arXiv:2304.03505v2 [cs.MM] 9 Aug 2023

通过跨层优化提升自适应实时视频通信

Yueheng Li，Hao C hen，*IEEE* 会员，Bowei Xu、Zicheng Zhang 和 Zhan Ma，*IEEE* 高级会员 *E*

摘要——**在疫情期间，为了追求不打折扣的体验质量（QoE），有效的自适应码率（ABR）算法或策略对于实时视频通信（RTVC）至关重要。现有的自适应码率方法主要将网络带宽估计与视频编码器控制分离，并根据估计带宽微调视频比特率，假设最大化带宽利用率即可实现最优 QoE。然而，RTVC 系统的 QoE 由压缩视频的质量、视频播放流畅度和交互延迟共同决定。在未全面考虑网络层和视频应用层共同影响的情况下，单纯最大化带宽利用率并不能保证令人满意的 QoE。此外，网络层与视频层的解耦还会因网络与编解码器不协调而进一步恶化用户体验。因此，本文提出了 Palette，一种基于强化学习的自适应码率方案，将网络层和视频应用层的处理统一起来，以直接最大化 QoE；该 QoE 被表述为视频质量、卡顿率和时延的加权函数。为此，本文提出一种跨层优化方法，利用网络状况、视频编码参数和视频内容复杂度等跨层观测，为后续帧推导细粒度的压缩因子。由此，Palette 能够解决网络与编解码器不协调的问题，并最大限度地适应网络波动。在真实环境测试中，与最先进的方案相比，在各种应用场景下，Palette 不仅将卡顿率降低了 3.1%–46.3%、时延降低了 20.2%–50.8%，还在带宽消耗相当的情况下，将视频质量提升了 0.2%–7.2%**。

索引词—**自适应码率、跨层优化、网络状况、视频编码参数和视频内容复杂度**。

一、引言

RECENT 年来，基于实时视频通信（RTVC）系统的应用呈指数级增长，例如在线教育、远程共享、视频会议等。特别是自COVID- 19 疫情暴发以来，由于旅行和社交距离限制 [1]，云端视频会议的使用量增加了约 5× 。这些实时视频通信系统如今已广泛应用于我们的日常生活，用于娱乐和工作，从而导致大量的互联网视频流量消耗 [2]（例如 > 17%）以及全球数十亿美元的收入回报 [3]。

为确保服务提供顺畅且服务质量（体验质量）不受影响，众多自适应码率（自适应比特率）方法应运而生，以应对意外波动的底层异构接入网络（例如WiFi、4G、5G等）中的实时视频通信应用。更重要的是，基于深度学习技术强大的表征能力，各种基于学习的自适应比特率决策方法已经被开发出来，重点支持云游戏 [4], [5], 移动视频电话 [6],电商直播和虚拟互动购物 [7], ，在定量层面平均体现出显著的性能提升，例如更高的网络带宽利用率、更低的时延（以毫秒计）等，相较于默认的基于规则的 Google拥塞控制（GCC） [8] ，后者已被广泛用于众多产品中。

A. 现有自适应比特率方案的现状

优化面向互联网规模实时视频通信系统的自适应比特率策略既复杂又充满挑战。因此，研究人员和从业者往往采用“分而治之”的原则：先将大问题拆解为更小且可解决的模块，以便于实现、快速验证并稳健地复用组件 [9]。因此，现有基于规则和基于学习的自适应比特率方案大多将可用网络带宽（例如带宽目标）的估计与视频编码器码率控制（例如压缩视频比特率）解耦，并简单地假设客户端接收的视频质量与视频码率高度相关，从而将视频码率调整至带宽目标。

然而，实时视频通信系统中终端用户的体验质量受到网络层和应用层多种因素的综合影响，而非仅由视频码率决定。例如，网络拥塞可能导致视频卡顿，从而严重损害感知体验 [10]。与此同时，正如 [11],中的分析推导所示，发送端压缩视频的质量由码率和内容复杂度共同决定。对于高动态云游戏和静态人物视频会议，即使采用相同码率，终端用户感受到的视频质量也会大不相同 [12]。尽管 [4], [7], [13] 中少数自适应比特率研究探讨了卡顿对体验质量的影响，但面向码率的方法仍在很大程度上忽视了跨层因素的复合影响，导致体验质量较差、卡顿率较高且时延较大（见图1）。

另一方面，在获得可用带宽目标后，现有的大多数实时视频通信系统仅依赖视频编码器生成比特率尽可能接近该目标的压缩视频。然而，网络与编解码器不协调（即网络带宽估计可以按毫秒级运行，而视频编码器至少需要一秒

所有作者均来自中国南京 210023 的南京大学。电子邮箱：yueheng.li@smail.nju.edu.cn、chenhao1210@nju.edu.cn、mario-gotze@163.com、zichengzhang@smail.nju.edu.cn、mazhan@nju.edu.cn。

0000–0000/00$00.00 © 2022年 IEEE

甚至数秒才能稳定下来并匹配目标比特率 [6]），往往会导致明显的“适应滞后”，使比特率时常相对于带宽目标发生过冲或欠冲，从而导致视频卡顿频繁发生且时延较高（见图 2）。

B. 我们的方法

因此，本研究提出 Palette[1]，，通过基于强化学习（RL）的方法，联合考虑过去的网络状况（例如丢包率、通过往返时延（RTT）测量的时延以及卡顿率）、视频编码参数（例如帧类型、恒定速率因子（CRF））以及通过空间感知信息（SI）和时间感知信息指数（TI）衡量的视频内容复杂度 [14] ，以确定后续帧适当的细粒度压缩因子，例如恒定速率因子（CRF）。

此外，我们采用多尺度策略，以促进细粒度的帧级码率控制：视频编码器根据上述连续帧组的平均恒定速率因子（CRF），相应地确定每帧的量化参数（QP）。通过应用这种 CRF 码率控制模式，编码器能够优化压缩视频的感知质量，使其具有良好的空间和时间连贯性 [15]，从而实现最优 QoE。

如图所示，Palette 将网络带宽估计与视频编码自适应统一起来，从而通过帧级视频码率调整及时响应网络带宽随时间的波动，最大限度地缓解上述“自适应延迟”问题（见图 2）。

为了评估 Palette 的性能，我们开发了*Echo*[2] ——一个基于 WebRTC [16], [17] 的实时视频会议测试平台，作为典型的实时视频通信应用。除 Palette 外，我们还在同一测试平台上部署了另外三种先进的自适应码率方法进行公平比较，包括基于规则的 GCC、基于 [8], 强化学习（RL）的自适应实时流媒体（ARS） [4], 以及基于模仿学习（IL）的 Concerto [6]。实验室管理的轨迹模拟和真实环境现场测试均表明，Palette 具有更优的性能。在轨迹驱动测试中，与当前先进算法相比，Palette 平均可将卡顿率和时延分别降低 26.1%–65.3% 和 39.5%–55.1%；在相同视频质量（通过广泛使用的 VMAF 指标，即视频多方法评估融合（Video Multi-Method Assessment Fusion）进行测量）下，可节省 12.3%–15.2% 的带宽 [18]。至于在各种应用场景中开展的真实环境测试，Palette 不仅可将卡顿率降低 3.1%–46.3%、时延降低 20.2%–50.8%，还可在相近的带宽消耗下将视频质量提升 0.2%–7.2%。

C. 贡献

所提出的 Palette 做出了以下贡献：

- 本研究统一处理网络层和视频应用层，以驱动帧级编码器自适应，有效解决网络与编解码器之间的不协调问题，从而更好地跟上网络波动；
- 本研究通过综合考虑面向体验质量的跨层因素（包括网络状况、视频编码参数和视频内容复杂度）所产生的复合影响，确定少量连续帧的平均恒定速率因子，以实现最佳体验质量，而非最大化网络利用率；
- 实验室模拟和真实世界现场测试均报告称，Palette 的性能优于最先进的方法。

II. 相关工作

近年来，旨在优化视频点播（VoD）和实时视频通信（RTVC）等各种网络视频场景下体验质量的自适应比特率方法迅速增多。以下分别简要介绍了这两类应用中采用基于规则或基于学习的解决方案的代表性案例。然而，实时视频通信服务的自适应比特率方法开发起来更具挑战性，因为视频点播应用中用于缓解网络波动的多秒播放缓冲区通常无法使用，而且在视频编码会话完成之前，自适应比特率引擎往往无法获取实际的压缩视频比特率。

基于规则的方法通常根据从播放缓冲区大小（如果适用）、时延、丢包和吞吐量等网络层面过去观察到的一系列状态，预测未来的网络带宽。对于视频点播服务，最具代表性的例子是MPC（模型预测控制） [19]，，它根据过去的吞吐量估计和播放缓冲区大小，决定下一个分片的流式传输比特率。对于实时视频通信服务，GCC是一种普遍采用的解决方案 [8], ，它利用前几个时隙观测到的端到端时延变化和丢包率，预测未来的带宽。如今，它已被用于WebRTC标准 [16],[17], 作为默认的拥塞控制算法，并已广泛部署于数十亿台设备中。

基于学习的方法最近展现出了卓越的性能。对于具有充足播放缓冲区支持的应用（如视频点播或基于 HTTP 的直播），知名的基于学习的模型包括 Pensieve [20], RLVA [21],QARC [22], Oboe [23], 和 Comyco [24]。其中大多数模型只是调整视频码率以最大化网络利用率，而 QARC 则建议将视频码率映射到相应的VMAF 得分，以优化质量。正如Concerto [6],所示，最大化网络利用率也被扩展应用于对延迟敏感的实时视频通信服务；Concerto使用模仿学习来积极匹配可用网络带宽。同时，基于强化学习的 ARS [4] 将编码码率与视频播放流畅度结合起来进行优化。

近年来，基于规则的方法与基于学习的方法相结合来进行自适应比特率决策，受到了广泛关注，旨在兼取两者之长。例如，Stick [25]通过调整其缓冲区边界，改进了基于缓冲区的自适应码率

