arXiv:2304.04527v1 [cs.MM] 10 Apr 2023

采用重要性加权 A3C 的深度强化学习：提升视频交付服务的体验质量

Mandan Naresh、Paresh Saxena、Manik Gupta计算机科学与信息系统系，比尔拉理工学院与科学学院皮拉尼分校印度海得拉巴{p20180420, psaxena,manik}@hyderabad.bits-pilani.ac.in

摘要—**自适应比特率（ABR）算法根据网络状况调整视频比特率，以提升整体视频体验质量（QoE）。近年来，强化学习（RL）和异步优势演员-评论家（A3C）方法已被用于生成自适应比特率算法，与固定规则的自适应比特率算法相比，这些方法已被证明能够提升整体体验质量。然而，A3C 方法普遍存在一个问题：行为策略与目标策略之间存在滞后。因此，行为策略与目标策略不再同步，导致更新结果并非最优。本文提出 ALISA：一种采用重要性采样、用于自适应比特率算法高效学习的演员-学习器架构。ALISA 引入重要性采样权重，对相关经验赋予更高权重，以解决现有 A3C 方法中的滞后问题。本文介绍了 ALISA 的设计与实现，并将其性能与先进的视频速率自适应算法进行比较，其中包括在 Pensieve 框架中实现的原始 A3C，以及 BB、BOLA 和 RB 等其他固定规则调度器。结果表明，ALISA 的平均体验质量比 Pensieve 提高了 25%–48%；与固定规则调度器相比，提升幅度更大。**

索引词—**深度强化学习，视频传输，体验质量（QoE），自适应比特率（ABR），演员-评论家方法**

I. 引言

近年来，互联网连接设备的使用迅速增长，预计这一趋势未来将持续下去。 [1] 的作者报告称，2021年上半年，视频流传输占所有互联网流量的53.72%。此外，预计到 2023 [2], ，连接到 IP 网络的设备数量将达到全球人口的三倍，而基于 HTTP 的视频流传输将占网络流量的很大一部分。然而，多项研究 [3]表明，视频质量低下往往会导致用户中止视频会话，给内容提供商造成重大损失。人工智能有能力显著改善多种移动服务，包括视频流传输、在线游戏、IP 语音、智能家居应用和远程健康监测。它可用于优化用户视频流传输的体验质量（QoE）。基于 HTTP 的动态自适应流传输（DASH） [4] 已成为视频流传输的一项重要标准内容通过尽力而为的互联网传输。总体而言，人们已广泛研究自适应比特率（ABR）算法在提升基于 DASH 的视频流传输体验质量方面的潜力 [5]。ABR 算法会根据缓冲区占用量和观测到的吞吐量等网络状况，自动调整视频比特率，从而为终端用户带来更好的体验质量。然而，这些算法依据预先设定的一组标准做出决策，并且通常针对特定状况进行调整。因此，要将此类方法推广至当今不断变化的网络中各种各样的网络状况，十分困难。

强化学习（RL） [6] 是机器学习的一个子领域，研究智能体应如何在环境中采取行动，以最大化某种累积奖励。近期有多项研究探讨了将强化学习方法应用于视频流传输 [7]，[8] ，旨在实现较高的体验质量。采用异步优势演员-评论家（A3C）方法的强化学习技术 [9] 已展现出相较于基于固定规则的自适应比特率算法的诸多优势。一些研究人员 [7], [8]采用原始 A3C方法生成自适应比特率，以提高整体体验质量。A3C [10]智能体由多个演员和一个带有评论家的中央学习器组成。每个演员根据自身的行为策略，独立且并行地生成经验。随后，各个经验被传达给中央学习器，中央学习器根据生成的经验修改目标策略（即A3C智能体试图学习的策略）。然而，A3C智能体需要海量数据才能学习适当的策略。增加演员数量是快速处理大量数据的常用方法。然而，在这种情况下，每个演员的行为策略开始滞后于中央学习器的目标策略[11]。因此，行为策略与目标策略变得不同步，导致更新结果不是最优的。这可能导致在使用强化学习进行自适应比特率算法时，带宽利用效率低下，并使整体体验质量下降。

