与 Oboe 这 3 个数据集都是低带宽网络环境，其余 5 个则为高带宽网络场景.

这个现象的本质原因在于决策树模型学到了训练数据集中与“总体网络”无关的特性，导致算法在带宽较好的数据集中依然选择较低码率的视频块(video chunk)，从而降低整体的 QoE 性能. 相反，如后文图 3 与图 4 所示，通过本文提出的蒸馏框架 NIA 生成的决策树策略能够较好地在蒸馏教师模型的原始策略，在多个网络场景下皆取得更好的泛化性能.

3.2 研究挑战

接下来，基于以上列出的研究动机，本文的核心方案旨在不使用真实场景网络环境训练学生模型，转而，通过 Data-free 的网络环境即人造生成的网络环境将算法与网络环境特性解耦，从而蒸馏出泛化能力更强的学生模型.

在该思想的引导下，本节进一步罗列了多个挑战. 并展示提出的训练框架对各个挑战的可行解决方案：

(1)生成多样的网络环境. 由于带宽瓶颈与竞争流等原因，真实世界的网络千变万化，导致无法在线下模拟出所有网络场景. 那么如何尽可能多地生成既贴近真实网络，又适合学生模型训练的网络环境？

解决方案：本文提出网络环境生成模块，在一定的取值范围内通过随机各种参数生成多个人工网络环境. 随后将这些网络环境合并在一起并存入网络环境池.

(2)选择适合训练的网络环境. 在建立好多个网络环境后，学生模型通过抽取网络环境池的环境蒸馏教师模型的策略. 因此，如何选择合适的网络环境供学生模型训练是另一个值得挑战的问题.

解决方案：本文进一步提出 Top-K 候选环境选择算法，每次结合探索-利用规则为学生模型推荐当前训练状态下较好的网络环境，在保证训练稳定的前提下提升学生模型性能和泛化性.

(3)高效训练学生模型. 最后一个挑战来自：在指定完训练的网络环境后，如何快速并高效地蒸馏教师模型的策略，训练学生模型？

解决方案：为了解决挑战，本文使用学生模型驱动，并模仿教师模型的策略. 简单来说，整体训练过程根据学生模型的策略与网络环境交互，教师模型跟随学生模型的轨迹为每个状态得出专家策略. 学生模型学习专家策略达成策略蒸馏.

综上所述，为了蒸馏出一个合适的学生模型，本文不仅需要构建丰富多样的网络场景，也需要提出合适的算法，在学生模型训练期间选择合适的网络场景从而更高效蒸馏教师策略，同时还需要提出一套更高效训练学生模型的方案.

4 NIA 设计

基于以上的研究动机与挑战，本文提出了 NIA，一个 Data-free 的码率自适应算法蒸馏框架. 本章详细阐述了 NIA 的系统结构，包括其每一个模块的设计以及总体训练步骤.

4.1 总体架构

如图 2 所示，NIA 的主要的执行过程分为预处理阶段和训练阶段. 在预处理阶段，NIA 利用环境网络生成器产生多个“人造”的网络环境，并将其放入网络环境池中. 在训练阶段，首先，每轮训练 NIA 通过网络环境选择器通过多臂赌博机算法从网络环境池中挑选合适的网络环境，并将其装载进虚拟播放器中. 学生模型将与虚拟播放器交互，采集当前的状态，并且根据状态选择合适的码率. 与此同时，状态也会输入教师模型并得到专家标签(Oracle). 最后，学生模型基于“状态，专家标签”数据对决策树网络训练，并将该网络环境下学生模型在专家标签下的平均选择概率回传给网络环境选择器，供多臂赌博机更新.

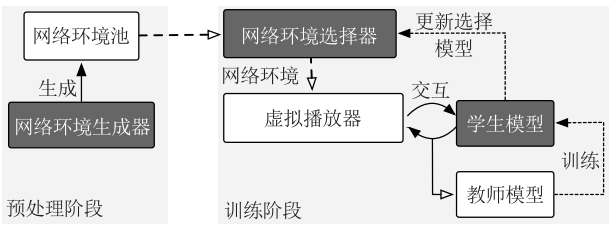


图 2 NIA 整体结构示意图(如图所示，NIA 主要包括环境生成模块、网络选择模块与学生驱动模仿学习算法)

4.2 网络环境生成模块

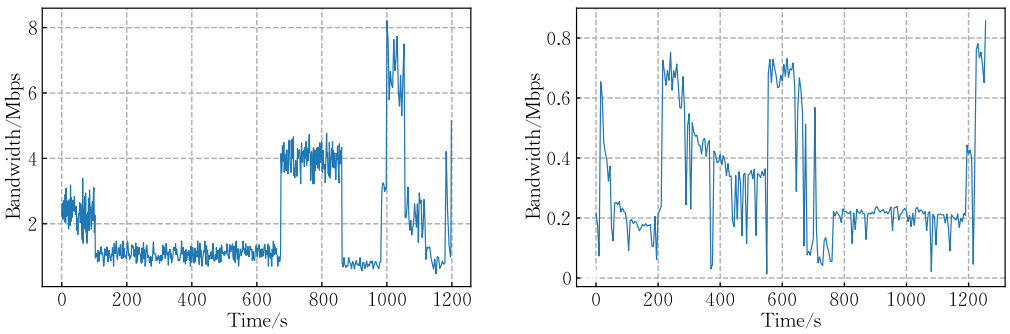
通过前面的实验可以得出，无论训练数据集包含的网络环境有多少“多样”，总会有一些特殊的网络环境并不包含在训练集内，导致原本泛化性就不高的决策树模型更加无法在这些网络环境中顺利执

行.因此,NIA 摒弃了之前的“构建训练集”的思想,转而使用“网络环境生成模块”构建人造的网络环境.

在预处理阶段,NIA 使用网络环境生成模块会随机产生 M 个人造的网络环境.在过去工作的指导下[37],每个网络环境将主要由以下参数生成:平均带宽、带宽标准差和持续时长.表 1 展示了每个参数的取值范围.每次生成模块根据表中每个参数的范围随机生成多条带宽轨迹(trace),并最终将这些轨迹拼接,输出为一个独立的网络环境.图 3 和图 4 展示了一个通过网络环境生成模块生成的网络环境与原始训练集中的网络环境,其中训练集来自 HSDPA 数据集[38].由结果可得,生成的网络环境保留了真实网络环境中的特性例如带宽具有隐马尔科夫迁移性[37],同时也发现网络环境具有自身的特性例如一定时间段内的随机波动性[27].值得注意的是,生成的网络环境个数 M 是一个重要的超参数.在第 6.2 节中会讨论生成的网络环境个数与决策树模块的性能之间的关系,在本文中 $M=1000$.

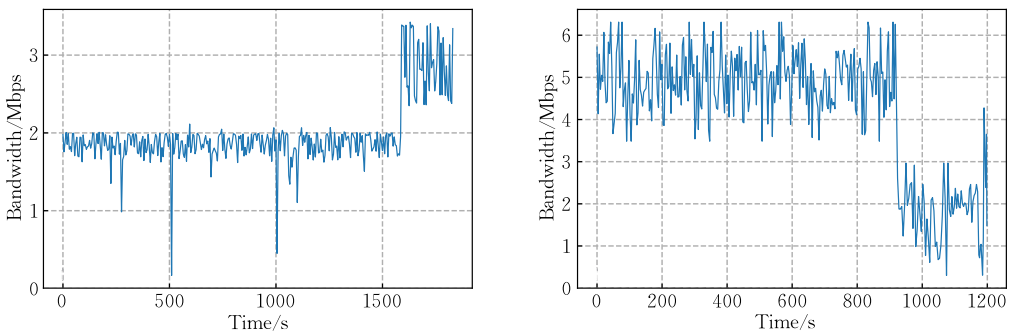
表 1　网络环境生成模块参数与范围

参数名	取值范围
平均带宽	0.1～7 Mbps
带宽标准差	0～1
带宽持续时长	1～5 s
生成带宽总长度	300～3000 s



(a) 数据集中网络环境:动态　　(b) 人工生成网络环境:动态

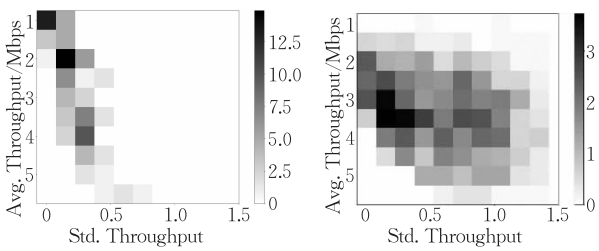
图 3　比较生成与真实数据集中的动态网络环境



(a) 数据集中网络环境:稳定　　(b) 人工生成网络环境:稳定

图 4　比较生成与真实数据集中的稳定网络环境

图 5 比较了 HSDPA 数据集与生成的网络环境分布并统计了每个数据集中的网络轨迹在某个带宽均值与标准差下的概率.可以看到,与真实网络数据集相比,使用环境生成模块生成的网络场景更具多样性,助力 NIA 生成更具泛化性的决策树算法.



