基于强化学习的实时视频流控与移动终端训练方法研究

张欢欢，周安福，马华东
（北京邮电大学智能通信软件与多媒体北京市重点实验室，北京 100876）

摘　要：以物联网、移动互联网为核心的服务平台加速发展，数以亿计的终端用户通过实时视频进行通信，实时视频已成为人们数字化生活中不可替代的核心工具。然而，互联网络呈现高动态、强异构的特性，对实时视频的流控技术提出了严格要求，用户体验质量仍然不佳。设计了适用于异构网络环境的强化学习驱动的自适应流控算法、研发了移动终端训练技术以降低服务端开销，并对算法的设计及结构进行了深入的评测研究。实验表明，所设计的自适应流控算法可以有效地预测网络带宽，相较于国际代表性的流控算法，将预测带宽误差降低了 48.48%。有效的带宽预测进一步提升了视频用户体验质量，如视频流畅度提升了 60.65%、视频清晰度提升了 16.52%。此外，测评分析可为实时视频流优化方案提供经验性指导，有力推动智能视频应用的发展。

关键词：实时视频；自适应流控；体验质量；强化学习；终端训练
中图分类号：TP393
文献标志码：A
doi: 10.11959/j.issn.2096–3750.2022.00306

Reinforcement learning-based real-time video streaming control and on-device training research

ZHANG Huanhuan, ZHOU Anfu, MA Huadong
Beijing University of Posts and Telecommunications, Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing 100876, China

Abstract: Service platforms centered on the Internet of things and mobile Internet are in accelerating process. Hundreds of millions of end-users communicate through network real-time video services, which have become an irreplaceable core tool in human's digital life. However, the Internet is becoming dynamic, and heterogeneous, which imposes stringent requirements on real-time video streaming control technology. Moreover, the QoE of real-time video is not satisfactory. An adaptive reinforcement learning-based video intelligent transmission algorithm was designed, which can deal with heterogeneous network environment. And then, an effective end-to-end on-device training framework was designed to decrease server overhead, and a detailed evaluation and analysis on the neural network design and structure was provided. Experimental results show that the proposed algorithm can effectively predict heterogeneous network bandwidth, and reduces the bandwidth prediction error by 48.48%, comparing with the representative streaming control algorithm. The effective bandwidth prediction can further improve the user QoE, such as improving the video fluency by 60.65%, and improving the video quality by 16.52%. Besides, the analysis can provide empirical insights for further study, and holds potential to push the development of intelligent video applications.

Key words: real-time video, adaptive streaming control, quality-of-experience, reinforcement learning, on-device training

---

收稿日期：2022–06–16；**修回日期：**2022–10–17
通信作者：张欢欢，zhanghuanhuan@bupt.edu.cn
基金项目：国家自然科学基金资助项目（No.61921003）；博士后创新人才支持计划（No.BX20220046）
Foundation Items: The National Natural Science Foundation of China (No.61921003), The China National Postdoctoral Program for Innovative Talents (No.BX20220046)

0 引言

以物联网、移动互联网为核心的服务平台加速发展，数以亿计的终端用户通过视频业务进行通信。网络视频业务逐渐由以视频网站、视频点播为代表的传统流媒体业务向视频直播、视频会议等新型实时视频业务扩展。不同于预先制作完成的传统流媒体业务，新型实时视频业务更强调将智能端设备产生的实时视频推流至服务云，并为终端用户提供实时交互、低时延的视频服务。思科发布的《可视化网络指数报告》[1]中指出，2022 年视频流将占据全球互联网流量的 82%以上。新型冠状病毒肺炎疫情在全球范围内的爆发，进一步推动了新型实时视频业务的发展。长期来看，新一代视频创新应用逐渐涌现[2-3]，如虚拟现实、全息体积视频、元宇宙等，新型视频网业务规模将会空前庞大。然而，动态网络环境对实时视频的用户体验质量（QoE, quality-of-experience）保障造成了严重的困难，如何设计具有网络自适应能力的实时视频流控算法，从而提升视频用户体验质量，是视频传输的关键。

实时视频传输流程如图 1 所示，视频内容经由发送终端（如手机、计算机、摄像机等）实时生成，进行本地视频的编码、视频帧生成，发送端将视频帧以数据包的形式源源不断地传输到终端用户，传输速率由发送码率进行实时调控。实时视频传输 QoE 的关键取决于发送端流控算法性能，其控制机理为根据当前网络的可用带宽实时调整数据发送速率，通过合理的码率调控策略，尽力适配高度动态的网络环境，避免网络拥塞的发生，以最大化利用网络资源、减少传输时延。然而，实时视频传输由于具有低时延、高清晰度与强交互的特性，所以在带宽和时延方面对网络提出了严苛的要求。由于互联网络长期处于复杂多变的状态，一旦网络的负载超过了其承受能力，则会产生网络拥塞，继而影响上层应用的 QoE 性能。因此，需要对现有的视频流控算法进行深入研究。

传统的流控方案大多使用基于固定映射规则的流控算法，如 CUBIC[4]、GCC[5]、BBR[6]等，已经无法满足当前视频流量下的高动态性和异构性需求。近年来，研究者利用机器学习技术优化视频传输中的流控难题。特别地，基于强化学习（RL, reinforcement learning）算法的流控设计是当前学术界研究和工业界实践的一个热点方向，如 Pensieve[7]算法、Concerto[8]算法、OnRL[9]算法。根据对上述前沿技术的深入分析，本文发现当前的机器学习驱动的算法主要可分为以下两种训练方式。

1) 基于视频仿真器/模拟器的离线训练方式：所训练的离线模型可直接部署于实际视频系统，作为固化模型直接进行模型的推演计算。代表性算法包括 Pensieve[7]算法、Concerto[8]算法。

2) 基于视频真实环境的在线训练方式：这种训练方式大多基于一种“云端–客户端”通信的模式。即模型在云端进行训练，并与视频客户端实时通信，模型可以根据网络信息进行实时训练与更新，继而推演为一个可对当前网络环境做即时反馈的视频传输控制算法。代表性算法有 OnRL[9]算法、Puffer[10]算法。

然而，本文发现，上述两种训练方式在动态网络系统实际应用中仍然存在不足。例如，离线训练方式由于模型过于依赖仿真器环境，无法全面适用于真实系统的异构环境，而且当前算法过度依赖所设置的视频码率阈值，一个算法无法同时满足多种视频清晰度要求。就在线训练方式而言，当前主流的“云端–客户端”通信模式虽然可以实现模型根据网络实时变化做即时推演，但是此种云端训练方式会消耗大量的云端资源，给视频服务商造成较大的经济损失。

本文利用强化学习的自适应潜力优化视频流控算法，设计了基于网络环境感知的动态码率阈值调控算法，还设计了一种基于终端设备（如苹果手机、安卓手机、平板计算机等）进行实时训练的框

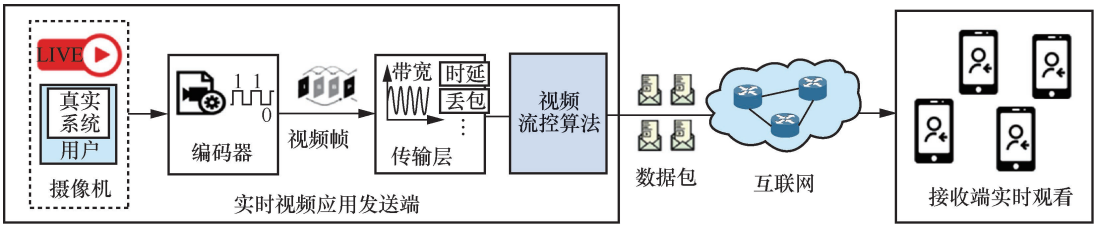


图 1　实时视频传输流程

架，继而推进智能算法下沉，节约服务器的开销。同时，本文具有如下 3 个方面的贡献。

1) 基于真实系统的视频数据集及实时视频传输系统，实现具有动态网络自适应能力的智能流控算法，使算法能在不同的网络条件下较准确地预估网络带宽。为了实现视频流控在动态网络中的自适应，本文根据用户感知网络环境设计了动态阈值调控算法，所设计的算法可以更好地提升动态网络环境中的视频 QoE，提升了视频用户体验质量，如降低视频卡顿率 60.65%，提升视频清晰度 16.52%。