为了解决这一问题，我们在使用 A3C 方法生成自适应比特率时引入重要性采样权重 [11]

以提升视频流媒体服务的体验质量。在根据经验的相关性为其分配权重时，我们提出的方法解决了行为策略与目标策略之间的不同步问题，从而提升了整体体验质量。我们的解决方案称为 ALISA：采用重要性采样的 Actor-Learner 架构，用于提升自适应比特率算法的体验质量。该方法能够通过基于强化学习的 Actor-Learner 架构生成自适应比特率，无需依赖任何预先编程的模型，也不必对底层系统作出假设。本研究的一项创新贡献是将重要性采样与 A3C 方法相结合，以便在训练过程中考虑可能出现的分布差异，更重要的是，在现实环境中部署模型时也考虑这些差异，从而训练、学习并生成自适应比特率。本研究的主要贡献如下：

- 首先，我们提出了一种新的高效自适应比特率方法，将重要性采样权重与演员-评论家方法相结合，以改善视频交付服务。通过分配重要性采样权重，并进一步赋予相关经验更高的重要性，我们的方法能够更快地学习，并获得比现有最先进的自适应比特率算法整体更高的体验质量。此外，这有助于模型以高效的方式从具有不同分布的样本中学习。
- 其次，我们使用广泛应用的基于 Python 的框架和 MahiMahi 模拟器 [12]分析了所提出方法的性能。我们利用来自 FCC [13],挪威 [14], OBOE [15], 和实时视频流 [16]的轨迹，选取多个数据集进行性能评估。我们针对三种不同的体验质量指标变体开展了全面研究，即 QoE_{lin}、QoE_{log}和 QoE_{HD}，并将其制定为用于深度强化学习的奖励。最后，我们还比较了不同网络特征下的性能，同时考虑无损和有损两种情况。
- 第三，我们将所提出的方法与其他最先进的自适应码率算法进行了比较。这包括与A3C的基础实现（原始 A3C，使用Pensieve框架） [7] 以及与各种非强化学习自适应比特率算法（如 RB、 [17], BOLA、 [18], RobustMPC、 [19], 等）的比较。我们的结果表明，与原始 A3C（Pensieve）相比，ALISA 可将平均体验质量提高 25%–48%。然而，与固定规则调度器相比，改进幅度要大得多。

本文其余部分的组织结构如下。第二节介绍与自适应比特率算法相关的工作。第三节介绍强化学习和演员-评论家方法的相关背景。此外，第四节介绍问题陈述、重要性采样权重的整合、所提出的算法和系统设计。我们分别在第五节和第六节介绍实验设置和结果。最后，我们在第七节总结本文的工作。

II. 相关工作

人们已经开发出几种自适应比特率算法 [18]–[21] ，以便通过无线网络为视频传输提供自适应比特率。这些算法本质上可分为基于速率或基于缓冲区两类。基于速率的算法[17]根据可用网络带宽和数据块历史记录，将未来数据块的比特率预测为所支持的最大比特率；而基于缓冲区的算法则根据客户端的缓冲区占用量进行预测 [18], [20]。由于其中大多数推荐技术都基于预定义规则，因此存在许多缺点。首先，这些算法容易受到网络状况突变的影响，可能导致预测错误。其次，尽管存在多种实现更高体验质量的方法，但每种选择都需要进行权衡。例如，为每个数据块使用所支持的最高比特率，可能会因视频分辨率变化而导致流畅性降低。最后，为当前数据块选择的比特率往往会影响后续数据块所选择的比特率。例如，以最高可用比特率下载数据块，可能会导致后续数据块的比特率和质量下降，以避免重新缓冲。

近年来，除了基于固定规则的自适应比特率算法外，机器学习和深度学习也被广泛用于生成自适应比特率算法。 [22] 中的工作将模型预测控制（MPC） [19] 与深度学习结合，以实现更准确的吞吐量估计。此外， [23] 中的工作结合了机器学习、深度学习和强化学习，与以往基于规则的方法和基于机器学习的方法相比，提升了体验质量。 [24]对比特率进行建模，将其表示为输入参数的线性组合，并使用深度神经网络学习合适的函数。[25]也使用深度学习模型，学习特定用户在视频中的兴趣区域，以有效分配码率预算。在模型预测控制框架下，该方法与常规的带宽和缓冲区占用量一并进行联合考量，结果表明其优于语义无关方法。