(a) HSDPA　　(b) 本文生成

图 5　HSDPA数据集与NIA的环境生成模块的网络分布对比

4.3　环境选择模块

在预处理阶段生成完 M 个网络环境后,NIA 需要在训练时选择使用何种网络环境训练学生模型.为了解决以上的问题,NIA 使用了强化学习(Reinforcement Learning)技术以选择网络环境.由于以下几点原因,强化学习非常适合该问题.(1)首先,强化学习能够基于整体训练阶段的反馈动态做出决策.基于反复的“探索与利用”技术,强化学习算法能够逐渐地在没有人工先验知识的情况下找到最

优解;(2)其次,强化学习能够很好地在探索次优解与利用当前最佳方案之间找到平衡.因此,针对该问题考虑了三种不同的强化学习方法实现环境选择模块,包括多臂赌博机算法(Multi-armed Bandit)、深度强化学习算法(Deep Reinforcement Learning)以及贝叶斯优化算法(Bayes Optimization).深度强化学习算法考虑选择对环境的长期影响,但通常需要更多的训练数据和更复杂的模型.如果面对高维、连续决策问题,贝叶斯优化算法是一种强有力的解决方案.该算法在高维空间中具有有效性,能够有效地找到全局最优解.但是,由于其算法的计算复杂度较高,以及需要更复杂的模型表示潜在的目标函数,因此其计算成本和实现难度较大.

本文需要在有限的资源下,执行离散决策获得最大的收益.同时,需要算法能够适应快速变化的环境,快速做出最佳决策.故本文将网络环境选择问题建模为多臂赌博机问题(Multi-armed Bandit Problem)[18],即在每个决策时间点,当每个选择的属性在分配时仅部分已知并且随着决策步骤的推移可能会获取更多信息的情况下,以最大化其预期收益的方式在分配固定的有限资源集的问题.在实现上,NIA 的环境选择模块使用了 Top-K 下的 UCB 算法来解决了该问题.图 6 显示了整体逻辑框架,环境选择模块整体流程如下所示:

步骤 A. 收集历史学生模型性能信息.

步骤 B. 根据输入历史性能信息由低到高对网络环境进行排序.

步骤 C. 根据排序的网络环境列表选择 Top-K 个候选的网络环境.

步骤 D. 在 Top-K 个候选人的基础上使用 UCB 算法在线探索并利用(E2)当前最有效的网络环境.

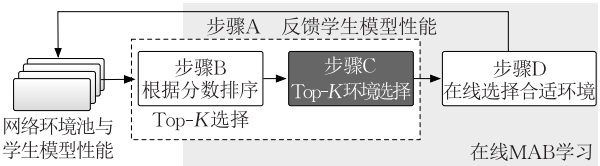


图 6 NIA 的环境选择模块整体流程图

因步骤 A 与步骤 B 已经足够清晰,故本文不会对其做太多解释.接下来详细解释步骤 C 与步骤 D.

4.3.1 Top-K 候选环境选择

尽管在预处理阶段生成了 M 个网络环境,但是对当前训练效率最高的网络环境通常是在学生模型性能相对较差的前 k 个网络环境中,其中 k 相对于 m 来说是一个很小的比例.例如,在带宽充足的情况下(即大于视频的最高码率档位),学生模型仅需简单地学习到选择最高码率档位即可;相反,在带宽不足的场景下,学习模型仅需并无法进一步提升学生模型的性能.因此,NIA 在每次选择前将会基于学生模型在该网络环境下的性能评分从小到大排序,并且选择前 k(Top-k)个网络环境作为环境候选.注意,在此过程中,每个网络环境的性能评分是基于长时历史信息构建的.本文将在第 6.2 节中讨论不同的 k 的取值对 NIA 性能的影响,并最终选择 $k=20\%$.

4.3.2 在线选择合适环境

在线探索前 k 个网络环境与学生模型的潜在性能问题可以被抽象为多臂赌博机(MAB)问题.在这个问题中,每个网络模型对应了该问题中的“臂”(arm),学生模型与教师模型的选择之间的差距对应了该问题中的“奖励”(reward).在每次决策的时候,NIA 根据选择的 k 个网络环境下学生模型表现出的性能选择出当前状态下最好的网络环境.由于学生模型随着训练步数的增加策略逐渐变化,本文使用了衰减 UCB(Discounted-UCB)算法应对这个挑战[39].一方面,它记录了过去一段时间内学生模型的整体表现情况,使结果更加鲁棒.另一方面,它进一步鼓励算法探测较多轮次没有被选择到的网络环境.式(2)(UCB)具象化了在 t 步下的整体选择过程,由两部分组成:

$$UCB_t=R_t(i)+P_t(i) \tag{2}$$

第一部分 R_t 为过去一定步数下的累计分数,r_t 则表示学生模型在网络环境 i 下的状态集合 S_i 的性能.R_t 计算公式如下所示:

$$R_t(i)=\sum_{k=0}^{t}\gamma^{t-k}\mathbf{1}_{c_k=t}r_t(i) \tag{3}$$

其中,$\gamma\in[0,1)$ 是一个折扣因子以控制过去的学生模型性能对现在的选择造成的影响.γ 较小则只看过去较少步下的学生模型性能.相反,γ 较大则会考虑较长步下的模型总体性能.本文参考强化学习中的经验公式[40],根据考虑的步数 τ 设置最有效的 γ:

$$\gamma=\frac{\tau-1}{\tau} \tag{4}$$

在 NIA 中,τ 被设定为 10,则 γ 被设定为 0.9.在实践中,在每一步 t 下 NIA 利用最新的学生模型性能迭代更新累计分数 R_t,即

$$R_t(i)=R_{t-1}(i)+\gamma r_t(i) \tag{5}$$

其中$\mathbf{1}$为二值指示函数(binary indicator function),当过去的第 k 步选择了当前的网络环境(即$\sqrt{c_k=t}$时),该函数返回 1,否则返回 0.

$$r_t(i)=E_{s\in S_i}[\pi(\arg\max_a \pi^*(a|s)|s)] \quad (6)$$

式(6)列出了 r_t，其意义为学生模型在当前网络场景下，在教师模型输出的最优动作下执行的动作概率的期望. 详细地说，$\arg\max\pi^*(a|s)$ 表示教师模型在状态 s 下选择概率最大的动作 a；$\pi(a|s)$ 代表学生模型在状态 s 下选择动作 a 的概率；$E_{s\in S_i}(\cdot)$ 为学生模型概率在当前网络环境的状态集合 S_i 下采集到的所有状态 s 的期望，即学生模型在该网络环境下的性能.