2) 设计并实现一个可基于终端设备进行在线训练的强化学习框架，尤其使其可以适配于不同内核的终端设备并且可以支持强化学习模型的端上在线训练任务。此端上训练平台可以降低服务器的开销，节约服务器的成本，降低云端传输的时延。基于此，所设计的强化学习驱动的流控算法有能力更快地对网络即时变化做出码率适应性反馈，进而更好地服务于用户体验质量。

3) 由于强化学习本身的“黑盒”属性，研究者通常无法深入理解其自适应原理，继而对其中重要的参数调整存在困惑。本文通过深入探索基于强化学习的流控算法的主要超参数对带宽预测的影响，如神经网络结构、神经网络类型、激活函数类型等，力争通过严格的对比实验得出有效的参数配置以及影响较小的参数指标，以期为后续的视频流优化提供经验性建议。

1　研究现状

国内外的相关工作致力于研究视频传输算法，如传输层的码率调控（拥塞控制）算法以及应用层的视频码率自适应算法。视频流控算法的设计与优化存在两种思路：基于固定规则的传统算法和基于学习的创新算法。本节将首先对上述两种设计思路进行介绍，并进一步介绍目前主流的视频点播与实时视频的区别以及算法差异。

1.1　基于规则的协议设计

传统的视频流控解决方案大多使用基于固定规则的策略，例如代表性的 TCP，以及 TCP Reno[11]、Vegas[12]、CUBIC 等。据本文所知，TCP 的传输机制适用于文件传输，其 3 次握手建连、慢启动、有效重传等机制可以保证传输的可靠性。但是 TCP 由于传输过于复杂，已不再适用于具有高度动态、流量不可控的实时视频传输机制。基于此，许多基于用户数据报协议（UDP, user datagram protocol）的视频传输协议逐渐兴起如，PCC[13]、PCC-Vivace[14]、PROTEUS[15]、GCC[16]、Salsify[17]等。然而，虽然基于规则协议的算法可以较好地平衡时延与接收端吞吐量之间的关系，但它们大多依赖一组预先定义的控制机制，如根据网络的往返时延、丢包率、吞吐量等指标信号对网络进行固定映射反馈。大量实验已经表明，固定规则的策略无法处理当代复杂网络环境中不断增加的网络异构性和动态性，特别是对于具有较大的带宽和时延标准的移动网络环境（如 4G/LET 蜂窝网络、Wi-Fi 连接、兴起的 5G 大带宽网络等）。

1.2　机器学习驱动的协议设计

基于机器学习的传输控制算法是当前学术界研究和工业界实践的热点。最流行的是利用强化学习技术进行策略调整。近几年的研究已验证了使用强化学习策略优化网络协议和应用体验的优势。如 2013 年，Remy 算法[18]使用马尔可夫模型来优化拥塞控制算法。为了生成基于规则的控制器，Remy 算法需要预先定义一系列关于网络环境的超参数。在实际使用中，如果真实的网络偏离了原始的训练条件，Remy 算法的网络预测性能就会下降。2017 年，Pensieve 算法利用强化学习算法预测适应网络条件的最优视频码率。不同于 Remy 算法，Pensieve 算法不预先规定任何决策规则，而是根据强化学习的策略使其自适应学习，且在实验平台上取得了用户性能提升，掀起了自适应学习算法的热潮。继 Pensieve 算法之后，研究者们不断提出自适应的流控算法，如 2018 年，Indigo[19]算法使用数据驱动的方案优化模拟器中的流控协议；2019 年，Concerto 算法使用深度模仿学习（一种有监督学习策略）动态调控传输层和应用层的协调码率；2020 年，OnRL 算法利用在线学习（训练）方案解决真实视频电话系统的独特挑战。上述方案表明，基于机器学习的网络控制算法较传统基于规则的算法具有更好的 QoE 性能优势。本文专注于基于机器学习，特别是强化学习算法中实时视频传输控制算法的研究，且对影响强化学习性能的重要参数做了细致性测评。

1.3　视频点播与实时视频系统

如今，视频传输系统可根据视频的实时性分为以下两种。

1) 预生成好、可根据用户意愿随时观看的传统

流媒体视频，如点播视频（如多种视频点播平台：腾讯视频、爱奇艺、优酷等，以及抖音、快手等短视频）。

2) 实时生成、实时观看的视频（如视频电话、视频直播、视频会议等应用）。

点播视频与实时视频特性对比见表 1。具体来讲，主流的视频点播应用大都基于 HTTP 的 DASH 系统，其具体播放流程如下：预先生成好的视频会存储在应用服务商的 Web 服务器中。Web 服务器会预先将视频分为不同码率编码的视频块（Video Chunk，视频块长度通常为 2～10 s 不等），以及描述这些视频块及其码率的 manifest 文件。通常来讲，越高的码率视频块可以提供越好的视频清晰度，从而带来更佳的用户体验质量，但同时也会引发更大的视频块的传输。DASH 系统中会有适配的码率控制算法来预测网络带宽，继而下载当前带宽能承受的最佳质量的视频块，从而为用户提供最好的用户体验质量。主流的服务于视频点播的流控算法包含 BBA[20]、BOLA[21]、Festive[22]等。

实时视频的传输框架主要基于 WebRTC 系统，它是一个基于 UDP 上层协议的实时通信框架。实时视频的特质为视频由用户实时生成，并且需要在极短的时间内（通常为毫秒级别）被观众侧观看，以此达到时延无感的实时视频体验。由此可知，实时视频相较视频点播而言，具有更高的时延要求，因此其对视频传输算法的要求也更高。实时视频中较经典的传输算法包括 GCC、BBR、Proteus、Concerto、OnRL、Legato[23]等。本文关注更具有挑战性的实时视频传输应用的流控传输和视频质量的增强研究。

2　问题建模

2.1　强化学习概述

强化学习的基本理论模型为马尔可夫决策过程[24-25]（MDP, Markov decision process），属于序列决策问题。通常来讲，MDP 由一个四元组<状态 S, 动作 A, 奖励函数 R, 状态转移函数 T>表示，强化学习状态机转换示意图如图 2 所示，其中，S 表示强化学习的智能体在某一环境下的状态空间，特别地，在强化学习中的时间序列 $t = \{0,1,2,\cdots,n\}$ 中，在任一时间 t，下一个状态 S_{t+1} 仅与当前状态 S_t 有关，而与过往状态无关。任一时间状态的 S_t 都会与环境进行交互，产生一个属于动作空间 A 的动作 A_t。继而，强化学习智能体将会转移到下一个状态 S_{t+1}，并生成此阶段的奖励函数值 R。状态转移函数 T 可表示为$<S_t, A_t, S_{t+1}>\rightarrow T$。强化学习的最终目标为在过去的经验中学习最优的动作，以期最大化累积奖励回报。