[1]调色板用于混合各种颜色，以有意生成更好的颜色；这与所提出的方法大致相同，即联合结合跨层因素的状态，以便为最优 QoE 确定精细化的压缩因子。

[2]我们基于开源 WebRTC 框架开发了 *Echo* ，该框架可在 https://github.com/yuanrongxi/razor 获取。

视频点播流媒体传输算法，而缓冲区边界由额外的强化学习模型确定。对于实时视频通信应用，OnRL [13]采用稳健的混合学习方法，当基于强化学习的算法运行异常时，会切换到 GCC。随后，Loki [7] 提出了一种更深度的融合方案，使基于学习的方法与基于规则的方法能够同步运行，共同确定视频码率以最大限度地提高带宽利用率。[3]

正如文献 [10]–[12], 中所广泛研究的那样，网络视频的体验质量受到卡顿率、时延、比特率和内容复杂度等跨层因素的综合影响。然而，现有的实时视频通信服务比特率自适应方法，包括基于规则的方法（例如GCC）、基于学习的方法（例如 Concerto、ARS）以及混合方法（例如OnRL、Loki），仍然仅估计网络带宽，而其他体验质量因素在很大程度上被忽略。相反，本文通过综合考虑上述各种因素来进行自适应决策，从而实现体验质量优化。

编解码器与传输不协调 最早由 Salsify [26]报告。为解决这一问题，Salsify 提出了逐帧码率自适应方案，以快速响应网络波动。然而，这需要对视频编解码器进行定制，使 Salsify无法兼容通用协议。尽管也有大量研究在 Concerto [6], 中开展，以分析这种编解码器与传输不协调inConcerto [6], it，但由于其仍完全依赖视频编码器执行码率控制，因此并未提出根本性的解决方案，并且不可避免地存在自适应延迟（见图2）。这种编解码器与传输不协调主要是由于网络带宽估计与视频编码器码率控制的处理相互解耦所导致的。因此，本文统一网络- 和视频应用层的处理，通过执行细粒度的帧级压缩因子确定，从根本上解决编解码器与传输不协调问题，并提升整体体验质量。

III. 测量与观察

在本节中，我们进行了全面测量，以考察实时视频通信服务中现有的自适应比特率方法，并了解其局限性。

A. 测量设置

三种常见的自适应码率方法，例如 GCC [8], ARS [4]和 Concerto [6] 被部署在 *Echo* 上，以研究它们的局限性。我们运行

Echo 为简单起见，我们在实验室管理的环境中使用这些自适应比特率方案。为了尽可能真实地模拟现实世界中的网络行为，我们通过合成真实网络轨迹（例如，HSDPA [27], Oboe [23],FCC [28], 和 MMGC2019 [29]）生成典型的轨迹片段。然后，我们使用广泛采用的 Linux 流量控制（TC）工具 [30] ，按照合成的轨迹片段模拟网络。有关*Echo* 和网络轨迹的更多详细信息，请参见第V-A节。

在这项测量中，两台个人电脑（PC）上均安装了*Echo* ，并通过以太网

[3]OnRL和 Loki都不是开源项目，其论文中展示的性能结果是在定制的商用系统中评估的。

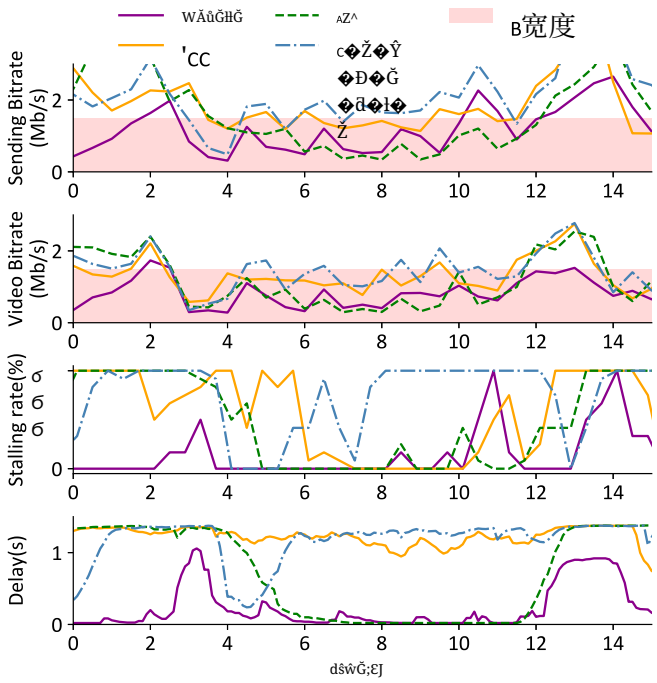


图1. 最大化带宽利用率并不能确保最优体验质量。GCC [8] 和 Concerto [6] 主要优化带宽利用率，因此产生了明显的视频码率超调，导致视频卡顿和高时延。采用在优化过程中纳入卡顿率的ARS [4] 可获得更好的性能。Palette通过综合考虑跨层体验质量因素，呈现出最低的卡顿率（定义为卡顿时长与总播放时长之比）和最小的时延（通过往返时延测量）。使用固定带宽对具有动态场景的游戏视频进行实验，以避免网络波动引起的干扰。

交换机互连，其中一台设置为视频发送端，另一台设置为接收端。发送端对预先缓存的视频文件渲染出的实时场景进行编码，并将其流式传输到接收端，以便实时观看。我们使用多种视频内容来模拟不同的应用场景（例如会议、游戏等）。

B. 观察结果

*1)*最大化带宽利用率并不能确保最优体验质量： 最大化体验质量是实时视频通信应用为取得成功而持续追求的目标。主流自适应码率方法选择将视频码率调整至带宽目标（即最大化网络带宽利用率），其依据的简单假设是，更高的视频码率能够带来更好的视频质量和体验质量。

然而，实时视频通信服务中的体验质量由压缩视频质量、视频播放流畅度和交互延迟共同决定，因此会受到多种因素影响，例如视频内容复杂度、卡顿率等。比如，对动态且复杂的内容进行编码时，视频码率会剧烈变化，容易引发比特率超调。同时，在具有典型流量突发特征的实时视频通信场景中，一味追求更高的带宽利用率会进一步增加比特率超调的风险 [6]。比特率超调通常会导致视频卡顿和高时延事件，严重损害体验质量。

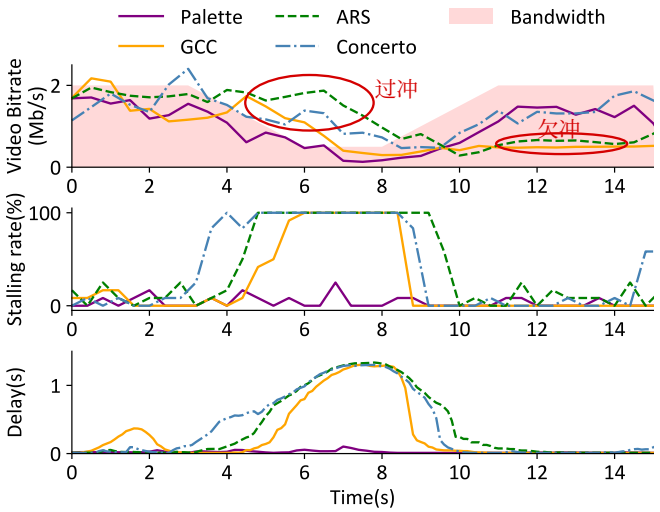


图 2。自适应延迟。由于网络与编解码器不协调，GCC [8], ARS [4], 和 Concerto [6] 通常会出现视频码率超调，从而导致卡顿率和时延显著增加。相比之下，所提出的 Palette 统一了网络层和视频应用层，以自适应帧级 CRF，因此能够及时响应网络波动，同时显著减少卡顿并降低时延。使用包含静态内容的视频，仅用于研究网络波动的影响。

为更好地理解这一问题，我们在固定带宽（例如 1.5Mbps）的网络轨迹下，使用一段具有高度动态场景的第一人称射击游戏视频运行 GCC、ARS 和 Concerto。保持带宽恒定可以避免网络波动带来的潜在影响，使我们能够专注于其他与体验质量相关的因素。结果如图1所示。可以看到，GCC 和 Concerto 的视频码率较高，明显的发送码率过冲导致严重的视频卡顿和高时延（例如从 6s 持续到结尾）。这是因为 GCC 和 Concerto 仅通过调整码率来最大化带宽利用率。随后，由于 ARS [4]除了码率之外，还将卡顿率纳入优化考量，因此在 5s 到 10s 期间，卡顿率和时延明显降低。但 ARS 仍未能避免约 2s 处以及 12s 到 14s 期间的比特率超调，而这些时段的测试视频中出现了快速的场景变化。

如图所示，仅调整视频码率以最大化带宽利用率，并不足以确保体验质量不受影响。更好的方法是直接优化体验质量，充分考虑网络和视频应用层的综合影响。

2）自适应延迟：在现有的自适应码率方法中，网络带宽估计与视频编码器控制模块彼此独立，这意味着我们仅依靠视频编码器在获取带宽目标后执行码率控制 [4], [6], [8]。然而，网络与编解码器不协调 [26]会导致显著的“自适应延迟”和严重的体验质量下降。

为了全面了解“自适应延迟”的影响，我们首先生成一个合成网络轨迹片段，尽可能模拟实际中的典型网络波动。如图 2 上半部分所示，可用带宽从 3 秒到 6 秒逐渐从 2Mbps 降至 0.5Mbps，随后在约 8s，直到 11s 时完全恢复。

如图所示，GCC、ARS 和 Concerto 各自对视频码率的调整明显滞后于网络波动，导致网络带宽下降时出现比特率超调，同时卡顿率和时延也急剧上升，如图 2 的中部和底部所示。网络逐渐恢复时也会出现这种滞后，从而导致码率欠冲。

我们注意到，尽管自适应延迟导致的码率过冲和欠冲总是会出现，但与 GCC和ARS 相比，Concerto [6]中的压缩视频比特率相对更接近实际网络带宽。由于我们在 *Echo*上使用相同的编码器，这一结果表明，Concerto 能够更好地估计可用网络带宽。

总而言之，这表明人们非常需要细粒度的视频码率控制，以便无延迟地即时响应网络动态。

IV. Palette 设计

为解决第III-B节前述现有自适应码率方案的两个关键问题，Palette旨在同时优化应用层和传输层中的端到端体验质量，其目标可归纳为避免视频卡顿和时延，同时最大化画质。Palette通过在帧级别直接决定编码参数（例如恒定速率因子），实现对实际视频比特率的细粒度调整，避免编解码器固有码率控制的长期收敛过程所导致的“自适应延迟”（第IV-A节）。此外，Palette并非最大化带宽利用率，而是通过充分考虑从网络和视频应用层观测到的、面向体验质量的跨层因素，直接优化体验质量（第IV-B节）。Palette利用深度强化学习（DRL）生成端到端自适应码率策略，该策略由神经网络表示（第IV-C节）。