第二部分的 P_t 为 UCB 中鼓励探索的填充函数(padding function)[39]，主要刻画了当前性能估算的不确定性. 对于每一个网络环境 i，NIA 的填充函数如式(7)所示，其中 B 是一个超参数，调整了 NIA 在选择之前未选择的网络环境的激进程度.

$$P_t(i)=B\sqrt{\frac{\log t}{\sum_t \mathbb{1}_{i=c_t}}} \quad (7)$$

综上，得出了每一步 t 下的选择 c_t 的公式，如式(8)所示. 值得注意的是，与过去 UCB 方案选择最好分数对应的环境不同，本文的目标是选择当前学生模型表现得较差的网络环境，故这里的超参数 B 最好设置为一个小于 0 的数. 第 6.2 节中讨论了 B 的选取对于 NIA 的最终性能的影响. 在本文中，$B=-0.2$.

$$c_t=\arg\min_i \frac{\sum_{k=0}^{t}\gamma^{t-k}\mathbb{1}_{c_k=t}r_t(i)}{\sum_{k=0}^{t}\gamma^{t-k}\mathbb{1}_{c_k=t}}+B\sqrt{\frac{\log t}{\sum_t \mathbb{1}_{i=c_t}}} \quad (8)$$

$$\text{s.t.},\ B<0.$$

4.4　学生驱动的模仿学习模块

本部分讨论如何更高效地蒸馏训练学生模型. 过去的工作[17,41]都采用教师驱动的方式与环境交互，即在训练阶段以教师模型的策略实时生成数据集(状态-动作对)，学生模型通过学习生成的数据集完成策略蒸馏. 然而，该方案采集到的训练数据源自教师模型策略而非学生模型策略，故可能在学生模型自行执行 ABR 决策时探索到不在数据集中的状态，选择不佳的码率视频块后，进一步探索到另一个不熟悉的状态，从而造成“累计误差问题”(compounding error)[36].

因此，在模仿学习算法[36]的启发下，NIA 考虑利用当前的学生模型与网络环境交互，通过学生的探索显著降低累积误差问题[12]. 整个过程由学生模型，教师模型与经验回放池 B 组成. 接下来将详细介绍学生模型的输入输出以及训练过程.

学生模型输入. 学生模型为一颗决策树模型，其状态输入 s_t 主要由三部分组成：

$$s_t=\{Q_{t-1},B_t,C_t,D_t\} \quad (9)$$

其中，Q_{t-1} 是上一个时刻 $t-1$ 的码率选择；B_t 为当前缓冲区大小；C_t 则为一个过去 5 个时刻下的带宽序列：$C_t=\{c_{t-4},\cdots,c_t\}$；$D_t$ 为过去 5 个时刻下的下载时长序列：$D_t=\{d_{t-4},\cdots,d_t\}$. 故学生模型的输入为 12 维向量. 第 6.1 节中讨论了输入的过去吞吐量的数量对整体性能的影响.

学生模型输出. 与教师模型的输出的个数保持一致，即视频的码率档位数量. 同时模型将输出选择每一个码率档位的概率：$\pi(a|s)\in(0,1)$，并且 $\sum_a \pi(a|s)=1$.

学生模型训练过程. 在第 t 步，NIA 采用学生模型根据当前观测到的状态 s_t 推断出当前策略 π_t，并根据 π_t 在每个选择上的概率选择当前动作 a_t. 为了提升整体效率，采样部分使用了 Gumbel-Softmax 采样算法[42]，如式(10)所示：

$$a_t=\arg\max_a(-\log\log\mu+\log\pi(a|s_t)) \quad (10)$$

其中 μ 是一个在一定范围内采样的随机噪声：$\mu\sim(0,1)$. 该公式的前半部分为 Gumbel(0,1)分布，后半部分为学生模型的输出概率.

随后 NIA 进一步执行教师模型策略，根据 s_t 推断出当前的最优动作 $a_t^*=\arg\max_a\pi^*(s_t,a)$. 而后将状态动作对 $\{s_t,a_t^*\}$ 加入经验回放池中. 之后，运行学生模型的动作 a_t 并观察下一个状态 s_{t+1}，并将该状态动作对 $\{s_{t+1},a_{t+1}^*\}$ 加入经验池. 每隔一段训练步长，NIA 读取经验回放池中的所有状态动作对并根据式(6)训练学生模型. 与过去的工作一样[16]，NIA 也采用 CART 算法作为决策树模型训练的核心算法，因为该方案能提供可靠的稳定性.

结合以上所有模块，NIA 的训练算法如过程 1 所示. 值得注意的是，NIA 可以使用多线程并行采集样本并完成学生模型的蒸馏训练.

过程 1.　NIA 执行过程.

输入：教师模型 π^*，网络环境生成模块，环境选择模块

输出：学生模型 π

初始化学生模型 π

初始化经验回放池 $B=\{\}$

$D\leftarrow$ 网络环境生成模块(M)

trace$\leftarrow$ 环境选择模块(D)

$t \leftarrow 0; k \leftarrow 0$
获取当前 ABR 状态 s_k
REPEAT
根据 $\pi(s_k)$选择当前动作 a_k
获取专家策略 $a_k^* \leftarrow \pi^*(s_k)$
$B \leftarrow B \bigcup \{s_k, a_k^*\}$
基于 B 与式(6)生成学生模型 π
根据 s_k执行动作 a_k并获得 s_{k+1}
IF 已经完成视频播放 THEN
trace←环境选择模块(D)
根据式(3)计算学生模型性能 r_t并更新网络环境表
$t \leftarrow t+1; k \leftarrow 0$
获取当前 ABR 状态 s_k
ENDIF
$k \leftarrow k+1$
UNTIL 运行收敛
返回 π

5 实验与分析

5.1 实验设置

5.1.1 虚拟播放器

首先,本文设计并实现了一个准确的 ABR 离线虚拟播放器,它可以通过网络环境和码率自适应的视频信息准确模拟码率自适应播放过程. 该虚拟播放器的逻辑参考了多个开源 ABR 模拟器,使用 Python3.6 编写[11,43].

5.1.2 视频生成器与测试集

同时,本文实现了视频生成器,使之能在 NIA 的训练期间观测到码率档位下不同大小的视频. 在测试阶段,利用 EnvivioDash3 对各个算法进行测试. EnvivioDash3[44]是 DASH.js[45]参考视频,它主要有 6 个码率档位,其档位分别是{0.3,0.75,1.2,1.85,2.85,4.3} Mbps. 该视频一共由 49 个长度为 4 s 的视频块组成.

5.1.3 网络环境测试集

之前提到,在训练阶段利用网络环境生成模块生成的网络环境蒸馏学生模型. 在测试阶段则会使用公共数据集作为网络环境测试,其中公共测试集分类为两大类,分别是:(1)低带宽网络场景(≤6 Mbps),包括 HSDPA[38]、FCC[46]和 Oboe 数据集[23];(2)高带宽数据集(10～300 Mbps),包括 FCC18[41]、HSR[16]和 5G 数据集[47]. 表 2 列出了每个公共数据集的网络平均带宽和网络标准差.

表 2 网络环境生成模块参数与范围

网络数据集	网络平均带宽/Mbps	网络标准差
HSDPA[38]	1.24	0.80
FCC[46]	1.31	1.00
Oboe[23]	2.64	1.50
FCC18[41]	6.51	6.61
HSR[16]	7.64	7.96
5G[47]	347.46	378.16

5.1.4 QoE 指标

本文使用 QoE_{lin}[10,11,23,27]作为 QoE 的评判指标,这是一个由多个指标线性加权组合的函数. QoE_{lin}的表达式如式(11)所示.

$$QoE_{\text{lin}} = \sum_{n=1}^{N} Q_n - \max Q \sum_{n=1}^{N} T_n - \sum_{n=1}^{N-1} |Q_{n+1} - Q_n| \quad (11)$$

其中,N 表示视频的总数,Q_n表示视频块 n 的码率,T_n是卡顿时长,$\max Q$ 表示码率阶梯的最大码率,在本文中该参数为 4.3. 故 QoE_{lin}的主要由整个视频的平均视频码率、平均卡顿时长和视频前后码率切换三个指标组成.

5.1.5 对比算法

本工作从各种类型的 ABR 算法中选择了几个代表性的方案,包括启发式 ABR 算法和基于学习的智能 ABR 算法.