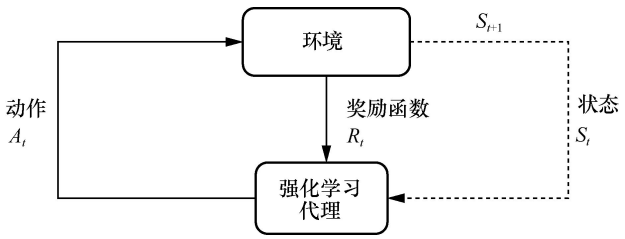


图 2　强化学习状态机转换示意图

2.2　基于强化学习的视频流控问题定义与设计

为了提升异构网络中的流控算法自适应性，以提高视频用户体验质量，本文将待解决的视频流控优化问题定义为强化学习模型，通过设置合理的网络环境、状态空间、动作空间、奖励函数，以及有效的学习策略，使模型通过大规模的数据训练，可以学习到真实的动态网络环境的变化规律，在训练环境和测试环境中能做出精准的带宽预测，继而提升用户质量（如视频卡顿率下降，视频清晰度提升等体验）。接下来，本文从状态空间、奖励函数设计，以及动作空间 3 个方面进行具体阐述。

2.2.1　状态空间

本文将状态空间定义为传输层的各项指标信息[26]，包括丢包率（loss rate，记为 l）、包时延（delay，记为 d）、包抖动时延（jitter delay，记为 j），以及实际编码速率（encoder bitrate，记为 b）。特别地，传输层协议通常以毫秒级的粒度预测传输码率，而应用层的编解码器以更粗的秒级别粒度改变视频

表 1　　**点播视频与实时视频特性对比**

性质	指标	传统流媒体视频，如视频点播	新型实时视频，如视频直播
传输特性	视频内容	提前生成，存储在资源服务器	实时生成、实时观看
	传输级别	块级别传输	数据帧/包级别细粒度传输
	缓存大小	2～10 s	100～300 ms
视频用户体验要求	卡顿要求	较低	较高
	视频画质要求	较高	相对较低

码率，所以本文为适配应用层和传输层的码率更新，综合选取 1 s 级别粒度的数据作为强化学习模型的学习状态。具体来讲，本文将 1 s 内的实时传输控制协议（RTCP, real-time transport control protocol）数据包作为历史数据信息，表示为 $S=\{l_i, d_i, j_i, b_i\}$。其中，每一个指标包含约 30 个数据的实时传输协议（RTP, real-time transport protocol）数据，所以输入的维度可近似约束为 shape =[30,4]。

此外，为了将各个指标约束到相同数据级别，本文将所有指标首先进行数据归一化处理，然后将其作为模型的输入，以避免异常值的干扰，影响神经网络的正常训练以及梯度更新。现举例归一化方式的具体实现，以丢包率为例，假设其序列为 $l=\{l_1, l_2, l_3, l_{\max}, \cdots, l_n\}$，则本文的归一化方式为 $l\_n=\{l_1/l_{\max}, l_2/l_{\max}, l_3/l_{\max}, \cdots, l_n/l_{\max}\}$。其他指标同理。所以，所有指标经过数据归一化后的状态输入为 $S_n=\{l_n, d_n, j_n, b_n\}$。

2.2.2　奖励函数设计

本文参考前沿的 Pensieve 算法与 OnRL 算法的奖励函数机制，设计了基于实时视频优化的奖励函数。具体来讲，此奖励函数主要涉及应用层领域的 3 个重要指标：视频码率（记为 V）、时延（记为 d），以及视频平滑度。视频平滑度的含义为连续的两个视频实际编码速率 b 之差的绝对值，视频平滑度值越低，代表视频的质量波动越小，用户体验越好。上述 3 种指标通过合理的超参数调整，如对视频码率及平滑度给予正向奖励（奖励因子），对视频缓冲时延给予负向惩罚（惩罚因子），然后进行加权综合，强化学习的奖励函数为

$$R = w_1 \cdot \sum_{i=1}^{N} V_i - w_2 \cdot \sum_{i=1}^{N} d_i + w_3 \cdot \sum_{i=1}^{N} |b_i - b_{(i-1)}| \quad (1)$$

其中，N 表示一个输入状态的 RTP 包的长度，$w_1>0$，$w_2>0$，$w_3>0$。本文在视频仿真系统上经过多种参数的组合与对比，设定一组最优组合作为实际使用，分别为：w_1 的值设置为 20，w_2 的值设置为 5，w_3 的设置值为 3。

2.2.3　动作空间

本文将连续动作空间定义为传输层的码率预测值，输出范围为 $A=[a\_\min, a\_\max]$，具体值为此空间内的任一离散数值。将状态空间输入给神经网络模型，算法会根据当前策略生成较优的码率动作值 a，此动作值会传输到发送端作为下一阶段的视频发送速率，继而模型会生成此次反馈的奖励函数值，用来判断上一次动作的收益情况，如若收益值较大，模型将会倾向于生成此类较优策略；如若收益值较小或者为负值，模型的参数训练则会避免生成类似动作。上述阶段循环往复，在训练一定规模的数据集并拥有较多的经验之后，则会演变为一个能够合理预测网络带宽的码率预测模型。

3　关键技术及模型设计

3.1　实时视频传输概述及设计

本文所设计的智能流控算法，利用强化学习算法本身的探索与开发原理、适当的反馈机制，与视频传输领域的网络固有属性进行深度结合，可以有效地达到自适应预测网络带宽的目的。具体而言，所设计的算法以及优势可以概括为以下两点。

1) 将视频传输领域的网络知识与强化学习算法进行深度结合，将拟优化的视频传输问题转化为强化学习模型。本文基于真实环境的视频传输框架，通过视频系统的网络反馈信息，设计有效的强化学习模块，如状态空间、动作空间、奖励函数，以及高效的学习策略，使模型通过大规模的数据训练，可以学习到真实网络环境的规律。继而在真实的网络环境中能够自适应地预测带宽，提升用户体验质量。

2) 考虑到视频应用的用户多样性以及网络环境异构性，本文改变之前算法的固定码率阈值设计思路。根据用户的网络环境感知设计了动态阈值调控算法，所设计的算法可以根据网络环境进行自适应阈值的调整，从而可以更好地满足用户多样性下的体验质量。

3.2　基于网络环境感知的动态码率阈值调控设计

本文发现，不同的用户网络环境具有异构性、强动态性特征[27-28]，包括 Wi-Fi、光纤、3G/4G/5G 蜂窝网络、卫星网络、跨大陆光纤链接等，每种网络都具有多样化的链路带宽、时延特性和缓存能力。此外，错综复杂的网络环境使网络带宽呈现高度动态性，通常在极短的时间内进行变化，如秒级别，而且下一时刻的网络带宽通常难以预测。还存在设备多样性（如苹果手机/不同厂商的安卓手机/计算机等）、偏好多样性（如直播中通常对时延的要求更高、点播中通常对清晰度的要求更高）等特征。在此情况下，如果所有的用户使用一种码率带宽上下限，即固定的动作空间，则无法使所有的用户体验都达到最优。例如，如果具有较好的网络环

境的用户（如平均瓶颈带宽大于 5 Mbit/s），它的 a_max 设置为 2 Mbit/s，则会造成未充分利用网络资源的现象，继而导致视频清晰度下降，这是限制强化学习自适应算法性能的瓶颈问题。

基于此，本文设计了一种基于用户网络感知的动态码率阈值算法。本文将网络分为 3 种类型：弱网、中网以及强网。根据对 3 种网络的判断，设计动态阈值调整算法，基于网络环境感知的动态阈值设计流程见算法 1。具体而言，在开启视频进行训练的初始阶段，在客户端与服务器端建立连接的过程中，通过网络的往返延迟（RTT, round trip time）与往返时间（t），估计客户端用户当前的网络带宽 b_avg，并判断其与网络带宽经验值 b_min、b_max 的关系，继而决定网络的类型。不同的网络类型对应着不同的动作空间，具体上下限会根据分析真实网络环境的分析而设定。