A. 系统架构

Palette 的架构如图 3 所示。首先，Palette 持续从环境中收集跨层因素的历史状态，包括网络状况、视频编码参数和视频内容复杂度（从原始帧中提取）。然后，Palette 依靠强化学习智能体将观测到的状态映射为精细的压缩因子（例如后续帧的平均恒定速率因子），再将其转换为逐帧量化参数，以压缩未来的视频帧并通过网络传输。该决策会传递给编解码器，并用于编码未来的视频帧，直到作出新的决策。我们还设计了一个帧级模拟器来加速训练过程，该模拟器忠实地模拟了真实端到端 RTVC 会话的过程。

B. 面向体验质量的跨层因素

所提出的 Palette 旨在直接优化体验质量，而非像现有工作那样最大化网络利用率。这促使我们全面定义一组具有实际可测量性、

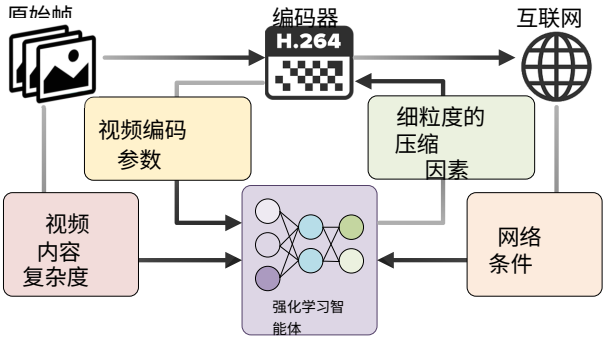


图 3。*Palette* 利用跨层因素决定实现最优 QoE 的细粒度压缩因子。训练强化学习智能体以捕获状态并反复执行动作。以符合 H.264/AVC 标准的 x264 为例，其他视频编解码器也可以轻松扩展。

且可轻松纳入强化学习引擎、用于压缩决策和体验质量优化的跨层因素。参照第III-B节中的讨论，实时视频通信系统中网络视频的体验质量可由压缩视频质量、视频播放流畅度和交互延迟来建模，而这些指标主要受网络、视频编码器和视频内容中的一个或多个因素影响。

网络状况。网络劣化会导致视频数据包丢失，严重降低接收端 [10], [31]的视频质量感知。我们首先纳入往返时延 d和丢包率 p。请注意，这些因素也用于其他自适应比特率方案中，并且现在可以在 WebRTC 客户端中轻松测量。此外，接收端流畅的视频播放对体验质量感知 [4]起着至关重要的作用。然而，往返时延和丢包率等传输层因素无法完全反映接收端客户端 [7]中的播放状态。因此，我们还纳入视频卡顿率 h，其定义为预定义时间间隔内（如本文使用的 0.2 秒）卡顿帧数量（n'）与视频帧总数（n）的比值，例如 $h = n'/n$。尽管视频卡顿用于反映播放状态，但我们仍将其归类为网络因素，因为其发生与网络行为高度相关。

视频编码参数。不同于现有仅通过比特率对视频质量建模的自适应码率方法，我们按照 [15]的建议，使用编码器压缩因子来推断压缩视频的质量。一个常用的因子是 CRF f （或等效的 QP），它决定了视频帧的压缩级别和重建质量。为了充分利用视频中的时空相关性，常见的视频编码器通常会根据场景变化，自适应地采用 I 帧（或帧内帧）或 P 帧（或帧间帧）编码。请注意，I 帧只能利用空间相关性，而 P 帧则可以利用空间相关性和时间预测，因此两者的压缩效率不同。即使 I 帧和 P 帧使用相同的 CRF，它们重建后的质量也可能大不相同。因此，我们的工作也使用帧类型 i。

视频内容复杂度。根据上述讨论，I帧或P帧均采用预测编码来进行冗余消除，其效率高度取决于-取决于内容复杂度。理想情况下，结构简单的相对静止内容更易于压缩；否则，具有复杂纹理的高运动帧通常难以编码，从而很可能产生不同的感知体验 [15]。此外，对具有复杂场景的视频进行编码通常会随着场景变化出现比特率波动（例如，如图1上半部分所示，在约2秒和13秒处出现比特率跳变），这可能导致由比特率超调引起的卡顿率和延迟增加。

正如 [12], [15], 中通过分析推导的那样，压缩视频的感知质量和比特率都可以很好地建模为压缩因子（如 CRF 或 QP）以及视频内容复杂度的函数。因此，我们建议将视频内容复杂度纳入 Palette，以便更好地调整压缩视频的比特率和质量。为简化处理，本工作采用 ITU-R BT.1788 [14] 中标准化的常用 SI u 和 TI v 指标来表示内容复杂度。我们在时间和空间维度上对原始视频帧进行下采样，再在 Palette 中计算 SI 和 TI，因为与使用未下采样视频计算 SI/TI 相比，结果差异可以忽略不计，而且计算开销要小得多。

C. 深度强化学习模型

Palette 使用深度强化学习生成自适应比特率策略，这种方法在以往的自适应比特率研究中展现出优异的性能 [4], [20]。在每个时间步 t，强化学习智能体会从实时视频通信会话中的环境观察状态 s_t，并根据其策略 $\pi_\theta(s_t, a_t)$（策略参数为 θ）采取动作 a_t。应用动作后，环境会转移到新状态 s_{t+1}，并向强化学习智能体返回奖励 r_t。这种离散化交互会在各个间隔之间反复进行。默认情况下，我们将间隔设为 0.2 秒，约等于 30 Hz 视频的 6 帧。在此，“时间步”可以与“间隔”关联起来，以简化符号表示。Palette 使用神经网络（NN）表示其策略，以充分发挥其强大的表征能力，在迭代过程中捕捉动态行为。以下段落将详细介绍 Palette 使用的状态、动作、奖励和神经网络模型。

状态。t 处的状态定义为s_t=($\vec{u_t}$、$\vec{v_t}$、$\vec{i_t}$、$\vec{f_t}$、$\vec{h_t}$、$\vec{d_t}$、$\vec{p_t}$)，其中每一项都是在过去 k 个时间间隔内观测到的值组成的向量（例如，$\vec{u_t} = \{u_{t-k+1}, \ldots, u_t\}$）。在这里，对于同一个 t时间间隔，u_t 和 v_t 分别表示其中原始帧的平均 SI 和平均 TI； i_t 是一个二值标志，用于表示 I帧是否出现。如果标记了 I帧，该标志为 1；否则为 0。 f_t 是应用于 t个时间间隔内所有帧的 CRF 值，而 h_t、 d_t 和 p_t 分别是在该时间间隔内计算得到的卡顿率、平均往返时延和丢包率。所有这些值都会归一化到 [0,1] 的范围内，以用于模型训练。在本文中，我们将 $k = 6$ 设置为捕获过去 6 个时间间隔内的动态变化，其持续时间为 1.2 秒（6×0.2 秒）。

动作。Palette的神经网络输出是动作的概率分布。为了确保时间平滑性，相邻时间间隔之间恒定速率因子的变化

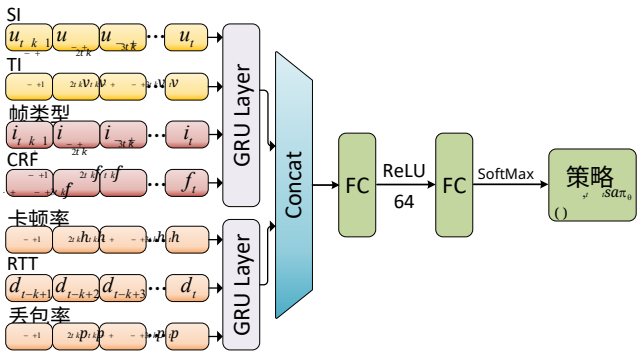


图 4. 神经模型。 Palette 将神经网络（NN）应用于映射过去观测到的跨层因素状态，从而推断动作（例如，本研究中的 CRF 确定）。

应尽可能小。因此，我们将delta 恒定速率因子设为动作集，即 $a_t \in \{+8, +4, +2, 0, -1, -2, -4\}$。强化学习智能体根据动作概率选择其中一个，然后视频编码器立即推导出恒定速率因子参数。目前，Palette 每 0.2秒收集一次状态并执行动作。其他间隔设置的影响将在第 V-D 节中讨论。

奖励。 在强化学习任务中，智能体通过最大化从环境中获得的预期累积（折扣）奖励来学习最优策略。因此，我们设置奖励r_t 以反映实时 RTVC 会话中前一时间间隔的体验质量。对于实时视频通信服务，编码视频的质量、视频播放流畅度和视频延迟是与体验质量相关的三个关键因素。随后，我们将 Palette 中的奖励函数定义如下：

$$r_t = \lambda q_t - \mu d_t - \nu h_t. \tag{1}$$

其中， q_t 是时间步t的编码 CRF 值与最大 CRF 值（例如 x264 中的 51）之差，它直接反映编码视频的质量，已有大量研究对此进行了探讨 [12], [15], [32]。 d_t 表示平均往返时延（毫秒），用于近似实时视频通信服务中发送端与接收端之间的交互延迟。 h_t 表示卡顿率，它直接影响视频播放流畅度。权重参数 λ、 μ和ν 对策略学习有显著影响。例如，较大的 λ 会产生追求高视频质量的“激进”策略，而较大的 μ或 ν 则会产生优先确保视频播放流畅度的“保守”策略。在 Palette 中，经过大量仿真，我们将这些参数的经验值设为 10、0。12 和 70。