Buffer-Based Approach(BBA)[8]. 基于普通缓冲区的 ABR 方法. BBA 基于当前缓冲大小结合两个参数决定和最小最大码率之间的对应关系. 本文中使用了 BBA 自带的参数,即在缓冲大小为 5 s 下选择最小码率,在缓冲大小大于 15 s 时选择最大码率,其余根据当前缓冲大小通过线性估计决定对应码率的视频块.

RobustMPC(RMPC)[10]. 是基于预测控制模型(MPC)的 ABR 算法. 详细地说,它使用过去的测量带宽预测未来带宽,并在虚拟播放器中连续推演多步,最后得出性能最高的轨迹对应的视频码率. 本文采用 RMPC 的默认参数,使用调和平均预测未来带宽,并对未来连续推演 5 步.

Metis[17]. 该方案使用决策树作为学生模型,在训练集提供的网络场景下蒸馏教师模型策略. 注意,Metis 是一种基于"教师驱动"探索的决策树蒸馏方案.

Pensieve[11]. 一个基于深度强化学习的 ABR 算法,在给定的网络环境下自主学习提升总体奖励分数. 本文使用作者提供的预训练模型,并将其作为教师模型供学生模型蒸馏.

5.1.6 NIA 参数设置

NIA 的详细参数设置如下:网络环境方面,网

络环境生成个数 $M=1000$，Top-K 比例为 20%，UCB 算法的探索参数 $B=-0.2$；学生模型方面，采用过去 5 个吞吐量作为带宽输入，深度为 9；训练方面，模仿学习经验回收池大小为 100 000，多线程并行数为 16. 本文采用 Pensieve 的预训练模型为教师模型.

5.2　NIA 性能分析

本文中实验主要包括各个 ABR 算法在低带宽网络环境和高带宽网络环境的表现，其中低带宽网络环境包括 HSDPA、FCC 和 Oboe，高带宽网络环境包括 FCC18、HSR 和 5G. 每个算法都会在各个网络环境的每一个带宽轨迹(network trace)中运行，并汇总为 CDF 曲线图.

低带宽网络. 如图 7 所示，在低带宽网络环境下，基本所有的算法几乎都表现出了较好的结果. 例如图 7(a)在 HSDPA 网络场景下，启发式算法 RobustMPC 和基于决策树的机器学习算法 NIA 都取得了较好的 QoE，然而在大部分网络轨迹下 NIA 的表现都优于 RobustMPC. 经过详细统计，NIA 在 QoE 指标上较 RobustMPC 高出 6.81%，较 BBA 算法高出 48.97%. 特别是较同样是决策树的机器学习 Metis 算法高出 1.39%. Metis 表现良好的潜在原因是其使用的训练集更加贴合 HSDPA 网络环境. 同理，RobustMPC 的原始参数也是基于该数据集的网络情况配置的. 同样，BBA 的参数由大规模 A/B 实验得出[48]，并非该数据集的最优解. 与这些算法不同的是，NIA 的所有训练网络环境都是“人造的”，即其策略并不依赖某个数据集的分布，故获得更鲁棒的结果. 相似的结论可以在 FCC 数据集的结果中得到. 如图 7(b)所示，NIA 的 QoE 性能较 BBA 提升 45.95%，与 RobustMPC 相比提升 1.32%，较 Metis 提升 8.41%——由于该网络场景与训练场景差距较大，故 Metis 与 NIA 的性能差距也变大了.

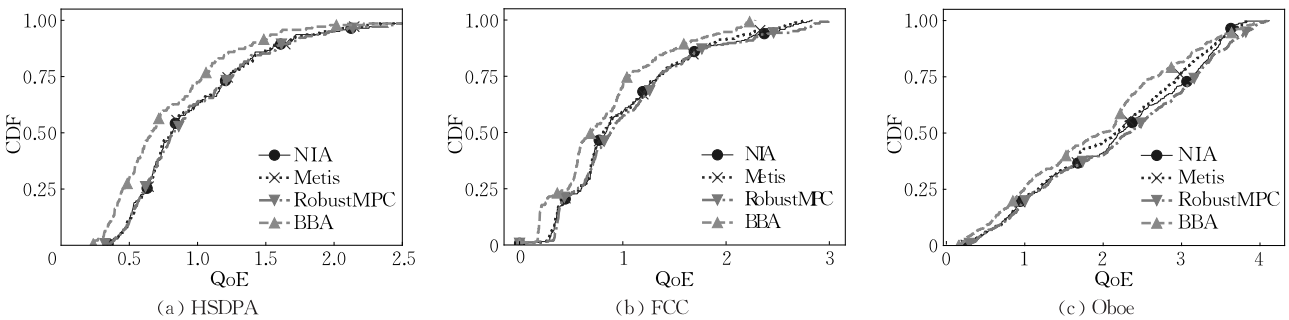


图 7　在低带宽场景(HSDPA，FCC，Oboe)下比较了 NIA 与各种 ABR 算法的性能，并绘制了 QoE 的 CDF 曲线，其中越靠右边表示该算法性能越佳

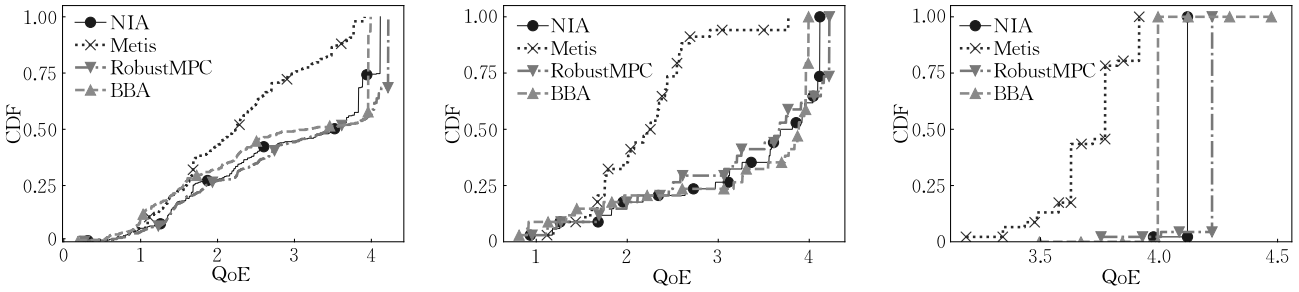
最后，在 Oboe 网络环境中(图 7(c))，RobustMPC 取得了最高的 QoE 评分(2.193)，比 NIA 的 QoE 高出 3.3%，同时较 Metis 算法高出 7.03%. 取得该结果的本质原因在于 NIA 和 Metis 的策略来自于蒸馏基于学习的 ABR 算法 Pensieve[11] 的策略，然而 Pensieve 在该网络场景下的整体性能较 RobustMPC 低 3.39%，最终由于 Pensieve 性能不足导致 NIA 在网络场景下取得一般的性能结果. 本文将在第 5.3 节讨论较 NIA 与 Metis 在各个网络场景中相较于 Pensieve 的收敛情况.

高带宽网络. 图 8 展示了各个 ABR 算法在高带宽网络场景下的表现性能，算法包括 NIA、Metis、BBA 和 RobustMPC. 基于结果，总结了两点信息.

(1) NIA 在高带宽网络场景下都取得了较好的结果，这代表着人工网络环境生成器的有效性——既有低带宽网络下复杂多变的场景，又生成了符合高带宽网络下“平稳”但“突变”的性质的网络环境. 例如，在 FCC18 场景中，NIA 的总体性能较 BBA 算法提升 5.8%；在 HSR 网络场景中，NIA 的 QoE 性能指标较 RobustMPC 算法提升 5.05%；在 5G 网络场景中，由于大部分情况下网络情况都远高于该视频的码率最大值(4300 kbps)，故 NIA，BBA 与 RobustMPC 几乎都得到了几乎满分的 QoE 性能.