最近，人们一直致力于开发一类利用强化学习的新型自适应比特率算法。人们已进行了大量尝试，将Q学习应用于此任务 [26], [27]。然而，这些工作采用表格型Q学习方法，这使得将其扩展到更大的状态空间成为不可能。此外，仅根据最近看到的分块来预测比特率是在[26]中完成的。它忽略了许多最近看到的数据块，而这些数据块能够提升整体性能。为了解决大状态空间中Q学习面临的问题， [7], [8], [28]–[30] 探索了用于生成自适应比特率方案的演员-评论家方法。在这些论文中，A3C智能体被用于生成自适应比特率方案，并且其体验质量高于大多数其他固定规则型自适应码率算法。然而，A3C面临的主要问题是，与中央学习器的目标策略相比，演员行为策略存在滞后 [11], [31]–[34]。这会影响现有强化学习中A3C智能体的性能-

基于视频分发系统，从而导致采样效率降低并获得次优策略。我们提出并评估了根据经验的相关性为其整合重要性采样权重的方法，以解决现有基于 HTTP 的视频分发系统中 A3C智能体实现的一个重要局限。

III. 背景

本节将简要介绍强化学习和演员-评论家方法。

A. 强化学习

强化学习解决方案 [6] 旨在通过强化学习智能体与环境之间的反复交互，学习从状态空间到动作空间的映射。强化学习问题被建模为包含状态和动作的马尔可夫决策过程。考虑一个离散系统，在每个时间步 $t \in \{0, 1, 2, ...\}$，强化学习智能体观测其状态 s_t，采取动作 a_t，转移到状态 s_{t+1} 并获得奖励 $R(s_t, a_t) = r_{t+1}$。此外，对于状态和动作序列，折扣累积奖励定义为$R(\tau) = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$ 其中 τ 是状态和动作序列，即 $\{(s_t, a_t),(s_{t+1}, a_{t+1}) ...\}$，而 $\gamma \leq 1$ 是折扣因子。智能体根据策略选择动作，策略为π：$\pi_\theta(s_t, a_t) \rightarrow [0, 1]$，其中 $\pi_\theta(s_t, a_t)$是状态 s_t 下采取动作 a_t 的概率，θ 是动作所依据的策略参数。遵循策略 π时，状态 s 的价值函数 $V(s)$定义为$V(s) = \mathbb{E}_\pi[R(\tau)|S_t = s]$。强化学习智能体的目标是找到使总体折扣奖励最大化的最优策略 π^*。最优策略表示为，

$$\pi^*(s_t) = \underset{a}{\operatorname{argmax}}[R(s_t, a_t) + \gamma V(s_{t+1})] \tag{1}$$

在这一框架下，我们可以采用基于价值的方法，该方法学习一个将每个状态-动作对映射到一个价值的价值函数。状态中价值最大的动作就成为要采取的最优动作。我们也可以采用基于策略的方法，直接优化策略函数，具体如下一小节所述。

B. 演员-评论家方法

作为对基于价值的方法的改进，[10]提出了演员-评论家方法，它通过降低估计量的方差来加快学习速度。演员-评论家方法由两个模型组成：一个学习最优策略的演员，以及一个近似价值函数（状态-动作对效用）的评论家。在每一步 t，我们使用当前状态 s_t，通过策略π预测动作 a_t。这也会返回奖励 r_{t+1}。利用这些信息，评论家现在计算状态-动作对的价值，$\hat{q}_w(s_t, a_t)$。关于其参数θ的策略更新，以梯度算子 ∇ 表示如下，

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) \hat{q}_w(s_t, a_t) \tag{2}$$

其中 α 是演员的学习率，$\hat{q}_w(s_t, a_t)$ 是评论家函数，用于表示在状态 s_t 下动作 a_t 的优劣。评论家函数的参数 w 按如下方式更新，

$$\Delta w = \xi(R(s_t, a_t) + \gamma \hat{q}_w(s_{t+1}, a_{t+1}) - \hat{q}_w(s_t, a_t)) \nabla_w \hat{q}_w(s_t, a_t) \tag{3}$$