算法 1　基于网络环境感知的动态阈值设计流程

输入　网络的往返延迟（RTT）与往返时间（t），当前带宽变化均值 b_avg

输出　码率动作空间范围

视频初始阶段，定时估计当前网络状态，如当前带宽变化均值 b_avg

判断 b_avg 与 b_min、b_max 的关系

If ($b\_avg \leqslant b\_min$)：//如果判断为弱网

Then：$a\_min = 0.5$，$a\_max = 2$；A = [0.5 Mbit/s, 2 Mbit/s]；

If ($b\_min < b\_avg < b\_max$)：　//如果判断为中网

Then：$a\_min = 1.0$，$a\_max = 2.5$；A = [1.0 Mbit/s, 2.5 Mbit/s]；

If ($b\_avg >= b\_max$)：　//如果判断为强网

Then：$a\_min = 2.0$，$a\_max = 5.0$；A = [2.0 Mbit/s, 5.0 Mbit/s]

按照不同网络状态的动态阈值训练不同网络模型，分配给用户使用

模型更新及保存

3.3　训练策略

本文优化目标为最小化强化学习算法预测码率与真实带宽的绝对值之差，目标函数 T 可表示为

$$T = \mathrm{Min}(|\, b - A(\arg\max(L(p)))\,|) \tag{2}$$

其中，b 表示网络即时带宽，$L(P)$为强化学习模型的最后一层神经网络预测概率矩阵，A 为强化学习模型的动作空间。神经网络训练的目标为随着训练时长的增加，期望目标函数 T 的值可以越来越低。在此过程中本文用全连接神经网络模拟算法的策略 π，使用具有强大特征处理能力的深度神经网络进行特征的提取及学习，并采用了强化学习中经典的 PPO（proximal policy optimization）[29]算法作为模型的更新策略。PPO 是一种新型的基于策略的梯度更新算法，主要解决强化学习模型的策略更新中，步长过大导致新旧策略变化差异过大而引发的模型难以收敛问题。PPO 算法的设计思想是通过控制新旧策略的差异不要过大，从而降低采样难度，提升算法收敛效率。该策略适用于本文的视频流控算法，这是因为平缓的策略更新对应着视频码率的相对平滑性，有益于提升视频的流畅性。具体训练过程在第 4.2 节阐述。

3.4　训练方法及架构

系统方法及架构示意图如图 3 所示。具体来讲，本文使用 Concerto 与 OnRL 的实时视频数据集进行模型的训练，该数据集包含了真实视频系统的大规模视频会话，容纳了不同的地区以及网络类型，是较具有代表性的数据集。将此数据集进行串联拼接，输入视频传输系统。所设计的强化学习流控机制内嵌在视频传输系统，可以根据预先收集好的网

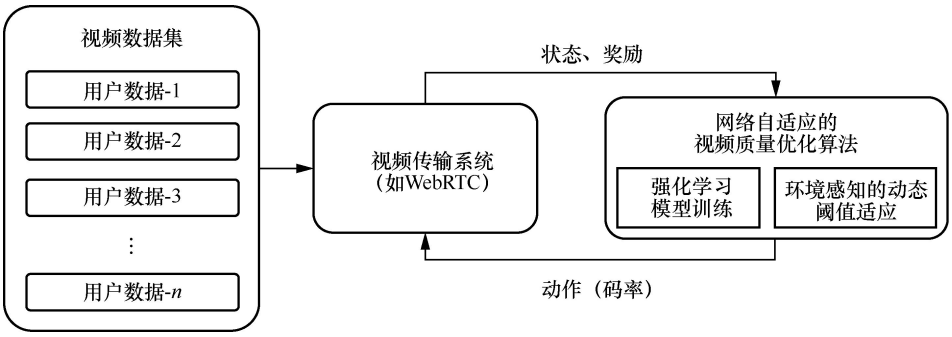


图 3　系统方法及架构示意图

络数据集，复现出异构的网络环境，以供强化学习模型实时与环境互动，进行探索与学习。此外，为使视频传输具有在真实互联网环境中的传输能力，本文的视频传输系统包含全面的功能模块，如发送端的视频编码、拆分视频帧、基于强化学习算法的传输控制算法，接收端的视频解码、视频渲染及观看反馈等（流程及模块将会在第 5 节做详细介绍）。

3.5　参数设计

神经网络中的重要参数配置最优组合为多次实验所得，如神经网络结构类型、神经网络层数、状态融合机制、激活函数类型、批处理大小等。算法中主要的参数配置见表 2，在本文的第 5.4 节验证主要参数配置的有效性。

表 2　　算法主要的参数配置

英文名称	中文描述	设定最优值
Network architecture	神经网络结构类型	全连接层
Fusion manner	输入状态融合方式	先融合
Activation function	激活函数	LeakyReLu
Batch size	批处理大小	32

4　基于强化学习的移动终端训练设计与方法

近年来，在移动终端进行深度学习模型的再训练，已经获得了学术界以及工业界的关注，但在视频传输方面还未有进展。本节将率先总结近年来较流行的机器学习框架平台（第 4.1 节），并结合主流的端上训练平台，设计并实现一个可支持移动终端训练的视频智能传输控制算法（第 4.2 节），实现了学习驱动流控算法部署新渠道，有潜力扩展到真实的、应用级的部署。

4.1　端上推理与训练平台汇总

本节介绍了工业界开发的使用较广的机器学习端上推理与训练平台，并介绍平台的主要功能以及优缺点，主流的支持移动端机器学习的框架见表 3，可以看出，目前的主要框架包含 Core-ML 、 ML-Kit 、 Paddle-mobile 、 Caffe2 、 TensorFlow Lite、TensorFlow.js，以及 MNN 7 个平台。通过本文的深入对比和分析，可初步得出结论：当前大多数的框架只支持端上的推理，大多不支持端上的训练任务，尤其是针对强化学习平台；目前发现，只有谷歌公司开源的 TensorFlow.js 支持苹果和安卓移动终端的强化学习训练任务，所以本文试图将第 2 节和第 3 节所述的强化学习算法迁移至 TensorFlow.js 框架并部署于移动终端，接下来在第 4.2 节介绍其具体实现。

4.2　基于端上训练的视频智能流控算法

本文提出了基于 TensorFlow.js 的视频智能传输控制算法，该算法可轻量地部署于移动终端，并支持移动终端的实时训练任务。基于端上训练的策略可以大幅度的节省之前算法（如 OnRL）基于云服务训练的资源占用，并可减少客户端与服务端的传输时延，继而可使算法更快地对当前网络变化做出反馈。

本文将基于 Python 语言的 PPO 算法重构为 TensoFlow.js，并设计了有效的 actor-critic 框架、训练策略以及损失函数。训练相关主要模块的网络拓扑如图 4 所示、神经网络结构如图 5 所示，主要功能模块为神经网络结构、actor 策略、critic 策略以及优势函数设计。特别地，actor 与 critic 网络结构都使用了 4 层级联的全连接层进行特征提取，不同

表 3　　主流的支持移动端机器学习的框架

框架名称	来源	发布年份	主要功能	特性
Core-ML	苹果公司	2017 年	使用简单；只支持端上推理，经常用于有监督习任务	较适用于苹果设备，只支持端上推理；在安卓设备的性能未知
ML-Kit	谷歌公司	2018 年	同时支持苹果和安卓设备，并且可以在两个平台上使用相同的 API	具有 6 个基本 API，易于实现，但只支持有监督学习任务，不支持强化学习的训练
Paddle-mobile	百度公司	2019 年	部署灵活，支持多硬件	只支持模型的端上推理，不支持模型的端上训练
Caffe2	Facebook 公司	2017 年	同时覆盖训练和推理的通用框架；支持云端深度神经网络的训练	只支持模型的端上推理，不支持模型的端上训练
TensorFlow Lite	谷歌公司	2018 年	TensorFlow 在移动终端上运行深度学习算法的平台；内存占用较低	只支持模型的端上轻量级推理，不支持模型的端上训练
TensorFlow.js	谷歌公司	2018 年	TensorFlow 的 JS 平台；灵活，可较好的与 Web 交互；同时支持苹果和安卓设备；支持强化学习的端上训练任务	同时支持移动终端的推理与训练任务；需要将机器学习相关的低代码重构为 JS
MNN	阿里巴巴公司	2019 年	轻量级的深度学习端侧推理引擎；同时支持苹果和安卓设备	只支持模型的端上推理，不支持模型的端上训练