(2) 尽管 Metis 在低带宽场景下的性能表现较好，在高带宽网络下性能则由于策略泛化问题导致较大下降. 如图 8(a)所示，与 RobustMPC 相比，Metis 的整体 QoE 性能降低 46.86%. 从 CDF 图上看，大部分性能损失来自较中高带宽网络下的保守码率选择策略. 例如在 QoE 性能大于 2.0，小于 3.5 的网络场景下，Metis 偏向于一直选择中等码率档位 1850 kbps. 即使在缓冲区增加后，Metis 依然不选择 4300 kbps 的高码率档位，最终导致其 QoE 性能

不佳.图 8(b)的实验结果也验证了类似的结论:中高带宽网络下,Metis 同样没有“蒸馏”来自教师模型的专家策略,反而保守地选择较低码率的档位并且不切换到最高码率档位.反观 NIA,其策略能够较好的在该网络下适应带宽变化,选择较好的码率.在 HSR 网络场景下,NIA 的整体性能较 Metis 高近一倍,即 99.42%.5G 网络的实验结果由图 8(c)所示.在高速带宽(347 Mbps)网络环境下,NIA 几乎总能够选择最高码率档位,从而得到较好的 QoE 性能.相反 Metis 虽然也能够选择高码率档位,但是其保守的策略只允许它在缓冲时间大于 40 s 的情况下切换到最大码率,浪费了大量带宽,降低了整体 QoE.



(a) FCC18 (b) HSR (c) 5G

图 8 在高带宽场景(FCC18,HSR,5G)下比较了 NIA 与各种 ABR 算法的性能,并绘制了 QoE 的 CDF 曲线,其中越靠右边表示该算法性能越佳

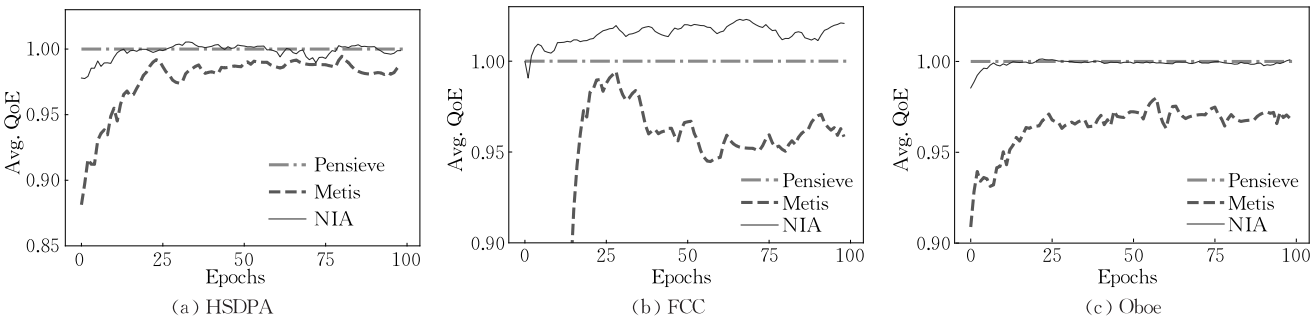
5.3 NIA 蒸馏性能

本实验测试了通过 NIA 系统,学生模型是否成功学习到了教师模型的策略.跟之前的实验一样,本实验也包括两个网络场景,分别是低带宽网络和高带宽网络.图 9 汇总了低带宽场景下 NIA 与 Metis 在训练时与教师模型(即 Pensieve)的相对归一化 QoE 性能.

NIA 与 Pensieve 的比较. 通过实验结果可以看到 NIA 较好地“复制”了教师模型的策略,在三个网络场景下都接近甚至超过了教师模型策略.例如,在 HSDPA 网络中,NIA 的 QoE 性能较 Pensieve 提升 0.91%,即不分上下;而在 FCC 网络场景中,NIA 与 Pensieve 相比在 QoE 上进一步提升了 2.41%.该现象很容易解释,因为在模型蒸馏过程中,NIA 算法(学生模型)是能够通过生成的网络环境分析出教师模型中不擅长策略的,并且利用教师模型在其他网络环境下的优质策略替换了这些次优策略.

NIA 与 Metis 的比较. NIA 在整体性能与训练效率上均优于 Metis.一方面,在低带宽场景下,NIA 的整体 QoE 性能较 Metis 提升 2.56%~7.53%.另一方面,NIA 的学生模型策略在运行 10 步后即进入平缓的收敛期,而 Metis 算法往往需要 25 步甚至更多数据才能进入收敛期.

此外,当 Metis 使用了环境选择模块与 Top-K 环境选择算法后,其本身的最终收敛后的性能也提升了.表 3 展示了各个算法在所有数据集下的 QoE 性能,其中 Metis-NIA 代表着使用 NIA 框架的 Metis 算法.可以看到使用人工数据集之后的 Metis 成功避免了泛化能力较差的问题,在多个数据集中均取得了较好的结果,但是与 NIA 相比,使用教师驱动的 teacher-student 方法的收敛速度不如学生驱动的模仿学习方法.值得注意的是在 FCC18 数据集上,所有的基于学习的 ABR 算法表现都不如启发式算法 RobustMPC.



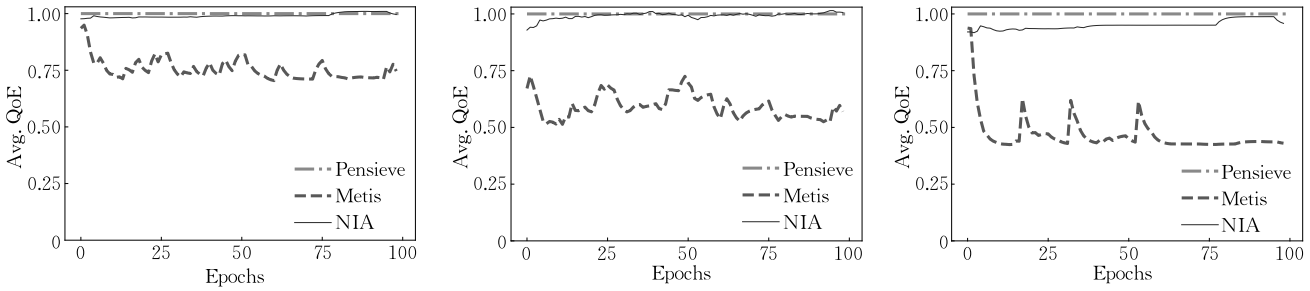
(a) HSDPA (b) FCC (c) Oboe

图 9 训练中 NIA 与 Metis 在低带宽数据集上的表现,结果被统计为归一化 QoE

表 3　将 Metis 放入 NIA 的训练框架后性能对比

ABR 算法/场景	HSDPA	FCC	Oboe	FCC18	HSR	5G
Pensieve	0.92	0.85	2.09	2.87	3.04	4.16
RobustMPC	0.89	0.86	2.19	3.04	2.97	4.21
Metis	0.94	0.81	2.04	2.2	1.85	3.69
Metis-NIA	0.94	0.85	2.12	2.85	3.04	3.96
NIA	**0.95**	**0.87**	**2.12**	**2.91**	**3.07**	**4.12**

NIA 在高带宽场景下的性能. 在高带宽网络下(如图 10),使用 NIA 训练的学生模型也展现出了泛化性能. 在三个场景下,NIA 的 QoE 性能都接近了教师模型的策略性能,并且一直性能稳定直到收敛. 反观 Metis 在训练初期(0～3 步)得到了最佳性能,但是其随着训练过程(3～100 步)整体性能不断下降,最后收敛到次优解. 其根本原因在于 Metis 的基于有限的真实网络场景作为训练集蒸馏教师模型策略,其策略学习到了数据集中的带宽信息而非宏观的策略,造成其总体选择较为保守. 特别是在 HSR 和 5G 这些带宽非常高的网络环境下,Metis 蒸馏的学生模型相较于教师模型在QoE性能上下降了50%.