其中 ξ 是评论家的学习率。

然而，以这种方式训练的策略网络可能具有较高的方差，这会导致训练过程不稳定。为缓解这一问题，优势演员-评论家（A2C） [10]框架引入了优势函数，用于确定在状态 s_t 下所采取动作相较于 s_t中动作平均价值的优势。优势函数定义为时序差分（TD）误差:

$$A(s_t, a_t) = R(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t) \tag{4}$$

演员的最终基于梯度的更新如下，

$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) A(s_t, a_t) + \beta \nabla_\theta H(\pi_\theta(.|s_t)) \tag{5}$$

其中，$H(\pi_\theta(.|s_t))$ 是熵因子，用于促进随机动作；β 是正则化项。熵项定义为

$$H(\pi_\theta(.|s_t)) = -\sum_a \pi_\theta(a|s_t) \log(\pi_\theta(a|s_t)) \tag{6}$$

β 的值最初被设为较高值，以促进早期探索，并随着训练的进行而降低。为提高训练速度，提出了异步优势演员-评论家 (A3C) 框架 [10]，以并行且异步的方式模拟多个演员。这些演员定期与中央学习器同步其参数。在我们的工作中，我们使用经过修改的 A3C 框架，为视频传输服务生成自适应比特率算法。

IV. 问题陈述与所提出的解决方案

本节讨论采用重要性采样的基于强化学习的视频分发服务的问题构建与所提出的解决方案，以及系统设计的具体细节。

A. 问题

强化学习智能体通常需要大量经验，才能有效且准确地对环境进行建模。实现这一目标的一种常见方法是在训练期间增加演员的数量。然而，这种策略存在固有缺陷。如图 1 所示，中央学习器首先将其权重与所有演员的权重同步（步骤 1），随后演员将其经验提供给中央学习器（步骤 2）。但是，有时一些演员可能会延迟向中央学习器发送更新信息。例如，如图 1 所示，中央学习器甚至在收到右侧演员的经验之前就更新了其目标策略（步骤 3）。因此，与该事件相关联的行为策略落后于目标策略，这些经验对于更新当前目标策略的作用可能较小。当演员开始使用较旧版本的行为策略生成经验时，演员数量越多，这一问题就越严重。

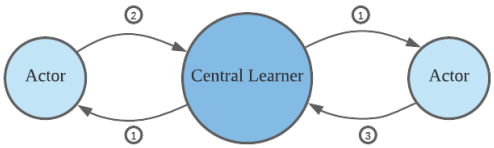

图 1. A3C 滞后问题示意图。该图展示了三个步骤。步骤 (1)：每个演员都与中央学习器同步其权重；步骤 (2)：其中一个演员向中央学习器提供经验，中央学习器据此更新目标策略的权重；步骤 (3)：另一个演员向中央学习器提供经验。该经验所依据的行为策略未与最新版本的目标策略（在步骤 2 中更新）同步，因此该经验是基于较旧的策略获得的。

行为策略与目标策略之间缺乏同步会导致次优更新。最终，这会导致学习到整体次优的策略。我们计划开发相应方法，以弥补训练期间行为策略落后于目标策略的问题，从而在未见过的测试数据上取得更高的性能。

B. ALISA：系统设计

我们提出的解决方案 ALISA 基于 DASH 框架，并采用基于深度强化学习的 A3C 方法，使视频流传输的体验质量高于现有最先进的自适应比特率算法。图 2 展示了 ALISA 的整体系统设计。用户通过视频播放器在设备上观看视频流，其主要组件是自适应比特率控制器。它会观测客户端侧的多个状态参数，例如带宽、比特率选择历史记录和缓冲区占用量，并决定要采取的操作，即为下一个分块选择比特率。每一步中，它也会观测其操作所产生的奖励（体验质量）。

现在我们描述ALISA 的 ABR 控制器所采用的训练过程。如图 3所示，训练环境由多个演员组成，这些演员由单个中央学习器进行协调。演员包含行为策略作为其参数，而中央学习器维护目标策略和评论家参数。行为策略、目标策略和评论家函数均建模为神经网络。训练过程可以视为重复执行以下步骤，直至收敛：