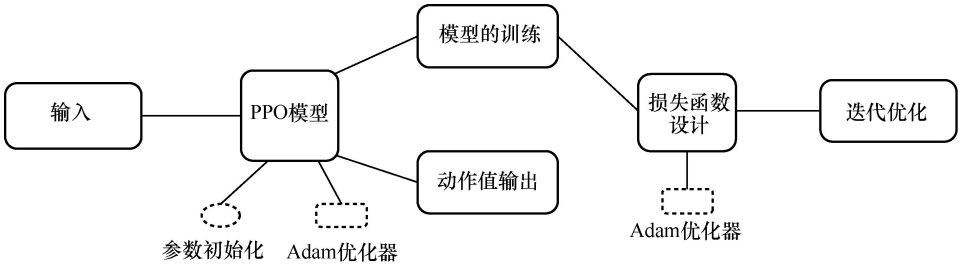


图 4　训练相关主要模块的网络拓扑

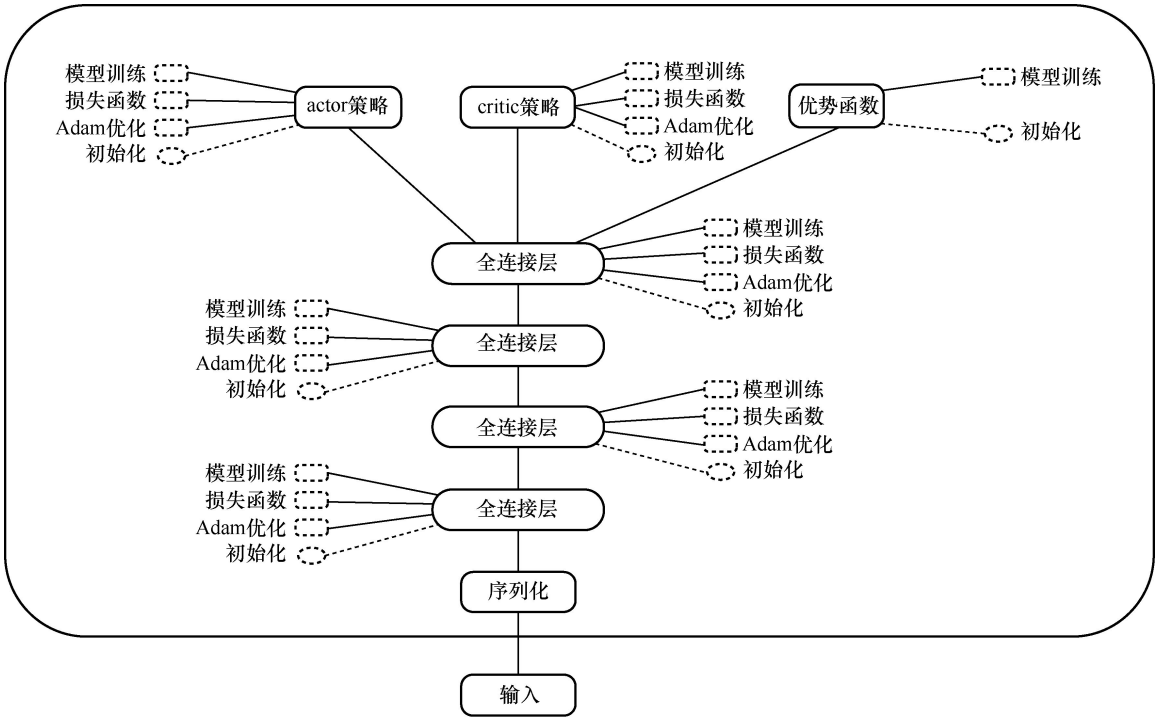


图 5　神经网络结构

的是，actor 网络输出码率决策（表示为 $Q^{\pi}(s,a)$），critic 网络输出 actor 决策的评价（表示为 $V^{\pi}(s)$），以供 actor 网络调整其策略。在 PPO 的训练过程中，优势函数是比较重要的一种训练策略，含义为当前动作决策相对于平均期望的优势，表示为

$$A^{\pi}(s,a)=Q^{\pi}(s,a)-V^{\pi}(s) \tag{3}$$

优势函数可以辅助基于策略的梯度更新算法，本文所采用的上述训练方式和迭代过程可以保证学习策略的有效更新。

5　实验结果与分析

本节使用 WebRTC 视频传输框架（包括 Linux 系统、安卓手机以及苹果手机的不用应用部署），评估所设计算法的在异构网络的实际效果。首先简述实验环境（第 5.1 节），继而设计全面的性能评估实验（第 5.2 节、第 5.3 节）。最终针对用户体验质量指标，设计对比实验，验证主要参数配置的有效性（第 5.4 节）。

5.1　数据集和实验设置

本文的视频传输框架基于 WebRTC 系统实现，实时视频传输主要模块设计如图 6 所示，包括视频发送端（主要模块：实时视频采集、视频编码、Pacer 机制以及基于强化学习的流控算法）、网络传输、视频接收端（主要模块：视频帧分析、视频解码、视频渲染以及视频播放）以及网络状态反馈模块。本文的神经网络实现基于主流的 TensorFlow[30]框架以及 TensorFlow.js[31]平台，运行系统为 Linux 16.04，GPU 型号为 Tesla P100、8 核 CPU。模型训练时间消耗 391 MB 内存。移动终端选择两种不同内核的手机，型号分别为 OPPO R17 以及 iphone XR。对比算法为目前在多个实时视频平台使用的内置算法 GCC。

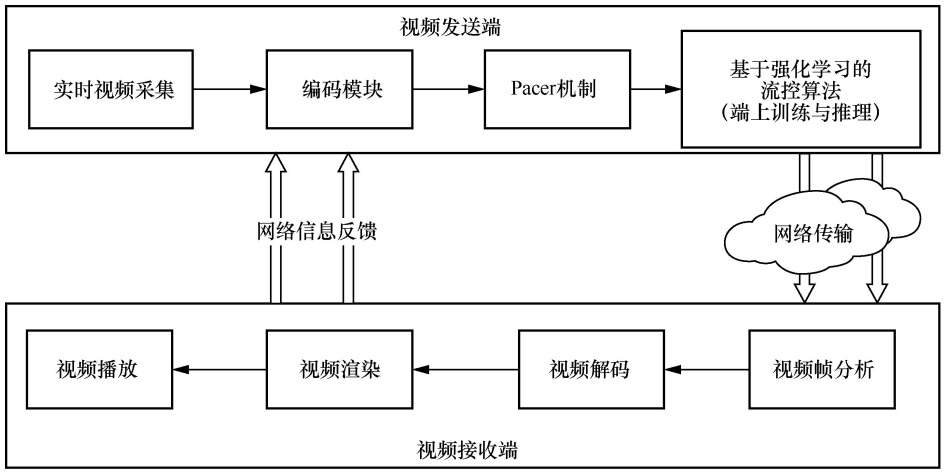


图 6　实时视频传输主要模块设计

5.2　用户体验质量评测

本节将所设计算法与现在实时视频中使用较多的 GCC 算法进行深入对比，并从视频流畅度与清晰度、带宽预测效果、本文算法与代表性的学习驱动流控算法对比、动态阈值策略效果、传输层性能解析 5 个方面深入展开讨论。