神经网络架构。 Palette 使用异步优势演员-评论家 (A3C [33]) 算法来学习其策略。我们还为 Palette 设计了一种轻量级神经网络结构，以从输入状态中提取特征，并对其进行处理，从而快速有效地做出决策。图 4 展示了演员网络的结构。具体而言，我们采用两个各含 16 个单元的门控循环单元层，分别从输入状态中提取视频内容/编码器特征和网络相关特征。随后，这些特征由展平层拼接，并传入一个含 32 个神经元的全连接网络。两个层均使用修正线性函数。以修正线性单元（ReLU）作为激活函数。在输出层，我们采用全连接网络和 SoftMax 激活函数，得到一个 7 维向量，用于表示执行每个动作的概率。对于评论家网络，我们使用与动作网络相同的神经网络结构，但输出层除外；输出层设计为不带激活函数的线性神经元。

请注意，我们还尝试过使用层数更多的神经网络，但实验表明，更深的网络并未显著提升性能，反而大幅增加了计算开销。另一方面，我们也尝试使用一个通用的门控循环单元层，从所有输入状态中共同提取特征，但其性能不如 Palette 中的默认设置；默认设置使用独立的门控循环单元分别表征网络和视频应用层的行为。

*D. Palette*训练

在本文中，我们使用策略梯度方法作为训练策略 [34]。策略梯度方法的核心思想是，通过观察期望总奖励来估计总回报增长最快的路径，并根据这条路径更新网络参数。累积奖励相对于策略参数 θ的梯度可表示如下 [20], [35]:

$$\nabla_\theta E_{\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right] = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(a|s)]. \tag{2}$$

这里， $A^{\pi_\theta}(a|s)$ 是优势函数，用于衡量在给定状态 s 下，某个动作 a 相较于其他动作的优势。我们使用 $A(a_t|s_t)$ 作为 $A^{\pi_\theta}(a_t|s_t)$ 的无偏估计，其通过价值函数 （记为 $V(s_t;\theta_v)$）进行经验计算，具体如下：

$$A(a_t|s_t) = r_t + \gamma V(s_{t+1};\theta_v) - V(s_t;\theta_v). \tag{3}$$

评论家网络从经验观测到的奖励中学习 $V(s_t;\theta_v)$ 的估计值，并输出估计的优势，以指导演员网络的训练。因此，演员网络参数 θ 和评论家网络参数 θ_v 的更新公式可以分别使用 [33] 表示：

$$\begin{aligned} d\theta \leftarrow d\theta + \alpha \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) A(a_t|s_t) \\ + \beta \nabla_\theta H(\pi_\theta(\cdot|s_t)), \end{aligned} \tag{4}$$

$$d\theta_v \leftarrow d\theta_v - \alpha' \sum_t \nabla_{\theta_v} \Big(\sum_{i=0}^{t} \gamma^{t-i} r_i - V(s_t;\theta_v)\Big)^2, \tag{5}$$

其中， α 和 α' 分别是评论家网络和演员网络的学习码率。根据 [20], [35], ，我们引入策略的熵（记为 $H(\cdot)$），以避免在训练早期收敛到次优策略，而 β 是熵项的权重。在本文中，我们将 α =0.00025、 $\alpha' = 0.0015$、 $\beta = 0.15$和 $\gamma = 0.9$配置为相应值。关于训练的更多细节可参见 [20], [33]。请注意，训练策略并非本文的重点，也可以替换为其他方法。

为加快训练过程，我们设计了一个基于轨迹的模拟器，无需实际传输视频帧。在模拟器中，我们仅根据时间对齐的视频码率和带宽采样，计算丢包率、往返时延及其他网络指标。我们还根据视频配置文件的帧级记录，收集视频内容和编码器相关状态。为了进一步加快训练，我们部署了 16 个工作智能体，并行独立探索环境，同时由一个中央智能体汇总所有经验，生成单一的自适应码率算法模型。

我们在一台配备 16 线程 Intel E5-2620 CPU 和 Nvidia GTX-1080TiGPU的服务器上训练 Palette。完成模型训练大约需要 5 小时。我们多次进行训练，每次报告的时间成本都相近，表明 Palette 能够稳定地收敛到最优策略。

V. 评估

我们实现了一个基于 WebRTC 的测试平台——*Echo* ，以便广泛且公正地与其他常见的自适应码率方法开展对比研究。我们全面考察了实验室管理的轨迹模拟，以及真实环境中的实地试验。

A. 方法论

测试平台。 *Echo* 是符合 WebRTC标准 [17]的实时视频会议测试平台，如图 5 所示。它由发送端客户端、接收端客户端以及中间的服务器节点组成，用于模拟或体验真实世界的网络连接。*Echo* 使用默认且应用最广泛的 H.264/AVC [36] 兼容的x264 [37] as底层视频编码器。由于我们的解决方案不要求对视频编解码器进行任何特殊定制，因此可以轻松启用 VP8 [38] 等其他视频编解码器。

对于实验室中的轨迹驱动测试，我们在不同的 Windows PC 上部署两个客户端，并在同一局域网内的一台 Linux PC 上部署服务器。我们在服务器节点上运行 Linux TC 工具，根据轨迹模拟网络行为。在真实环境测试中，我们从一家领先的互联网服务供应商处租用商用云服务器，并在各种环境（步行、乘坐公交车/地铁、咖啡馆等）中，通过 4G、5G 或WiFi 接入为服务器到接收端设置下行链路，以模拟各种真实环境场景。需要注意的是，发送端和接收端客户端部署在全球不同城市。我们直接将使用训练数据集训练的模型部署在接收端客户端中，未进行进一步训练或针对不同网络场景进行调优。

如第 III-A 节的测量研究所述，我们播放预缓存的视频文件以简化模拟。发送端实时编码现场场景，然后将视频流传输到接收端客户端。为模拟多种场景，我们从 YouTube 用户生成内容（UGC）视频数据集 [39]，中随机选取视频片段，包括游戏、电视节目、动画等，组成一段 1 分钟长的视频，然后在持续约 10 分钟的实时视频通信会话中反复播放。为尽可能真实地模拟大规模互联网中的实时视频通信应用，我们将这段长视频配置为以

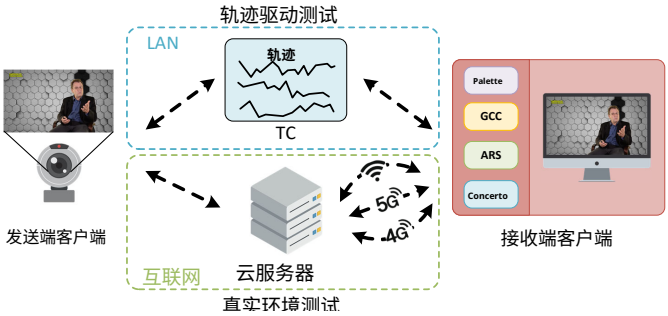


图 5。*Echo* 是一个符合 WebRTC 标准的实时视频会议测试平台。通过调整服务器节点的位置，我们可以在受控实验室中进行轨迹驱动仿真，也可以直接在真实环境中开展实地测试。本研究部署了四种自适应比特率方案。我们之后会向公众开放该测试平台，以支持可复现的研究。

一种相对较低的分辨率（即 720P），以确保视频质量和播放流畅性。

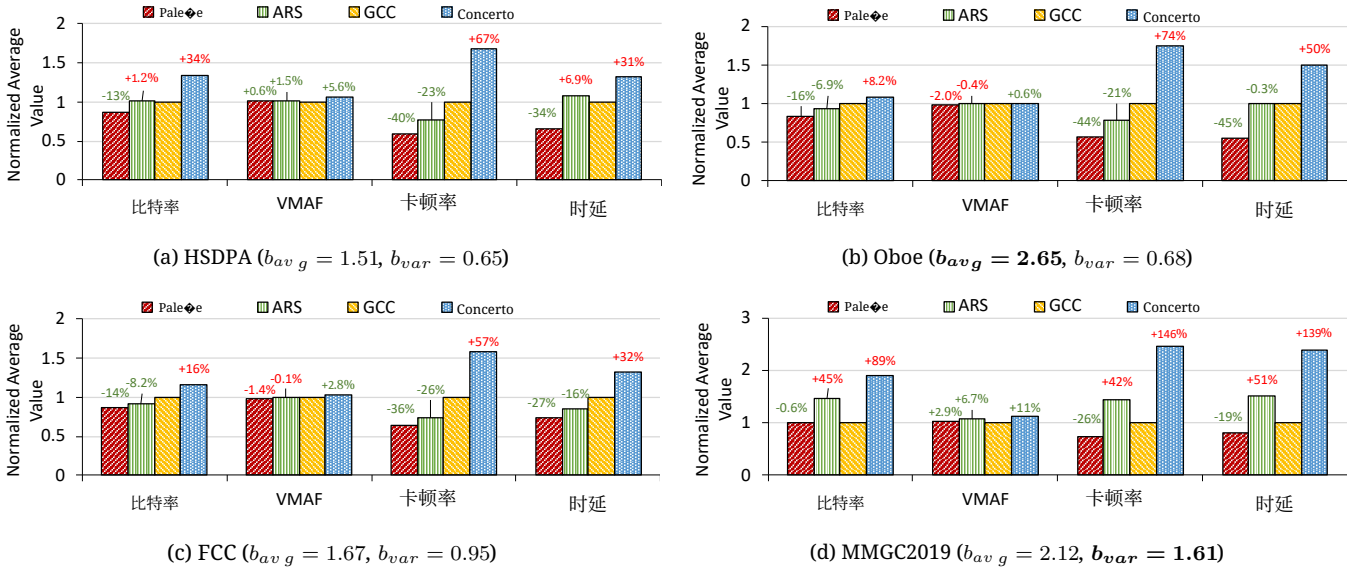
在接收端客户端，我们收集 WebRTC 内部机制测得的往返时延和丢包率 [16]，并主动测量卡顿率以及视频消费情况。所有四种自适应码率方法均在接收端客户端中设计，以供评估。请注意，我们也可以将这些自适应比特率方案部署在发送端或中间云节点上 [4]。

网络轨迹。与其他研究类似 [4], [6], [8], ，我们使用广为人知的公共网络轨迹进行实验室模拟，包括：(1) HSDPA [27]：使用移动设备在乘坐车辆时采集的第三代移动通信/HSDPA 网络轨迹数据集；(2) Oboe [23]：在台式机和移动设备上采集的无线保真/蜂窝网络轨迹数据集，其平均网络带宽相对较高；(3) FCC [28]：包含超过 100 万条轨迹的宽带网络轨迹数据集；以及 (4) MMGC 2019 [29]：为直播视频流媒体竞赛采集的轨迹，其带宽变化幅度更大，覆盖了更广泛的场景。可见，这些轨迹采集自各种接入网络，服务于不同应用，能够很好地模拟真实世界的网络行为。我们在图6的副标题中列出了这些轨迹的网络带宽均值和方差。