(a) FCC18　(b) HSR　(c) 5G

图 10　训练中 NIA 与 Metis 在高带宽数据集上的表现(结果被统计为归一化 QoE)

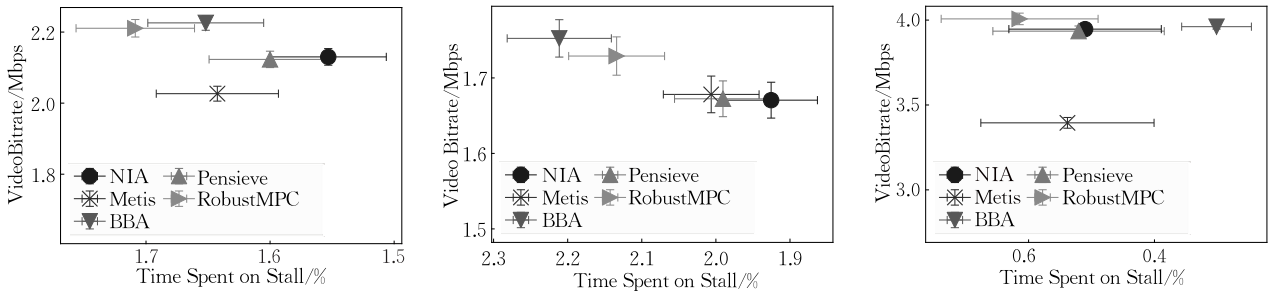
5.4　ABR 算法的深入研究

为了更加深入地了解各个 ABR 算法在不同网络场景下的性能,本部分尝试设计了实验以研究每个算法在各个场景下的码率与卡顿率的关系. 具体来说,首先将网络场景根据网络带宽大小将其分为低带宽场景和高带宽场景,随后在这些场景下分别对每一个 ABR 算法测试,并记录其在每一个网络轨迹下的码率与卡顿率. 最后,将信息汇总并绘制到图,其中误差棒表示每一个算法的 95% 置信区间. 结果如图 11 所示,由实验结果可以得出以下结论:

首先,通过算法在所有场景中的表现,可以发现 NIA 获得了最小的平均卡顿率. 在平均码率指标上,NIA 也取得了 Top-3 的性能表现. 特别的是,与其教师模型 Pensieve 相比,NIA 在码率和卡顿率两个指标上均占优,展现出了 Data-free 蒸馏框架的优势——通过生成并学习更丰富的网络场景得到泛化能力更好的 ABR 算法. 此外,Metis 虽然也蒸馏自教师模型,但是其两项指标均低于教师策略,特别是码率相对下降了 200 kbps.

其次,在低带宽场景下 NIA 和 Metis 都获得了最小的平均卡顿率,并且整体表现与教师模型几乎相同. 与这些算法相比,以往的启发式方案虽然能够取得较好的平均码率(与 NIA 相比提升了 1.79%),但是它们的平均卡顿时长相较 NIA 却提升了 18.42%. 与此同时,从误差图来看,NIA 的整体表现较其他方案更加稳定.

最后,高带宽场景的实验结果表明了 Metis 在所有场景上表现不佳的根本原因:在各个算法几乎都没有卡顿时长的情况下,Metis 的平均码率仅 3.4 Mbps,与其他方案的 4.0 Mbps 相差较大. 在富足的带宽下较保守的码率选择策略也是之前实验中展示的 Metis 在高带宽下获得较低 QoE 的理由之一. 反观 NIA 的学生模型策略,虽然其没有在训练阶段接触过真实的网络环境数据,但是其性能与教



(a) 所有场景　(c) 高带宽场景　(b) 低带宽场景

图 11　在各种网络场景下各个 ABR 算法的平均视频码率与平均卡顿率(其中误差棒为 95%置信区间)

师模型相比依然不错，甚至在各个维度下均小幅度优于教师模型，故取得了较优的 QoE 结果. 此外，BBA 算法在高带宽场景中即取得了最高的视频码率又取得了最低的卡顿时长，表明其默认参数比较适合普遍的高带宽场景. 但是相同的参数在低带宽下表现不佳，导致算法在所有场景下取得较高的卡顿时长. 该结果进一步验证了启发式算法需要在各种网络场景下精心调整其参数，包括表格查询或者神经网络辅助调整[14]. 而 NIA 从零开始学习网络的变化，在不同网络环境下蒸馏教师模型的策略，达到一个策略适应多个场景(one-fits-all)的目的.

6 算法讨论

本章讨论 NIA 不同模块中不同设置对整体性能的影响，包括学生模型的参数、网络环境生成器生成个数、Top-K、网络环境选择算法以及各种类型的教师模型.

6.1 学生模型参数

本部分首先讨论学生模型中的输入参数与决策树深度与整体性能的联系.

过去带宽个数. 在学生模型的输入中，过去测量的吞吐量的数量占了特征总量的 50% 以上. 本实验研究了过去吞吐量数量对策略性能的影响. 具体而言，本节从零开始分别训练了多个学生模型，其中每个模型的输入中对过去测量的吞吐量序列的数量不同，包括过去 0 个到 8 个吞吐量. 图 12 显示了过去吞吐量序列与 NIA 性能的关系，其圆点为策略在 8 个网络测试集上的均值，误差棒为标准差. 通过结果可以得到“过去吞吐量”对 NIA 性能提升相对较大. 与不同该指标做决策相比，只用过去最近的一个吞吐量作为输入即可在 QoE 上提升 7.8%. 同时可以看到，在输入 3 个过去吞吐量后，学生模型的整体性能趋于稳定，并且将过去 5 个吞吐量作为输入可以得到最优性能. 于是，NIA 采用了过去 5 个吞吐量作为带宽信息输入.

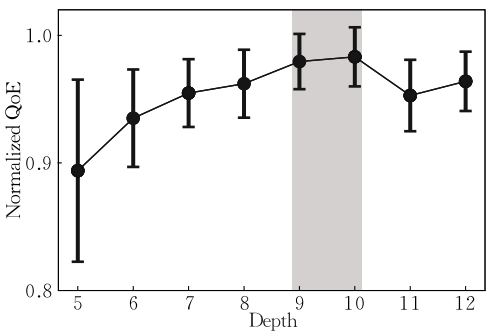


图 12 过去带宽输入数

决策树深度. 决策树的深度影响了模型的性能与泛化能力，但是过深的决策树有可能过拟合数据集，造成性能下降[49]. 本实验的详细设置如下：分别将学生模型的决策树深度设为 5 到 12. 对于每一个设置，决策树将由 NIA 从零训练至收敛，并在本文使用的八个网络环境测试集上依次测试，并使用均值与标准差描述每一个设置下学生模型的性能.

模型深度的实验结果如图 13 所示，其中圆点表示了每一个深度下在网络环境测试集上的平均归一化 QoE 性能，误差棒则表示它在不同测试集上的标准差. 随着决策树网络的深度增加，NIA 的学生模型性能大致呈先缓慢上升后迅速下降最后平稳的趋势. 其中，当深度在 9 到 10 时，NIA 的学生模型性能最优，且整体性能更稳定(标准差较小). 考虑到深度为 10 时其决策树的参数量为深度为 10 的模型的 2 倍，本文采用深度为 9 的决策树作为学生模型.

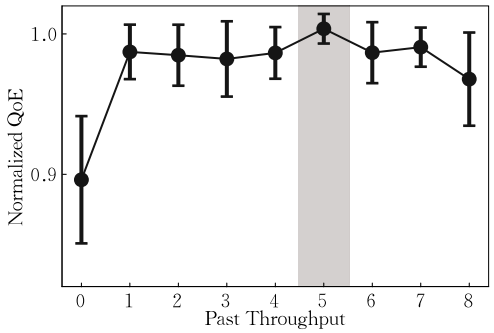


图 13 决策树深度与性能

此外，本节比较了决策树的模型大小与执行开销的比较. 实验结果表明，随着决策树深度的增加，模型大小从 4.2 KB(深度为 2)逐渐增加到 215 KB(深度为 9)，然而它们的整体执行时长并未有明显增加(整体波动小于 70 ms). 这个现象是合理的，因为与深度为 2 的决策树相比，深度为 9 的决策树仅仅多了 7 个判断，并不会增加整体执行时长.