5.2.1　视频流畅度与清晰度对比

本文首先在测试平台上基于 WebRTC 框架进行受控实验。具体来讲，随机选择某一网络变化动态数据集（截取时长为 2 h 的数据作为测试数据），通过流量控制（TC, traffic control）工具复现该用户的网络变化情况。首先，选择 GCC 算法作为控制算法，统计该测试数据在基准算法 GCC 算法下的清晰度、流畅度等指标表现。其次，使用本文所设计的算法替换 GCC 算法进行视频传输。在同等条件下统计两种算法的视频流畅度及清晰度表现。实时视频流控算法比较见表 4，相比于 GCC 算法，本文所设计的 PPO 算法在视频流畅度方面，可降低卡顿率（提升视频流畅度）60.65%，在视频清晰度方面可提升 16.52%，说明本文算法能够较好地预测网络实时带宽，从而提升用户的流畅度及清晰度体验。

5.2.2　带宽预测效果

本文将训练好的模型与 GCC 算法在随机挑选的不同网络测试集上比较带宽、GCC 算法预测的码率以及 PPO 算法预测的码率。带宽、视频码率对比如图 7 所示（GCC 算法的预测带宽曲线（蓝线曲线）、PPO 算法的预测带宽性能（红线曲线）、网络的真实带宽 BW（黑线曲线））。通过分析可以得出如下结论。

表 4　　**实时视频流控算法比较**

主要参数	PPO 算法	GCC 算法	性能提升百分比
视频卡顿率	0.353%	0.897%	–60.650%
视频清晰度/(Mbit·s^{-1})	1.340	1.150	+16.520%
网络带宽估计误差/(Mbit·s^{-1})	0.152	0.297	–48.480%

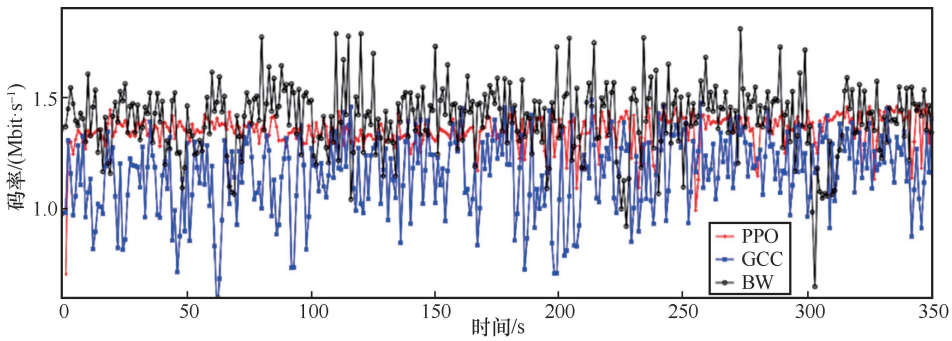


图 7　带宽、视频码率对比

1) 相比于 GCC 算法，本文所使用的 PPO 算法所预测的码率与带宽更贴合，具有更准确的带宽预测能力。

2) 具体而言，PPO 算法相比 GCC 算法，将预测带宽误差降低 48.48%，即设计的流控算法对网络的预测更准确，从而可以更实时准确的响应网络状况，提升用户体验质量。

5.2.3 本文算法与代表性的学习驱动流控算法对比

除了与基于固定规则的算法对比，本文还与代表性的学习驱动流控算法 Concerto、OnRL 进行了比较。Concerto 是一种基于仿真器训练的离线学习算法，它通过深度模型学习进行码率决策的学习；OnRL 是一种基于在线强化学习策略的视频智能流控算法，它通过将强化学习智能体直接部署到设备终端来完成实时训练与推演，无须经历“仿真器–实际系统”的跨平台推演。但是，上述两种智能流控算法的动作空间都是固定不变的，均为[0.1 Mbit/s, 2.5 Mbit/s]，码率空间的上限，它们无法高效地在高带宽网络环境中进行自适应调控，例如无法支持 1080P 清晰度所对应的 4～8 Mbit/s 码率空间。经过实验发现，在多种异构的网络环境中，相较于 Concerto、OnRL 算法，本文所设计的 PPO 算法可以将视频码率提升 34% 以上，且没有导致更大的时延，此结果验证了本文所设计的动态阈值策略的有效性。

5.2.4 动态阈值策略效果

本节验证基于用户网络感知的动态阈值设计策略的有效性。本文模拟两种简单的网络环境：网络正常变动以及网络带宽突然下降。在实验过程中发现，所设计算法感知到网络带宽发生了较大变化，将码率动作阈值进行了向下调整。基于环境感知的动态阈值算法验证如图 8 所示，基于 PPO 算法的码率决策可以快速地跟随网络带宽变化，进而避免了大丢包和高时延等 QoE 不佳现象的发生，继而验证了该算法可以有效地避免用户体验质量不佳的情况发生，上述结果验证了本文的动态阈值算法在异构网络环境中的有效性。

5.2.5 传输层性能解析

为了深入地理解端上训练的 PPO 算法的性能，本节随机挑选了一条 600 s 的真实网络带宽 trace，并在实验床进行网络复现，实时观测网络传输层最重要的两个性能指标：时延和丢包率，同一时刻下的 QoE 指标对比如图 9 所示，可以得出两个重要结论。

1) PPO 算法可以较好地跟随网络的波动，与网络带宽的实时波动拟合得较好，实验结果与图 7 相符，验证了算法表现的一致性。

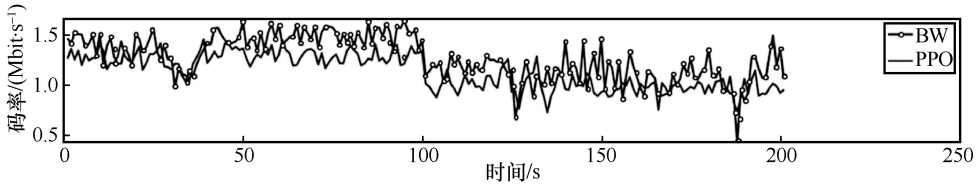


图 8　基于环境感知的动态阈值算法验证

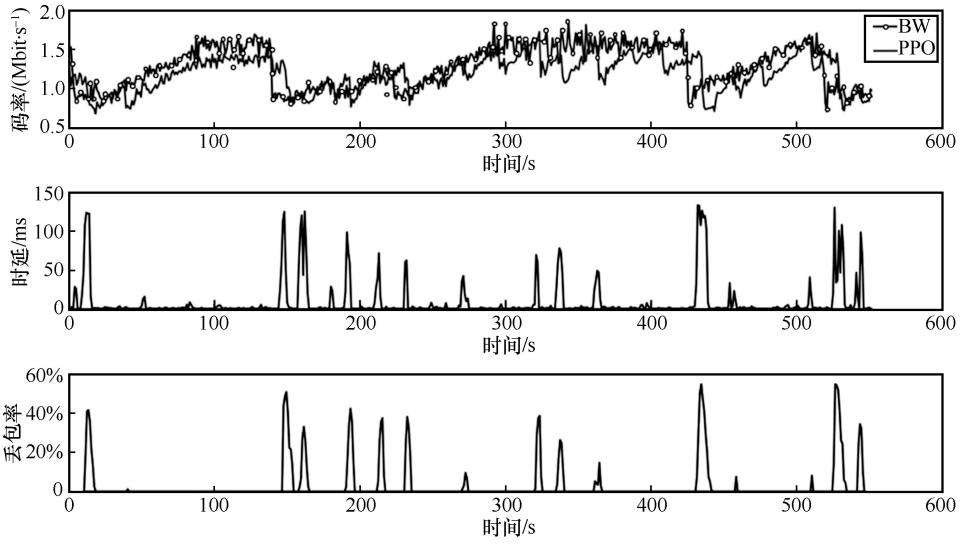


图 9　QoE 指标对比

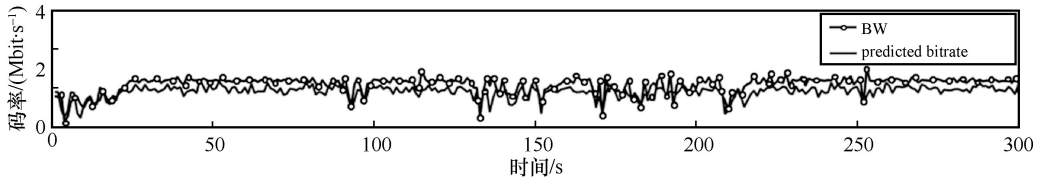
2) 测量发现，视频平均时延为 9.92 ms，平均丢包率为 3.34%，且最大时延值小于 150 ms，本文认为当前算法可以达到“零秒时延”的用户体验。