值得注意的是，用于训练的网络轨迹会影响自适应码率策略学习。例如，当网络带宽长时间保持在极高或极低水平时，始终选择最佳或最差的视频质量就是最优方案，这会妨碍训练快速收敛。为避免此类情况，我们过滤掉最小网络带宽低于 0.5Mbps 且平均吞吐量高于 4.0Mbps 的网络轨迹。所有使用的轨迹也具有精细的时间粒度，网络探测和带宽采集间隔均小于 0.5s。按照先前工作的建议，我们将所有轨迹随机分为两部分，其中 80% 用于训练，其余 20% 用于测试。总体而言，我们的训练集和测试集分别包含约 500 条和 120 条网络轨迹。

最先进的自适应码率方法用于实时视频通信场景。

- **Google拥塞控制（GCC）** [8] 是一种广泛采用的基于规则的自适应比特率方法，首先动态估计——



(a) HSDPA ($b_{av\,g} = 1.51$, $b_{var} = 0.65$)

(b) Oboe ($\boldsymbol{b_{avg} = 2.65}$, $b_{var} = 0.68$)

(c) FCC ($b_{av\,g} = 1.67$, $b_{var} = 0.95$)

(d) MMGC2019 ($b_{av\,g} = 2.12$, $\boldsymbol{b_{var} = 1.61}$)

图6. 平均增益。 对同一类别中的所有轨迹指标取平均值，然后以使用 GCC 作为 *Echo* 上自适应比特率方案的锚点进行归一化。还提供了每个轨迹数据集的平均带宽 b_{avg} 和带宽变化 b_{var}，单位为 Mbps。

网络带宽，其依据是端到端时延变化[4] 和丢包率，然后根据带宽估计值调整视频码率。它作为默认的码率自适应算法集成于 WebRTC 协议中。WebRTC 已成为事实上的标准，几乎所有主流网络浏览器都支持它，以维持实时视频通信服务。

- **自适应实时流传输（ARS）** [4] 是一种基于强化学习的自适应比特率方法。它训练神经网络，根据过去观察到的网络和播放状态，预测下一组图像（GOP）的编码比特率。请注意，ARS 最初是针对云游戏开发的，采用了相对较高的比特率；为进行公平比较，本研究扩大了比特率范围，并使用与 Palette 和 Concerto 相同的数据集对其重新训练。

- **Concerto** [6] 是一种基于强化学习的方法，深入研究了网络传输与视频编解码器之间的协调问题。尽管该方法提出融合网络带宽估计与过去的实际视频码率，以获得更好的预测结果，但它仍依赖底层视频编码器在秒级执行码率控制；同样地，这实际上并未解决跨层协调问题。由于 *Concerto* 并非开源项目，我们根据已发表的技术细节对其进行了忠实复现。

评估指标。由于体验质量（QoE）衡量的是接收端视频服务的最终质量，因此会受到网络层和视频应用层多种因素的影响。因此，要建立一个能够精准量化体验质量的封闭形式模型极其困难。尤其是在实时 RTVC 会话中，网络和视频编码的快速自适应会带来截然不同的感受，当然，体验质量也难以建模。因此，我们建议综合使用几种常用指标，包括平均视频码率、平均视频质量、平均时延和平均卡顿率，以进行更全面的评估。

关于视频质量，由于其他方法无法获得，公式（1）中的恒定速率因子并未使用。因此，为了进行公平比较，我们采用了 VMAF [18], [40], 一种在视频质量评估领域广受认可的客观感知质量指标。先前的研究 [24] 报告称，与视频码率、结构相似性（SSIM） [41], 等其他指标相比，VMAF与人类感知的相关性更高。视频码率和视频质量均在发送端收集，以反映压缩引起的体验质量感受。

平均往返时延表示整个会话期间的平均时延，定义为每个时隙采样的往返时延之和除以采样次数。平均卡顿率可通过卡顿的视频帧数除以视频帧总数计算得出。往返时延和卡顿率均用于衡量网络可能导致的体验质量下降，再结合发送端的 VMAF 和视频码率，形成整体体验质量感知。

B. 实验室中的轨迹驱动测试

我们使用前文提到的轨迹，在不同网络特征下，针对四种自适应码率方法运行 *Echo* 。首先，按轨迹类别对性能指标（例如比特率、VMAF、卡顿率和时延（以往返时延为单位））取平均值，然后以最常用的 GCC 方法为基准进行归一化，如图6所示。此外，图7展示了这些指标的累积分布函数（CDF），其数值为所有轨迹的平均值（但未经归一化）。

平均增益。如图6所清晰报告的那样，所提出的 Palette 带来了最大的增益，始终将卡顿率降低26%-44%，将时延降低19%-45%，将视频码率降低0.6%-16%，同时几乎保持相同的视频

[4]我们使用 采用趋势线滤波器的最新 GCC 实现，用于 s 研究。

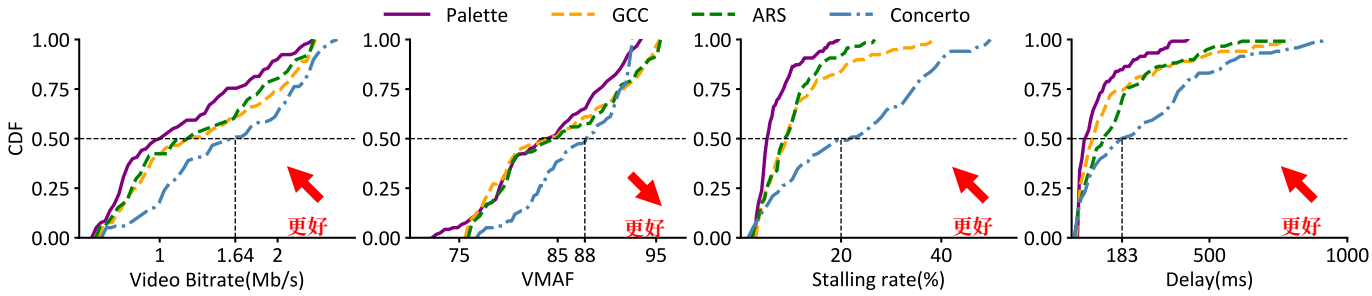


图7. 累积收益。所有轨迹的指标取平均值，并以CDF的形式呈现。

在所有轨迹集上，与 GCC 锚点相比，按 VMAF 测量的帧质量（即变化在±3% 以内）。

尽管对于大多数轨迹，卡顿率和时延的降幅较小，ARS 仍保留了性能增益。相比之下，与 GCC 锚点相比，Concerto 会大幅提高这两项指标。这是因为 ARS 在优化时也考虑了卡顿率，而 Concerto 只是调整比特率以最大化带宽利用率。如图所示，Concerto 生成的视频码率有所提高，这或许能改善视频质量（例如，图6(d)中相较于 GCC 锚点，VMAF 最高提升 11%），但也大大增加了因比特率超调（见图2）而导致卡顿和高时延的可能性。我们特别注意到，在图6(d)的 MMGC2019 轨迹集上，所提出的 Palette 所取得的增益最小；与此同时，ARS 和 Concerto 也都出现了显著的性能损失（例如，相较于 GCC 锚点，卡顿率和时延分别增加了 $\approx 2.5\times$ 和 $\approx 2.4\times$ ）。深入分析各类轨迹后，我们发现，这主要是因为随着带宽变化幅度逐渐增大，网络行为越来越难以刻画。对比图6(a)中 HSDPA 与图6(d)中 MMGC2019 的结果可以看出，网络带宽变化幅度越小，性能越好，反之亦然。

累计收益。 除了前几节讨论的平均收益外，我们再次表明，所提出的 Palette 在整个范围内，相较于其他自适应码率方法，在卡顿率（降低 26.1%-65.3%）和时延（降低 39.5%-55.1%）指标上均表现出明显且稳定的领先优势。我们注意到，Palette、GCC 和 ARS 的 VMAF CDF 几乎重叠，即发送端的视频质量几乎相同，但 Palette 所需的比特率更低（节省 12.3%-15.2%），从而实现了更好的率失真性能 [42]。与上述发现类似，Concerto 在发送端提供了优于 GCC、ARS 和 Palette 的 VMAF，但代价是更高的压缩视频比特率，这可能超过带宽，例如图2和图1中的比特率超调，并大幅增加卡顿率和时延。为便于说明，当 $CDF = 0.5$ 时，我们绘制了一条水平线。尽管 Concerto 的 VMAF 为 88，但时延达到 183ms，卡顿率达到 20%，这意味着由于网络高度拥塞，压缩视频帧可能无法到达接收端客户端。

有趣的是，时延和卡顿率均有界，且与】【。

如图7所示，Palette 中存在一个上限。然而，其他方法则呈现出很长的尾部。这种“长尾问题”在 Loki [7] 中通过融合基于规则的和基于学习的自适应码率方法得到了专门研究。由于本工作主要致力于面向 QoE 的自适应比特率系统开发，因此仅针对 GCC、ARS 和 Concerto开展了比较研究。Palette 中的长尾问题得到部分解决是一个额外的收获，表明 Palette 能够最佳地适应网络波动，从而在不牺牲整体体验质量的情况下运行。像Loki [7]那样彻底解决Palette 的长尾问题不在本工作的范围之内，将作为我们未来的研究加以推进。

C. 真实世界现场测试

按照第V-A节中的设置，我们使用主流接入网络在各种场景下开展真实世界现场测试，包括4G、5G和WiFi，场景涵盖日常经历的咖啡馆、步行以及乘坐公交车/地铁等。除了由第三方互联网云服务提供商托管服务器节点外，*Echo* 接收端还分布在同一大陆内相距数百英里的不同城市，甚至分布在跨大陆相距数千英里的城市。对于每种场景中的每种算法，我们重复进行3次RTVC会话，以确保实验环境对所有算法都是公平的。此外，我们确保所有算法在相同网络类型（4G/5G/WiFi）下连接到同一服务提供商，并在相同移动场景（步行/公交车/地铁）中经历相同的移动路线和速度。每次会话持续10分钟，以充分体验各种视频内容，从而模拟众多应用（例如游戏、会议等）。整个实地测试持续数周。