最后，本节讨论了决策树深度对于训练时长的影响. 假设训练的收敛轮数为 5000 步，那么决策树的训练时长为 13 min(深度为 2)到 25 min(深度为 9). 于半小时内的蒸馏时长都是可接受的，因为在同样的配置下 Metis 通常需要训练 2 h 才能得到较好的性能.

6.2 网络环境参数

最后，本文研究网络环境生成器与选择器的各个环节的参数选取. 表 4 列出了所有可以调整的参数与 NIA 平均归一化性能的关系. 以下将详细介绍每个参数.

表 4　网络环境参数表

环境生成 M	NIA 性能	Top-K/%	NIA 性能	UCB 参数 B	NIA 性能	环境选择算法	NIA 性能
100	0.95±0.02	5	0.97±0.05	−0.1	0.95±0.05	随机	0.94±0.03
500	0.96±0.03	10	0.97±0.05	−0.2	**0.98±0.02**	Top-K	0.97±0.03
1000	**0.98±0.02**	20	**0.98±0.02**	−0.5	0.97±0.03	UCB	0.97±0.03
2000	0.92±0.08	40	0.89±0.11	−1.0	0.97±0.03	Top-K+UCB	**0.98±0.02**

网络环境生成数量. 如前文所示，该数量是 NIA 重要的超参数之一. 一方面，如果生成的网络环境个数较多，则可以增加虚拟网络场景的多样性. 但是过多场景可能包含重复特性的网络环境，从而增加环境选择的难度. 另一方面，如果生成的网络环境个数较少，则网络环境池缺少足够的代表性的网络场景，导致模型性能较差. 如表 4 所示，随着环境生成数量的提升，NIA 性能在数量 M 为 1000 时取得最佳性能. 在生成数量达到 2000 后，NIA 的整体性能下降，符合预期. 故本文选择的网络生成数量为 $M=1000$.

Top-K 比例. 本实验进一步比较了网络选择器中的 Top-K 比例下的候选数量. Top-K 参数包括前 5%、10%、20%和 40%. 实验结果表示当候选数量为整个环境池的前 20%时，NIA 的性能取得最大. Top-K 参数取值为 5%与 10%对 NIA 最终性能的影响不大，但是当取值为 40%时，性能出现较大下降，归一化性能从 0.98 下降到 0.89. 综上所述，本文选择的 Top-K 比例为 20%. 值得一提的是，当 Top-K 的比例为 1%时，NIA 的性能将退化到 0.9，显著地低于 5%下的性能. 这里可以认为 NIA 的环境选择器不停地挑选相对于决策树来说的边角案例(Corner Case)，无法对整体性能提升带来帮助. 相反，使用 Top-K 算法能够避免频繁选择部分网络场景，陷入局部最优的情况.

环境选择器算法. 随后，本部分确认了环境选择模块每个步骤的重要性. 该实验包括：(1) 随机选择网络，即每次在环境网络池中随机挑选一个网络场景训练；(2) 在 Top-K 范围内随机选择网络，即在前 K%比例的网络环境中随机选择一个供学生模型训练；(3) 使用 UCB 算法直接在网络环境池中挑选合适的网络场景；(4) 同时使用 Top-K 和 UCB 算法，在 Top-K 的方位内利用 UCB 算法选择网络环境训练. 从实验结果可以得出随机算法在各个网络数据集的性能最低，为 0.94，而 Top-K 与 UCB 两者性能几乎相同. 最优的搭配是同时使用 Top-K 与 UCB 算法，因为它们即可以稳定训练过程(Top-K)又可以避免局部最优解(UCB)[50].

UCB 算法探索指数 B. 最后，本节探索了式(2)中的超参数 B 的取值. B 值越大则 NIA 的“探索权重”越大，越不容易陷入局部最优，但是不容易收敛. 相反，B 的取值越小则 NIA 更侧重“利用”当前已经探索的经验，整体训练稳定但容易陷入局部最优. 实验结果表明当 $B=-0.2$ 时 NIA 取得性能与效率的平衡.

7　NIA 局限性

虽然 NIA 能够取得较好的结果，但它仍然存在一些不足：首先，NIA 使用了人工生成的网络环境蒸馏决策树策略. 这意味着策略的泛化能力仍受限于人工网络环境的覆盖范围——如果人工网络环境无法充分代表真实世界中的网络情况，NIA 在某些场景下可能表现不佳.

其次，决策树策略是通过学生驱动的模仿学习从教师模型中蒸馏得出的. 因此，教师模型的性能是关键因素之一. 如果教师模型本身无法适应多个网络场景，在蒸馏过程中，教师模型策略的缺陷可能会传递给决策树策略，影响性能.

最后，NIA 的训练方式限制了其对网络环境的动态适应能力. 由于采用了固定的人工网络环境，NIA 无法在训练过程中动态添加或删除某类网络环境. 若新增新网络环境，就需要从头开始重新蒸馏决策树策略，增加训练成本.

8　结　语

本文提出了一个 Data-free 的蒸馏框架 NIA 用于训练基于决策树的 ABR 算法. 为了进一步提升决策树策略在真实场景下的性能与泛化性，NIA 设计了环境网络生成模块生成多样的网络环境，利用环境选择模块进一步准确地在训练期间选择适合的网络环境，并提出高效的学生驱动模仿学习训练决策树策略. 实验结果表明，提出的框架能够大幅度改

善训练的决策树的泛化性,在低带宽与高带宽数据集中取得了较好的性能.

参 考 文 献

[1] Sandvine. 2020 COVID-19 phenomena spotlight report. 2020. https://www.sandvine.com/covid-internet-spotlight-report

[2] Sandvine. The global Internet phenomena report January 2022. 2022. https://www.sandvine.com

[3] Mondal A, Sengupta S, Reddy B R, et al. Candid with YouTube: Adaptive streaming behavior and implications on data consumption//Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video. Taipei, China, 2017: 19-24

[4] Huang T Y, Ekanadham C, Berglund A J, et al. Hindsight: Evaluate video bitrate adaptation at scale//Proceedings of the 10th ACM Multimedia Systems Conference. Massachusetts, USA, 2019: 86-97

[5] Mao H, Chen S, Dimmery D, et al. Real-world video adaptation with reinforcement learning. arXiv preprint arXiv: 2008.12858, 2020

[6] Taleb A, Taani B, Begen A C, et al. A survey on bitrate adaptation schemes for streaming media over HTTP. IEEE Communications Surveys & Tutorials, 2018, 21(1): 562-585

[7] Li Z, Zhu X, Gahm J, et al. Probe and adapt: Rate adaptation for HTTP video streaming at scale. IEEE Journal on Selected Areas in Communications, 2014, 32(4): 719-733

[8] Huang T Y, Johari R, Mckeown N, et al. A buffer-based approach to rate adaptation: Evidence from a large video streaming service//Proceedings of the 2014 ACM Conference on SIGCOMM. Illinois, USA, 2014: 187-198

[9] Spiteri K, Urgaonkar R, Sitaraman R K. BOLA: Near-optimal bitrate adaptation for online videos. IEEE/ACM Transactions on Networking, 2020, 28(4): 1698-1711

[10] Yin X, Jindal A, Sekar V, et al. A control-theoretic approach for dynamic adaptive video streaming over HTTP//Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication. London, UK, 2015: 325-338

[11] Mao H, Netravali R, Alizadeh M. Neural adaptive video streaming with pensieve//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. Los Angeles, USA, 2017: 197-210

[12] Huang T, Zhou C, Zhang R X, et al. Comyco: Quality-aware adaptive video streaming via imitation learning//Proceedings of the 27th ACM International Conference on Multimedia. Nice, France, 2019: 429-437

[13] Huang T, Sun L. DeepMPC: A mixture ABR approach via deep learning and MPC//Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP). Abu Dhabi, United Arab Emirates, 2020: 1231-1235

[14] Huang T, Zhou C, Zhang R X, et al. Stick: A harmonious fusion of buffer-based and learning-based approach for adaptive streaming//Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications. Toronto, Canada, 2020: 1967-1976