5.3　算法鲁棒性测评

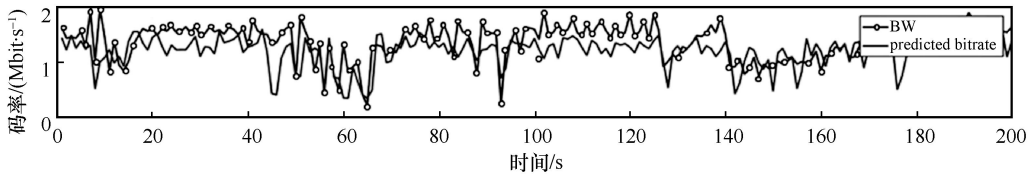
5.3.1　鲁棒性：在异构网络中的 QoE 性能表现

为了验证算法鲁棒性，本文在异构网络环境下进行了测试：测试集-1，代表较稳定的网络，真实带宽的波动性较小，如室内稳定的 Wi-Fi 环境；测试集-2，代表较动荡的网络，真实带宽在短时间粒度内经常出现较大变化，如室外不稳定的 4G 环境。

不同测试集中的 QoE 性能展示如图 10 所示，可以看出，无论网络情况如何，所设计算法可以较准确地响应瞬时变化的带宽，从一定程度验证其性能鲁棒性。



(a) 测试集-1 较稳定的网络变化



(b) 测试集-2 较动荡的网络变化

图 10　不同测试集中的 QoE 性能展示

5.3.2　训练有效性：损失函数下降趋势示意图

为了进一步验证训练算法的有效性，训练过程中的损失函数曲线如图 11 所示，可以得出结论：损失函数随着迭代次数的增加，呈现较稳定的下降趋势，有效验证了模型可以快速收敛并最终趋向稳定。

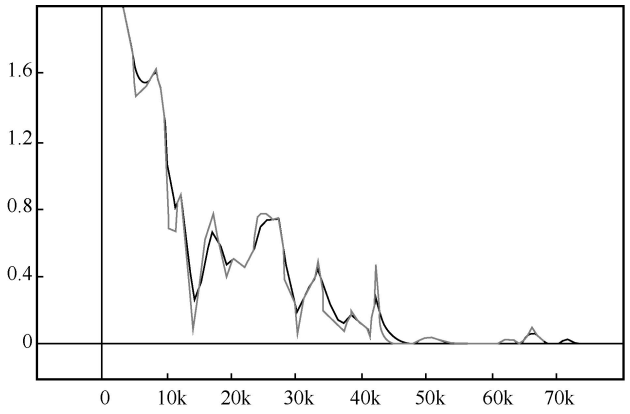


图 11　训练过程中的损失函数曲线

5.4　消融实验讨论

本节对影响模型的两个重要机制：状态机制融合方式以及激活函数展开深入讨论。

5.4.1　输入状态融合方式的影响

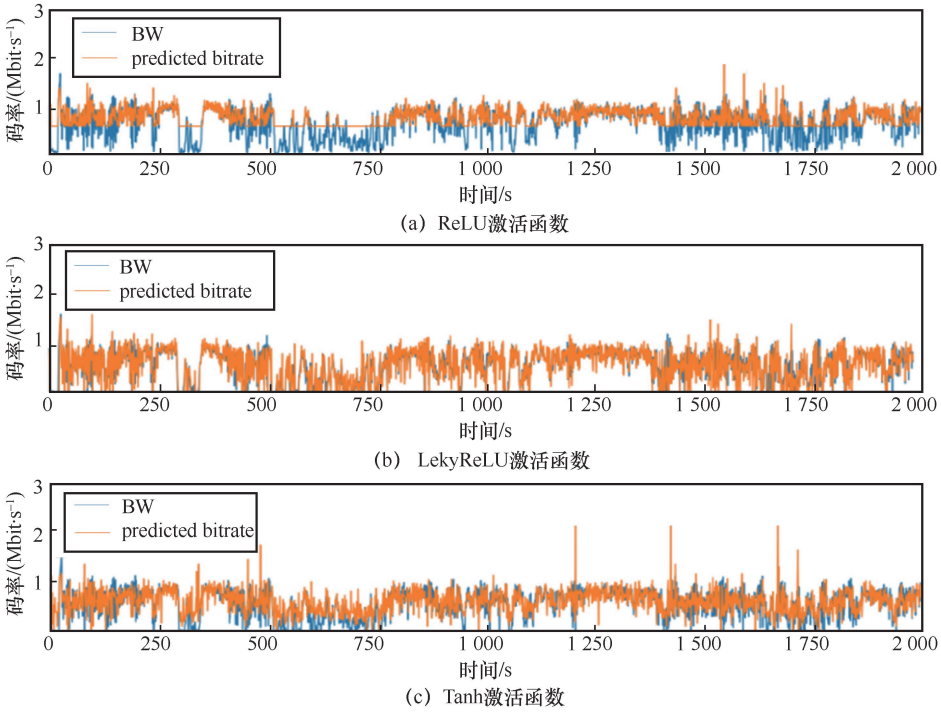
融合方式为深度学习领域中特征提取方面的一个经典问题。先融合表示先将输入状态的各个特征进行拼接，继而输入神经网络进行特征提取；后融合表示首先将状态的各个模块分别输入各自的神经网络，经过神经网络提取特征后，进行多个特征的后融合。本文使用相同的测试集测试了两种不同融合方式的效果，实验发现，相比于先融合机制，后融合机制能够更加准确地预估网络带宽。

5.4.2　激活函数的影响

激活函数将非线性概念引入到神经网络的训练过程中，可以降低网络的稀疏性，保证训练的有效进行。特别地，本实验专注于主流的 3 种激活函数：ReLU、LeakyReLU 及 Tanh 对带宽预测的影响，激活函数的影响如图 12 所示，可以看出，ReLU 激活函数的效果较差，推测原因为输入归一化的限制，神经网络计算过程中负值较多，而 ReLU 激活后大多数神经元被抑制，从而影响到特征提取效果。进一步地，实验发现 LeakyReLU 较 Tanh 激活函数具有更精准的网络带宽预测能力，所以本文使用 LeakyReLU 激活函数作为最终的实验配置。

5.4.3　神经网络结构的影响

特征提取的两种主要方式为全连接神经网络与卷积神经网络。本文对比了两种神经网络结构对智能流控算法的影响。在全连接结构中，多种输入，如丢包率、包时延、包抖动时延，以及实际编码速率，会先进行序列化处理，然后经过级联的全连接层进行语义信息提取，生成高层次语义特征，进行



(a) ReLU激活函数

(b) LekyReLU激活函数

(c) Tanh激活函数

图 12　激活函数的影响

码率决策；在卷积结构中，本文将 4 种不同的输入重构为三维矩阵，然后使它经过级联的 2 维卷积层进行特征提取。保持其他变量不变，分别使用上述两种结构进行特征提取，神经网络结构比较见表 5，两种结构导致的 QoE 相关参数差距不大，相较而言，全连接层具有更高的视频清晰度与平滑度，所以，本文使用全连接结构进行网络信息的特征提取。