总体评估。 图8(a)展示了所有会话的平均总体性能。由于编码器侧的VMAF可以使用比特率的对数函数进行单调近似 [18]，且卡顿率和时延的变化趋势与图6中几乎相同，为简洁起见，我们仅绘制VMAF和卡顿率。对于四种自适应码率方法，我们同时使用它们的平均值和标准差，不仅为了了解其平均性能，还为了了解由地理位置、网络、场景等方面的异构连接所导致的性能变化。总体而言，所提出的Palette在所有场景和连接下都具有明显的性能优势，例如更高的VMAF（提升0.2%-7.2%）、更低的卡顿率（降低3.1%-46.3%）以及更低的

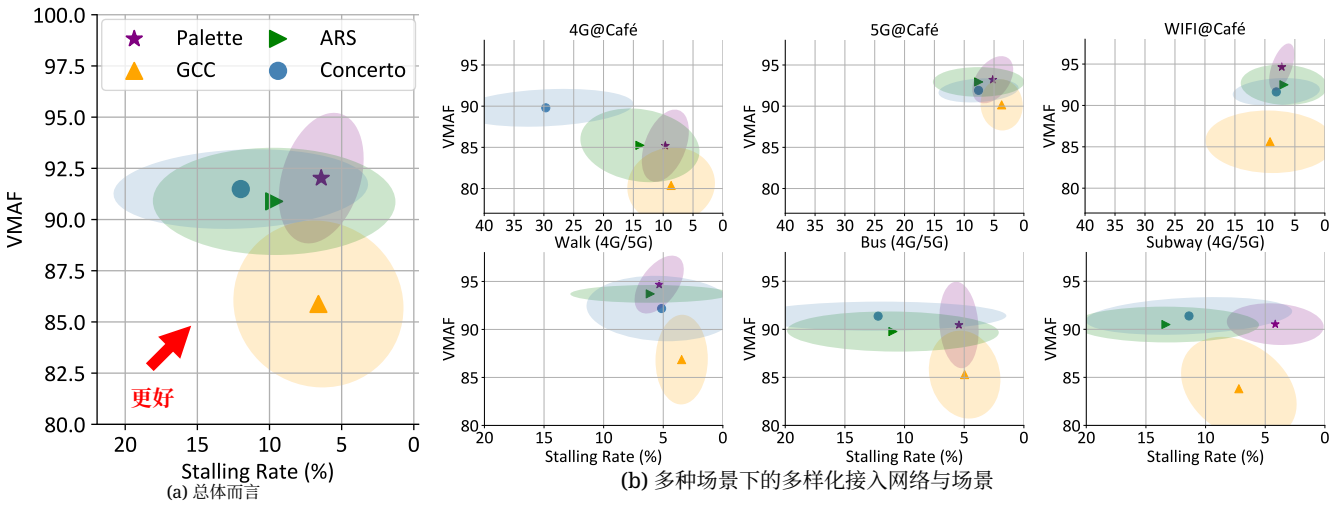


图 8。真实世界现场测试。Palette在各种接入网络和场景中均展现出相较其他方案的一致性能优势。每个椭圆的中心表示配对的 VMAF 和卡顿率的平均值，而椭圆在 x-轴或 $y-$ 轴上的范围分别表示卡顿率或 VMAF 的标准差。

平均时延（降低20.2%-50.8%）（阴影中心）以及 VMAF 和卡顿率更小的标准差（例如，阴影覆盖范围更小）。请注意，此处所有方法的整体性能都远好于轨迹驱动测试中的表现。这是因为真实世界现场测试是在各种高带宽网络连接（例如5G 和 WiFi）下进行的，而这些连接在轨迹驱动测试中被过滤掉了。

逐场景评估。逐场景评估详见图 8(b)。图 8(b)上半部分的三个子图展示了在4G、5G 和 WiFi 接入下的咖啡馆场景，我们希望借此最佳地代表工作场所、家庭等处的静态场景。同样，图8(b)下半部分的三个子图展示了日常生活中广泛经历的热门运动场景，例如步行以及乘坐公交车/地铁，这些场景具有更便利的 4G/5G 连接。

对于 Concerto 和 ARS，卡顿率及其平均值和标准差在 4G@Caf´e、Bus (4G/5G) 和 Subway (4G/5G) 等场景中均大幅上升。这主要是由于这些场景中网络波动频繁，而 Concerto 和 ARS 无法有效应对。例如，在 4G@Caf´e 场景中，连接到同一 4G 移动通信基站的用户数量会随时间动态变化。与 WiFi 和 5G 接入相比，4G技术提供的带宽非常有限，因此用户数量的动态变化很容易导致带宽显著波动。另一方面，在 Bus (4G/5G) 和 Subway (4G/5G) 场景中，车辆行驶过程中通信基站的切换通常会导致网络带宽波动。相比之下，所提出的 Palette 能够很好地捕捉网络波动，并带来更好的体验质量（卡顿率大幅降低，VMAF 相近）。

在 4G@Caf´e 场景中，尽管 Concerto 的 VMAF 表现领先，但它显著提高了卡顿率的平均值和标准差。这与第 V-B 节中的观察结果一致：在 Concerto 中，激进地调整视频码率以最大化网络利用率。可以带来更高的 VMAF，但显然会增加卡顿率。同样，发送端拥有更高的 VMAF 并不一定能保证相同水平的体验质量，因为网络劣化会严重损害服务。由于 5G 和 WiFi 方案能够提供的带宽远高于 4G 网络，因此这些场景中的所有自适应码率方法，例如 5G@Caf´e 和 WiFi@Caf´e，都能提供更出色的表现，带来更高的 VMAF 和更低的卡顿率。

时延抖动。令人惊讶的是，GCC在真实环境测试中的表现与轨迹驱动仿真中的表现不同。如图所示，它保守地倾向于选择较低的视频码率，卡顿率相对较低，但VMAF 显著下降。这是因为 GCC 对网络时延（往返时延）抖动非常敏感，而这种抖动在公共互联网中十分普遍，却无法在基于吞吐量的轨迹中得到良好模拟。

为了获得更多见解，我们对两条相似的网络连接进行了实验。在其中一条连接（“无抖动”）中，前 10 秒（图 9 中虚线之前）的带宽和时延分别固定为 0.4Mbps 和 100ms，之后分别调整为 4Mbps 和 90ms。另一条连接（“有抖动”）的设置相同，唯一不同的是 10 秒后时延在 $90ms \pm 2ms$ 范围内变化。即使时延变化如此之小，我们也已重现出 GCC 在真实环境测试中的行为。

随后，我们分别在上述两种不同设置下测试GCC，即一种存在时延抖动，另一种不存在抖动，并在图9 中绘制随时间变化的视频码率。对于不存在抖动的网络连接（橙色线），GCC 会逐步改善对可用带宽的估计，从而使视频码率不断提高，直到约 55秒时达到目标。相比之下，在存在网络抖动时（绿色线），GCC 通常会先改善带宽估计，然后在一段时间后降低该估计，因而导致类似的视频码率变化（例如，逐渐增加一段时间后再下降）。我们多次运行仿真，观察到了几乎相同的趋势。这主要是由保守的

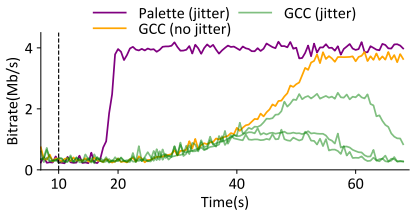


图 9. 时延抖动影响。

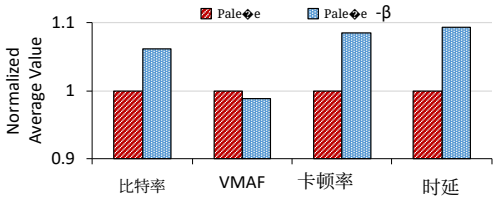


图10. 无 SI& TI 的 Palette。

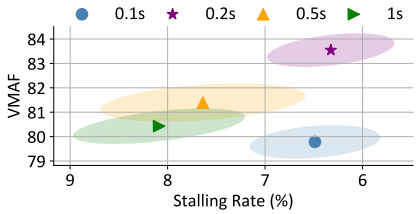


图11。恒定速率因子粒度。

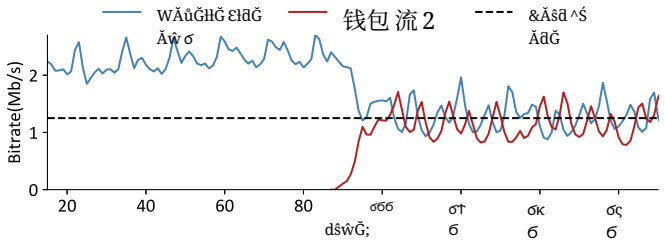


图12。两个 Palette 流之间的竞争。

GCC在发生网络抖动时的表现。不过，无论网络抖动如何，Palette（紫色线）最终都能够恢复，并将视频码率维持在接近带宽的水平。

D. 深入分析 *Palette*

本节深入剖析 Palette，以获取更多洞见。

内容复杂度。在 Palette 中引入SI 和 TI，以反映视频内容对码率/质量控制的贡献。这里我们移除这些内容因素，仅依赖网络状态和视频编码参数进行码率控制，并将其记为 Palette-β。在训练和测试中，Palette-β 和原生 Palette的其他设置均保持不变。我们从第V-A节所讨论的数据集中随机选取相同的轨迹，在这些轨迹下运行这两个版本的Palette，并对其进行比较。如图1所示，Palette-β 使卡顿率和时延的平均值分别增加了 8.6%和9.4%。此外，尽管Palette-β使用了更高的比特率，其 VMAF 仍略有下降。这是因为由于缺少 SI/TI 的嵌入，Palette-β 在压缩过程中无法准确进行码率和体验质量建模，捕捉内容特征的能力较弱，因此在复杂的高运动帧中浪费了更多比特。