[15] Huang T, Zhang R, Sun L. Zwei: A self-play reinforcement learning framework for video transmission services. IEEE Transactions on Multimedia, 2021, 24: 1350-1365

[16] Meng Z, Chen J, Guo Y, et al. Pitree: Practical implementation of ABR algorithms using decision trees//Proceedings of the 27th ACM International Conference on Multimedia. Nice, France, 2019: 2431-2439

[17] Meng Z, Wang M, Bai J, et al. Interpreting deep learning-based networking systems//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. USA, 2020: 154-171

[18] Kuleshov V, Precup D. Algorithms for multi-armed bandit problems. arXiv preprint arXiv:1402.6028, 2014

[19] Waheed T, Bonnell R, Prasher S O, et al. Measuring performance in precision agriculture: CART-A decision tree approach. Agricultural Water Management, 2006, 84(1-2): 173-185

[20] Hssina B, Merbouha A, Ezzikouri H, et al. A comparative study of decision tree ID3 and C4.5. International Journal of Advanced Computer Science and Applications, 2014, 4(2): 13-19

[21] Jiang J, Sekar V, Zhang H. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with Festive. IEEE/ACM Transactions on Networking, 2014, 22(1): 326-340

[22] Spiteri K, Urgaonkar R, Sitaraman R K. BOLA: Near-optimal bitrate adaptation for online videos//Proceedings of the IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. San Francisco, USA, 2016: 1-9

[23] Akhtar Z, Nam Y S, Govindan R, et al. Oboe: Auto-tuning video ABR algorithms to network conditions//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. Budapest, Hungary, 2018: 44-58

[24] Huang T, Zhang R X, Sun L. Self-play reinforcement learning for video transmission//Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video. Istanbul, Turkey, 2020: 7-13

[25] Lv G, Wu Q, Wang W, et al. Lumos: Towards better video

streaming QoE through accurate throughput prediction//Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications. London, UK, 2022: 650-659

[26] Zhang H, Zhou A, Lu J, et al. OnRL: Improving mobile video telephony via online reinforcement learning//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. London, UK, 2020: 1-14

[27] Yan F Y, Ayers H, Zhu C, et al. Learning in situ: A randomized experiment in video streaming//Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20). Santa Clara, USA, 2020: 495-511

[28] Li W, Huang J, Wang S, et al. An apprenticeship learning approach for adaptive video streaming based on chunk quality and user preference. IEEE Transactions on Multimedia, 2022, (1): 1-12

[29] Zuo X, Yang J, Wang M, et al. Adaptive bitrate with user-level QoE preference for video streaming//Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications. London, UK, 2022: 1279-1288

[30] Hu Q, Nori H, Sun P, et al. Primo: Practical learning-augmented systems with interpretable models//Proceedings of the 2022 USENIX Annual Technical Conference (USENIX ATC 22). Carlsbad, USA, 2022: 519-538

[31] Zheng Y, Chen H, Duan Q, et al. Leveraging domain knowledge for robust deep reinforcement learning in networking //Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications. Vancouver, Canada, 2021: 1-10

[32] Alves Esteves J J, Boubendir A, Guillemin F, et al. On the robustness of controlled deep reinforcement learning for slice placement. Journal of Network and Systems Management, 2022, 30: 43. https://doi.org/10.1007/s10922-022-09654-8

[33] Dethise A, Canini M, Narodytska N. Analyzing learning-based networked systems with formal verification//Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications. Vancouver, Canada, 2021: 1-10

[34] Dethise A, Canini M, Kandula S. Cracking open the black box: What observations can tell us about reinforcement learning agents//Proceedings of the 2019 Workshop on Network Meets AI & ML. Beijing, China, 2019: 29-36

[35] Mao H, Negi P, Narayan A, et al. Park: An open platform for learning augmented computer systems//Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 2494-2506

[36] Zheng B, Verma S, Zhou J, et al. Imitation learning: Progress, taxonomies and challenges. arXiv preprint arXiv:2106.12177, 2021

[37] Sun Y, Yin X, Jiang J, et al. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction //Proceedings of the 2016 ACM SIGCOMM Conference. Florianopolis, Brazil, 2016: 272-285

[38] Riiser H, Vigmostad P, Griwodz C, et al. Commute path bandwidth traces from 3G networks: Analysis and applications //Proceedings of the 4th ACM Multimedia Systems Conference. Oslo, Norway, 2013: 114-118

[39] Garivier A, Moulines E. On upper-confidence bound policies for non-stationary bandit problems. arXiv preprint arXiv:0805.3415, 2008

[40] Xu T, Li Z, Yu Y. Error bounds of imitating policies and environments. Advances in Neural Information Processing Systems, 2020, 33(1): 15737-15749

[41] Meng Z, Guo Y, Shen Y, et al. Practically deploying heavy-weight adaptive bitrate algorithms with teacher-student learning. IEEE/ACM Transactions on Networking, 2021, 29(2): 723-736

[42] Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-Softmax. arXiv preprint arXiv:1611.01144, 2016

[43] Spiteri K, Sitaraman R, Sparacio D. From theory to practice: Improving bitrate adaptation in the dash reference player//Proceedings of the 9th ACM Multimedia Systems Conference. Amsterdam, Netherlands, 2018: 123-137

[44] EnvivioDash3. https://dashif.org/, 2016

[45] DASH. Dash. Catalyzing the adoption of MPEG-DASH, 2019. https://dashif.org/

[46] Burger E W, Krishnaswamy P, Schulzrinne H. Measuring broadband america: A retrospective on origins, achievements, and challenges. ACM SIGCOMM Computer Communication Review, 2023, 53(2): 11-21

[47] Narayanan A, Ramadan E, Mehta R, et al. Lumos5G: Mapping and predicting commercial mmWave 5G throughput//Proceedings of the ACM Internet Measurement Conference. New York, USA, 2020: 176-193

[48] Huang T Y, Handigol N, Heller B, et al. Confused, timid, and unstable: Picking a video streaming rate is hard//Proceedings of the 2012 Internet Measurement Conference. Boston, USA, 2012: 225-238

[49] Zhang R X, Ma M, Huang T, et al. A practical learning-based approach for viewer scheduling in the crowdsourced live streaming. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2020, 16(2s): 1-22

[50] Jiang J, Das R, Ananthanarayanan G, et al. VIA: Improving internet telephony call quality using predictive relay selection//Proceedings of the 2016 ACM SIGCOMM Conference. Florianopolis, Brazil, 2016: 286-299

HUANG Tian-Chi, Ph.D. His research work focuses on multimedia network streaming, CDN scheduling, and deep reinforcement learning.

LI Chao-Yang, M. S. candidate. His research work focuses on multimedia network streaming.

ZHANG Rui-Xiao, Ph. D. His research interests lie in the area of the optimization of CDN scheduling and deep learning.

LI Wen-Zhe, M. S. His research work focuses on federated learning.

SUN Li-Feng, Ph. D., professor. His research interests include the area of networked multimedia, CDN scheduling, federated learning, video analytics, etc.

Background

The paper focuses on the learning-based adaptive bitrate (ABR) algorithm and how to optimize it through decision tree distillation technology. In detail, the paper first introduces the development trend of the streaming video industry and the importance of streaming video services. Next, the paper introduces the background and current status of the ABR algorithm, including heuristic algorithms and learning-based algorithms. Then, it discusses the problems of existing ABR algorithms based on decision tree distillation technology, that is, poor generalization ability, especially in high-bandwidth network scenarios. To solve this problem, this paper proposes a data-free lightweight ABR algorithm distillation framework called NIA. The framework consists of three modules, including network environment generation module, environment selection module, and student distillation module. The paper describes the design and implementation of NIA in detail and conducts a large number of experiments. The experimental results show that NIA has good generalization ability in various network environments, and its performance is better than off-the-shelf ABR algorithms. Finally, the paper carefully analyzes and compares the critical parameters of NIA to determine the optimal setting range of each parameter.

This work was supported by the National Natural Science Foundation of China under Grant No. 61936011.