表 5　神经网络结构比较

主要参数	全连接层	卷积层
视频延迟/ms	9.11	9.37
视频清晰度/(Mbit·s^{-1})	1.22	1.20
视频平滑度/(Mbit·s^{-1})	0.089	0.081

5.4.4　批处理大小的影响

批处理大小代表训练过程中一次性输入状态的个数，不同的批处理大小影响着神经网络训练过程中的训练速度、参数更新频率。本文分别进行了批处理大小为 32、64、128 的实验，从实验结果可以看出，对于本文所设计的 PPO 神经网络结构而言，不同批处理大小所呈现的 QoE 相似（如码率差异保持在 0.1 Mbit/s 以内），所以本文选择更小的批处理大小——32，它能够加速神经网络更新频率，使智能流控算法更快地响应网络动态变化，从而实现实时地自适应决策调整。

6　结束语

视频智能流控技术是物联网、移动互联网的热点研究问题，得到了学术界及工业界的密切关注，对智慧网络以及人工智能的发展具有重要推进作用。本文结合前沿技术，利用强化学习领域的动态特征提取机制、自适应学习模式以及基于用户网络环境感知的动态码率阈值调控算法，实现了准确探测网络实时带宽的算法，较传统算法有明显的性能提升。此外，本文设计了可支持移动终端训练的视频智能传输算法，可以有效节约云端服务开销。最后，本文对算法的鲁棒性及参数设计进行了细致的评测与分析。

参考文献：

[1] LUO J G, ZHANG M, ZHAO L, et al. A large-scale live video streaming system based on P2P networks[J]. Journal of Software, 2006, 18(2): 391-399.

[2] FENG D G, XU J, LAN X. Study on 5G mobile communication network security[J]. Journal of Software, 2018, 29(6): 1813-1825.

[3] Cisco visual networking index: forecast and trends[EB]. 2019.

[4] HA S, RHEE I, XU L. CUBIC: a new TCP-friendly high-speed TCP variant[J]. Operating Systems Review, 2008, 42(5): 64-74.

[5] CARLUCCI G, DE CICCOL, HOLMER S, et al. Congestion control for web real-time communication[J]. IEEE/ACM Transactions on Networking, 2017, 25(5): 2629-2642.

[6] NEAL C, YUCHUNG C, STEPHEN G, et al. BBR: congestion-based

congestion control[J]. Communications of the ACM, 2017, 60(2): 58-66.
[7] MAO H, NETRAVALI R, ALIZADEH M. Neural adaptive video streaming with pensieve[C]//ACM Special Interest Group on Data Communication (SIGCOMM) 2017. Los Angeles: ACM Press, 2017: 197-210.
[8] ZHOU A F, ZHANG H H, SU G Y, et al. Learning to coordinate video codec with transport protocol for mobile video telephony[C]//Proceedings of the 25th Annual International Conference on Mobile Computing and Networking (MobiCom) 2019. Los Cabos: [s.n], 2019: 21-25.
[9] ZHANG H H, ZHOU A F, LU J M, et al. OnRL: improving mobile video telephony via online reinforcement learning[C]//Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (MobiCom) 2020. London:[s.n], 2020: 1-14.
[10] YAN F Y, HUDSON A, ZHU C Z, et al. Learning in situ: a randomized experiment in video streaming[C]//Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Santa Clara:[s.n], 2020: 495-511.
[11] JACOBSON V. Congestion avoidance and control[C]//Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM). Stanford: [s.n], 1988: 314-329.
[12] BRAAKMO L S, O'MALLEY S W, PETERSON L L. TCP vegas: new techniques for congestion detection and avoidance[C]//Proceed ings of the ACM Special Interest Group on Data Communication (SIGCOMM). London: [s.n], 1994: 24–35.
[13] DONG M, LI Q, ZARCHY D, et al. PCC: re-architecting congestion control for consistent high performance[C]//Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Oakland:[s.n], 2015: 395-408.
[14] DONG M, MENG T, ZARCHY D, et al. PCC vivace: online-learning congestion control[C]//Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Renton: [s.n], 2018: 343-356.
[15] XU Q, MEHROTRA S, MAO Z M, et al. PROTEUS: network performance forecast for real-time, interactive mobile applications[C]// Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys). Taipei: [s.n], 2013: 347-360.
[16] Web RTC homepage[EB]. 2018.
[17] FOULADI S, EMMONS J, ORBAY E, et al. Salsify: low-latency network video through tighter integration between a video codec and a transport protocol[C]//Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation, (NSDI), Renton: [s.n], 2018: 267-282.
[18] WINSTEIN K, BALAKRISHNAN H. TCP ex-machina: computer-generated congestion control[C]//Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM). Hong Kong: [s.n], 2013: 123-134.
[19] FRANCIS YY, MA J, HILL G D, et al. Pantheon: the training ground for Internet congestion-control research[C]//Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC). Boston: [s.n], 2018: 731-743.
[20] HUANG T Y, JOHARI R, MCKEOWN N, et al. A buffer-based approach to rate adaptation: evidence from a large video streaming service[C]//Proceedings of the ACM Symposium on Communications Architectures and Protocols (SIGCOMM). [S.l.:s.n], 2014: 187-198.
[21] SPITERI K, URGAONKAR R, SIATRAMAN R K. BOLA: near-optimal bit rate adaptation for online videos[J]. IEEE/ACM Transactions on Networking, 2020, 28(4), 1698-1711.
[22] JIANG J C, SEKAR V, ZHANG H. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive[C]// Proceedings of IEEE/ACM Transactions on Networking. Piscataway: IEEE Press, 2012: 326-340.
[23] ZHANG H, ZHOU A, MA H. Improving mobile interactive video QoE via two-level online cooperative learning[J]. IEEE Transactions on Mobile Computing, 2022, Early Access.
[24] 刘克. 实用马尔可夫决策过程 [M]. 清华大学出版社, 2004.
LIU K. Applied Markov decision processes[M]. Beijing: Tsinghua University Press, 2004.
[25] 范长杰. 基于马尔可夫决策理论的规划问题的研究[D]. 中国科学技术大学, 2008.
FAN C J. Research on planning problem based on Markov decision theory[D]. Hefei: University of Science and Technology of China, 2008.
[26] ZHANG H, ZHOU A, MA R, et al. Arsenal: understanding learning-based wireless video transport via in-depth evaluation[J]. IEEE Transactions on Vehicular Technology, 2021, 70(10): 10832-10844.
[27] SUN Y, YIN X, JIANG J, et al. CS2P: improving video bitrate selection and adaptation with data-driven throughput prediction[C]//Pro ceedings of the ACM Special Interest Group on Data Communication (SIGCOMM). New York: ACM Press, 2016: 272-285.
[28] HU Y X, LI D, SUN P H, et al. Polymorphic smart network: an open, flexible and universal architecture for future heterogeneous networks[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(4): 2515-2525.
[29] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB]. 2017.
[30] Dillon J V, LANGMORE I, Tran D, et al. Tensorflow distributions[J]. arXiv preprint , 2017, arXiv:1711.10604.
[31] SMILKOV D, THORAT N, ASSOGBA Y, et al. Tensorflow.js: machine learning for the web and beyond[J]. Proceedings of Machine Learning and Systems, 2019: 309-321.

[作者简介]

张欢欢（1994– ），女，博士，北京邮电大学在站博士后，主要研究方向为物联网、移动计算、视频传输。

周安福（1981– ），男，博士，北京邮电大学教授，主要研究方向为物联网感知、毫米波、实时视频传输。

马华东（1964– ），男，博士，北京邮电大学教授，IEEE Fellow，主要研究方向为物联网与传感网、多媒体系统与网络。