恒定速率因子的自适应粒度。所提出的 Palette 支持逐帧压缩自适应：我们通常首先确定一组连续帧的平均恒定速率因子，然后让编码器进一步确定每帧的量化参数。这种多尺度压缩控制使我们能够更好地刻画帧间的时空相关性，从而为压缩因子决策提供依据，以实现最优 QoE。前文假设恒定速率因子的自适应粒度（AG）为 0.2 秒。这里我们进一步探究了 0.1 秒、0.5 秒和 1 秒的其他自适应粒度。如图11所示，默认的 0.2 秒自适应粒度 Palette 比其他自适应粒度设置具有更优的性能。例如，采用其他自适应粒度的 Palette 会降低 VMAF 并提高卡顿率，在一定程度上明显损害最终的 QoE。这是因为较小的自适应粒度，例如，0.1s-AG 只能使用过去 0.6s 内的状态观测，这不足以做出最优决策。虽然更大的 AG（例如 0.5s-AG 和 1s-AG）可以使用丰富的状态信息，但正如第III-B节所述，它们会受到“自适应滞后”的影响（即，网络行为在如此长的时间内已经发生变化）。

多流。为研究 Palette 在多流竞争场景中的性能，我们使用第V-A节所述的测试平台进行了实验。该平台包含两对发送端和接收端，负责传输两路 RTVC 流，且两路流均经过同一个中继服务器。我们在服务器上运行 TC 工具，将带宽限制为恒定的 2.5Mbps。第二路流在第一路流加入约 90 秒后加入。图12展示了两路 Palette 流竞争时的发送比特率。可以看到，第二路流加入后，流1的发送带宽迅速下降，这主要是由竞争导致的拥塞所致。约 15 秒后，两路流均达到公平带宽，并在其附近波动。测得的 Jain 公平性指数 [43] 达到 0.993，表明两路流公平地共享链路带宽。

VI. 讨论

Palette 的通用化。 Palette利用跨层状态（例如网络状况、视频编码参数、视频内容复杂度）来推导实现最优 QoE 的压缩因子。在当前实现中，Palette 调整 x264 中的 CRF 以进行视频压缩，并使用异步优势演员-评论家算法训练跨层状态与 CRF 之间映射关系的策略。Palette还可以轻松扩展以支持其他视频编码器，例如 VP8 [38],x265 [44],AV1 [45]。此外，还可以应用其他先进的强化学习算法，例如PPO [46],DDPG [47], SAC [48],和 TD3 [49],，来训练 Palette。

网络视频的码率控制。我们的广泛研究表明，在 Palette 中使用跨层状态来确定压缩因子，在面对网络带宽变化时，相比视频编码器中嵌入的传统方法（例如 GCC、ARS 和Concerto 中所体现的方法），能够实现更好的码率控制。这是因为：1）尽管视频编码器中的传统码率控制也会利用时间参考帧之间的内容复杂度来更准确地估计码率，但它只能使用标准规范中定义的非常有限数量的参考帧。相比之下，所提出的Palette可以从大量连续帧中收集视频内容复杂度，从而实现更好的码率建模。2）另一方面，Palette学习内容复杂度与压缩因子之间的非线性映射，其性能明显优于简单的二次函数甚至线性函数

应用于视频编码器中。本研究为开发更优的网络视频服务视频编码速率控制提供了全新的视角。

视频质量指标。在 Palette 中，我们在状态空间、动作空间和奖励中使用 CRF 作为质量指标，因为与此前采用比特率作为视频质量指标的方法相比，它能够在不增加任何计算复杂度的情况下，更好地进行视频质量评估。我们承认，与 CRF 相比，VMAF 等全参考指标可能更能反映主观视觉质量。然而，由于全参考指标需要原始图像的副本，而在客户端的实时会话中无法做到这一点，因此 VMAF 不能用于 Palette 的观测状态。

另一方面，考虑到强化学习的特性，在奖励函数中单纯用 VMAF 替换 CRF，未必能有效提升 Palette 的性能，甚至可能降低训练效率。包括 Palette 所使用的 A3C 在内的先进强化学习算法，都采用演员-评论家架构。在这种架构中，评论家的作用是拟合价值函数，直观而言，该函数将观测状态映射到奖励。为了拟合这一函数，神经网络需要充足的输入信息。否则，输入与输出之间的信息差距对智能体而言不过是噪声，进而降低训练效率。例如，VMAF 的计算涉及大量像素级计算，而预测 VMAF 需要充足的像素级信息以及更复杂的信息提取网络。以 QARC [22], 等为代表的先前工作，设计了专门且复杂的神经网络，直接根据过去的无损视频帧预测 VMAF。相比之下，Palette 当前的设计是在图像质量与计算复杂度之间取得平衡。

VII. 结论

我们提出并评估了 Palette，这是一种面向 QoE 的自适应比特率系统，利用来自网络和视频应用层的跨层因素，在实时视频通信应用中决定细粒度的压缩因子，以实现最优 QoE。实验室中的轨迹驱动仿真和真实环境中的测试均表明，在广泛的网络、场景和视频内容下，所提出的 Palette 优于最先进的方法。我们相信，Palette 将为跨层自适应比特率研究带来更多启发。

参考文献

[1] TrustRadius。（2020年4月）2019冠状病毒病软件行业统计数据。 [在线]。可访问：https://www.trustradius.com/vendor-blog/covid-19-software-industry-data-and-statistics [2] 思科。（2022年1月）全球——2022年预测摘要。 [在线]。可访问：https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/ vni-forecast-highlights/pdf/Global 2022 Forecast Highlights.pdf [3] Research and Markets。（2022年1月）全球视频会议市场。 [在线]。可访问：https://www.researchandmarkets.com/ reports/5415305

[4] H. Chen、X. Zhang、Y. Xu、J. Ren、J. Fan、Z. Ma 和 W. Zhang，“T-gaming：大规模高性价比云游戏系统”，IEEE 并行与分布式系统汇刊，第30卷，第12期，第2849–2865页，2019年。

[5] X. Zhang，H. Chen，Y. Zhao，Z. Ma，Y. Xu，H. Huang，H. Yin，and D. O. Wu，“通过移动边缘计算改善云游戏体验”，IEEE Wireless Communications，第26卷，第4期，第178–183页，2019年。[6] A. Zhou、H. Zhang、G. Su、L. Wu、R. Ma、Z. Meng、X. Zhang、X. Xie、H. Ma和X. Chen，“学习协调视频编解码器与传输协议以实现移动视频电话”，载于第25届移动计算与网络国际年会，系列：MobiCom ’19，2019 [ ] 10 1145 3300061 。在线。可访问：https://doi.org/ 。/ 。

[7] 3345430 H. Zhang、A. Zhou、Y. Hu、C. Li、G. Wang、X. Zhang、H. Ma、L. Wu、A. Chen和C. Wu，Loki：通过融合基于规则的模型，提升基于学习的实时视频自适应的长尾性能。美国纽约州纽约：计算机协会，2021年，第775–788页。 [在线]。可访问：https://doi.org/ 10 1145 3447993 3483259 。/ 。Gaetano、L.

[8] De Cicco、S. Holmer和S. Mascolo，“Web实时通信的拥塞控制”，IEEE/ACM网络汇刊，第25卷，第5期，第2629–2642页，2017年10月。 [在线]。可访问：https://doi.org/ 10 1109 2017 2703615 。/TNET。。V.

[9] Goyal，“变换编码的理论基础”，IEEE信号处理杂志，第18卷，第5期，第9–21页，2001年。[10] Y. Qi和M. Dai，“帧冻结和跳帧对视频质量的影响”，载于2006年智能信息隐藏与多媒体国际会议，2006年，第423–426页。[11] H. Hu、Z. Ma和Y. Wang，“码率受限视频编码与可伸缩视频自适应的空间、时间和幅度分辨率优化”，载于2012年第19届IEEE国际图像处理会议。IEEE，2012年，第717–720页。[12] Z. Ma、M. Xu、Y.-F. Ou和Y. Wang，“将压缩视频的码率和感知质量建模为帧率和量化步长的函数及其应用”，IEEE视频技术电路与系统汇刊，第22卷，第5期，第671–682页，2011年。[13] H. Zhang、A. Zhou、J. Lu、R. Ma、Y. Hu、C. Li、X. Zhang、H. Ma和X. Chen，OnRL：通过在线强化学习改善移动视频电话。美国纽约州纽约：计算机协会，2020年。 [在线]。可访问：https://doi.org/10. 1145/3372224.3419186[14] “多媒体应用中视频质量主观评估的方法”，国际电信联盟，瑞士日内瓦，标准，2007年1月。[15] Y.-F. Ou、Z. Ma、T. Liu和Y. Wang，“同时考虑帧率和量化伪影的视频感知质量评估”，IEEE视频技术电路与系统汇刊，第21卷，第3期，第286–298页，2010年。[16] WebRTC。（2021年1月）面向网络的实时通信。 [在线]。可访问：https://webrtc.org/ [17] W. P. Releases。（2021年1月）Web实时通信（webrtc）成为万维网联盟建议标准和互联网工程任务组标准，改变通信格局。 [在线]。可访问：https://www.w3.org/2021/01/pressrelease-webrtc-rec.html.en [18] R. Rassool，“VMAF的可复现性：验证实用感知视频质量指标”，载于2017年IEEE宽带多媒体系统与广播国际研讨会（BMSB），2017年，第1–2页。[19] X. Yin、A. Jindal、V. Sekar和B. Sinopoli，“通过控制理论方法实现基于HTTP的动态自适应视频流”，载于2015年ACM数据通信特别兴趣小组会议论文集，系列：SIGCOMM ’15。美国纽约州纽约：计算机协会，2015年，第325–338页。 [在线]。可访问：https://doi.org/ 10 1145 2785956 2787486 。/ 。H.

[20] Mao、R. Netravali和M. Alizadeh，“基于Pensieve的神经网络自适应视频流”，载于ACM数据通信特别兴趣小组会议论文集，系列：SIGCOMM ’17。美国纽约州纽约：计算机协会，2017年，第 197 210 [ ] 10 1145 3098822 3098843 – 页。在线。可访问：https://doi.org/ 。/ 。Z. Jiang、X. Zhang、Y. Xu、Z. Ma、J.

[21] Sun和Y. Zhang，“基于强化学习的360度视频流码率自适应”，IEEE广播汇刊，第67卷，第2期，第409–423页，2021年。[22] T. Huang、R.-X. Zhang、C. Zhou和L. Sun，“QARC：基于深度强化学习的实时视频流视频质量感知码率控制”，载于第26届ACM多媒体国际会议论文集，系列：MM ’18。美国纽约州纽约：计算机协会，2018年，第1208–1216页。 [在线]。可访问：https://doi.org/ 10 1145 3240508 3240545 。/ 。Z. Akhtar、Y. S.

[23] Nam、R. Govindan、S. Rao、J. Chen、E. Katz-Bassett、B. Ribeiro、J. Zhan和H. Zhang，“Oboe：针对网络状况自动调优视频自适应比特率算法”，载于论文集。

2018年计算机协会数据通信特别兴趣小组会议，系列号 SIGCOMM '18。美国纽约州纽约：计算机协会，2018年，第 44–58 页。 [在线]。可访问：https://doi.org/ 10 1145 3230543 3230558 。/。

[24] T.Huang、C. Zhou、R.-X. Zhang、C. Wu、X.Yao和 L. Sun，"Comyco：通过模仿学习实现质量感知的自适应视频流"，载于第 27届 ACM多媒体国际会议论文集，系列号 MM '19。美国纽约州纽约：计算机协会，2019年，第 429–437 页。[在线]。可访问：https://doi.org/ 10 1145 3343031 3351014 。/。

[25] "Stick：基于缓冲区的方法与基于学习的方法在自适应流中的和谐融合"，载于 IEEEINFOCOM2020 -IEEE计算机通信会议，2020年，第 1967–1976 页。[26] S. Fouladi、J. Emmons、E. Orbay、C. Wu、R.S.Wahby 和K.Winstein，"Salsify：通过更紧密地整合视频编解码器与传输协议实现低延迟网络视频"，载于第 15届USENIX网络系统设计与实现研讨会（NSDI18）。华盛顿州伦顿：USENIX协会，2018年 4 月，第 267–282 页。 [在线]。可访问：https://www.usenix.org/ conference/nsdi18/presentation/fouladi [27] H. Riiser、P. Vigmostad、C. Griwodz 和P.Halvorsen，"来自第三代移动通信网络的通勤路径带宽轨迹：分析与应用"，载于第4 届 ACM多媒体系统会议论文集，系列号MMSys '13。美国纽约州纽约：计算机协会，2013年，第 114–118 页。 [在线]。可访问：https://doi.org/10. 1145/2483977.2483991[28] F. C. Commission。（2016年12月）原始数据——测量美国宽带 2016。[在线]。可访问：https: // www.fcc.gov/reports-research/reports/measuring-broadband-america/raw-data-measuring-broadband-america 2016 [29] 2019, 2019 AItrans。（6 月）ACM多媒体大奖赛挑战——实时视频流。 [在线]。可访问：https://www.aitrans.online/MMGC/ [30] T.L. P. Interface。（2001年 12月）tc(8)——Linux手册页。 [在线]。可访问：https://man7.org/linux/man-pages/man8/tc.8.html[31] X.Feng、T.Liu、D. Yang 和 Y. Wang，"受显著性启发的丢包受损视频全参考质量指标"，IEEE广播汇刊，第57 卷，第 1 期，第 81–88 页，2010年。[32] J.Bienik、M. Uhrina 和 P. Kortis，"恒定速率因子对客观视频质量评估的影响"，电气与电子工程进展，第 15 卷，第 4期，第 673–682 页，2017年。 [在线]。可访问：https://www.proquest. com/scholarly-journals/impact-constant-rate-factor-on-objective-video/docview/2051177953/se-2[33] V. Mnih、A. P. Badia、M. Mirza、A.Graves、T. Lillicrap、T. Harley、D.Silver和K. Kavukcuoglu，"深度强化学习的异步方法"，载于第 33 届国际机器学习会议论文集，系列号机器学习研究论文集，M. F. Balcan 和 K.Q.Weinberger编，第 48 卷。美国纽约州纽约：PMLR，2016年 6月 20–22 日，第 1928–1937 页。[在线]。可访问：https://proceedings.mlr.press/v48/mniha16.html[34] R. S. Sutton、D. McAllester、S. Singh 和Y.Mansour，"采用函数逼近的强化学习策略梯度方法"，载于神经信息处理系统进展，S. Solla、T.Leen 和 K. M¨uller 编，第 12卷。麻省理工学院出版社，1999年。 [在线]。可访问：https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf [35] H.Mao，"利用强化学习进行网络系统优化：方法与应用"，博士论文，麻省理工学院，2020年。[36] T. Wiegand、G. Sullivan、G. Bjontegaard 和 A. Luthra，"H.264/AVC视频编码标准概述"，IEEE视频技术电路与系统汇刊，第 13 卷，第 7 期，第 560–576 页，2003年。[37] VideoLAN。（2013年 8 月）x264，最佳H.264/AVC编码器——VideoLAN。[在线]。可访问：https://www.videolan.org/developers/x264.html[38] J.Bankoski、P. Wilkins 和Y.Xu，"VP8（一种面向网络的开源视频编解码器）的技术概述"，载于2011年IEEE多媒体与博览会国际会议，2011年，第 1–6 页。[39] Y. Wang、S.Inguva 和 B. Adsumilli，"用于视频压缩研究的 YouTube用户生成内容数据集"，载于 2019年 IEEE 第21 届多媒体信号处理国际研讨会（MMSP），2019年，第 1–5页。[40] Z.Li、C. Bampis、J. Novak、A.Aaron、K. Swanson、A. Moorthy 和 J. Cock，"VMAF：旅程仍在继续"，Netflix技术博客，第 25卷，2018年。[41] Z. Wang、A. Bovik、H.Sheikh 和E. Simoncelli，"图像质量评估：从误差可见性到结构相似性"，IEEE图像处理汇刊，第 13 卷，第 4 期，第 600–612 页，2004年。

[42] T. Berger 和 J. Gibson，《有损源编码》，IEEE Transactions on Information Theory，第 44 卷，第 6 期，第 2693–2723 页，1998 年。[43] R. K. Jain、D.-M. W. Chiu、W. R. Hawe 等人，《公平与歧视的定量度量》，1984 年。[44] G. J. Sullivan、J.-R. Ohm、W.-J. Han 和 T. Wiegand，《高效视频编码（HEVC）标准概述》，IEEE视频技术电路与系统汇刊，第 22 卷，第 12 期，第 1649–1668 页，2012 年。[45] Y. Chen、D. Murherjee、J. Han、A. Grange、Y. Xu、Z. Liu、S. Parker、C. Chen、H. Su、U. Joshi、C.-H. Chiang、Y. Wang、P. Wilkins、J. Bankoski、L. Trudeau、N. Egge、J.-M. Valin、T. Davies、S. Midtskogen、A. Norkin 和 P. de Rivaz，《AV1 视频编解码器核心编码工具概述》，载于 2018 年图像编码研讨会（PCS），2018 年，第 41–45 页。[46] J. Schulman、F. Wolski、P. Dhariwal、A. Radford 和 O. Klimov，《近端策略优化算法》，CoRR，第 abs/1707 06347，卷。。在线。2017 [ ] 1707 06347 可访问：http://arxiv.org/abs/ 。

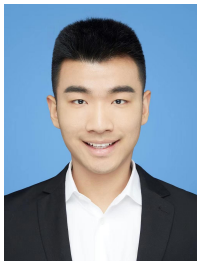
[47] T. P. Lillicrap、J. J. Hunt、A. Pritzel、N. Heess、T. Erez、Y. Tassa、D. Silver 和 D. Wierstra，《基于深度强化学习的连续控制》，arXiv 预印本 arXiv:1509.02971，2015 年。[48] T. Haarnoja、A. Zhou、P. Abbeel 和 S. Levine，《软演员-评论家：基于随机演员的离策略最大熵深度强化学习》，载于第 35 届国际机器学习会议论文集，机器学习研究论文集，J. Dy 和 A. Krause 编，第 80 卷。PMLR，2018 年 7 月 10–15 日，第 1861–1870 页。[在线]。可访问：https://proceedings.mlr.press/v80/haarnoja18b.html [49] S. Fujimoto、H. van Hoof 和 D. Meger，《解决演员-评论家方法中的函数逼近误差》，载于第 35 届国际机器学习会议论文集，机器学习研究论文集，J. Dy 和 A. Krause 编，第 80 卷。PMLR，2018 年 7 月 10–15 日，第 1587–1596 页。 [在线]。可访问：https://proceedings.mlr.press/v80/fujimoto18a.html



Yueheng Li 于2021年获得南京航空航天大学电子与信息工程学院工学学士学位。目前，他正在南京大学电子科学与工程学院攻读理学硕士学位。他的研究兴趣包括实时视频流和实时视频通信。



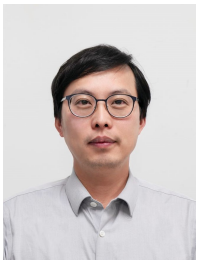
Hao Chen （IEEE会员）于2013年获得中国西北工业大学电子与信息工程专业工学学士学位，并于2020年获得中国上海交通大学博士学位。目前，他任职于南京大学电子科学与工程学院。他的研究方向包括视频流、实时视频传输和机器学习。他是2019年IEEE广播技术学会最佳论文奖的共同获奖者。



Bowei Xu 于2019年获北京邮电大学电子工程学院工学学士学位，并于2022年毕业于北京邮电大学信息与通信工程学院，获硕士学位。目前正在电子科学与工程学院攻读博士学位。



Zicheng Zhang ZichengZhang 于2021年获得苏州大学电子与信息工程学院的工学学士学位。目前，他正在南京大学电子科学与工程学院攻读理学硕士学位。



Zhan Ma （IEEE高级会员）于2004年和2006年分别获得华中科技大学（中国武汉）的理学学士学位和理学硕士学位，并于2011年获得纽约大学（美国纽约州纽约）的博士学位。他目前是南京大学（中国南京）电子科学与工程学院的正教授。2011年至2014年期间，他先后任职于三星美国研究院（美国得克萨斯州达拉斯）和未来科技有限公司（美国加利福尼亚州圣克拉拉）。他的研究研究兴趣包括学习型图像/视频编码和计算成像。他曾获得2018年PCM最佳论文入围奖、2019年IEEE广播技术学会最佳论文奖以及2020年IEEE多媒体信号处理大挑战最佳图像编码方案。