1) 首先，一个演员模拟一个回合，并生成一批经验，其中包括状态、演员采取的相应动作以及由此获得的奖励。2) 随后，这些经验被传回中央学习器。3) 中央学习器使用评论家参数计算经验中每一步的值。4) 在纳入下一小节讨论的重要性采样策略后，中央学习器计算 $V-$迹目标。5) 评论家梯度使用观测到的状态及其相应奖励进行计算，而目标策略梯度则使用观测到的状态、相应动作、获得的奖励，以及 $V-$跟踪目标。目标策略和评论家网络现已通过反向传播进行更新。

6) 最后，中央学习器将最新版本的目标策略共享给每个执行器，使其将行为策略设置为最新的目标策略，以生成下一批经验。

ALISA 有效地将执行过程与学习过程解耦，同时补偿由此产生的离策略偏移。这对自适应比特率算法的开发具有重要意义。由于全球用户传输的视频数量极其庞大，ALISA 架构能够持续微调自适应比特率算法，并适应不断变化的网络状况，同时不会危及用户隐私。尽管流媒体设备持续进行比特率选择，但这些决策可以传送至位于远程云服务器上的中央学习器，在那里可以进行联邦学习 [35]。最新策略可以定期在终端用户设备与远程服务器之间同步。这使视频流媒体服务能够更快地响应网络状况的波动以及视频流传输行为随时间发生的变化，从而大幅加快新模型的重新训练速度。尽管受可用数据限制，我们无法充分展示 ALISA 在微调方面的优势，但我们在第VI节中表明，ALISA 不仅能够实现更高的体验质量，而且所需时间还不到可比方法的 50%。我们预计，从头开始训练模型时观察到的这一优势，在按照我们所建议的方式对模型进行微调时同样能够体现出来。

C. ALISA：将重要性采样用于策略更新

重要性采样是一种用于解决数据分布不匹配问题的常用技术。它通过从同一域 $\mathcal{D}$ 上的另一分布 b 中采样值，来估计函数 $f(x)$的期望值，其中 x遵循定义在域 $\mathcal{D}$上的概率密度函数 a ，即：

$$\mathbb{E}(f(X)) = \mathbb{E}_r\left(\frac{f(X)a(X)}{b(X)}\right) \tag{7}$$

重要性采样会改变从分布 b 中收集的数据，使其看起来像是从分布 a中采样得到的。这有效地解决了分布不匹配问题。在本文中，我们将重要性采样与 ALISA相结合，以克服目标策略 π 与行为策略μ之间的分布不匹配问题。对应于公式（7）中的符号，我们有 $a \equiv \pi$和 $b \equiv \mu$。与 [11], 的作者类似，我们使用 n-步V-迹目标来调整异策略偏移。 n-步 V- 迹目标现在充当目标策略 π 的价值函数 V的估计，其中使用的是行为策略 μ的较早版本。 n-步 V-迹目标定义为：

$$v_j \doteq V(s_j) + \sum_{t=j}^{j+n-1} \gamma^{t-j}\left(\prod_{i=j}^{t-1} c_i\right)\delta_t V \tag{8}$$

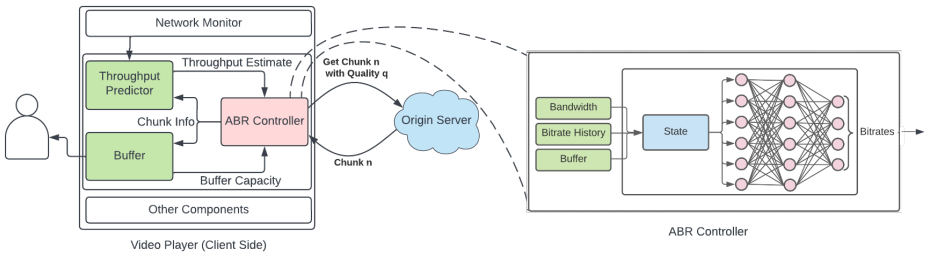

图 2. ALISA：系统设计

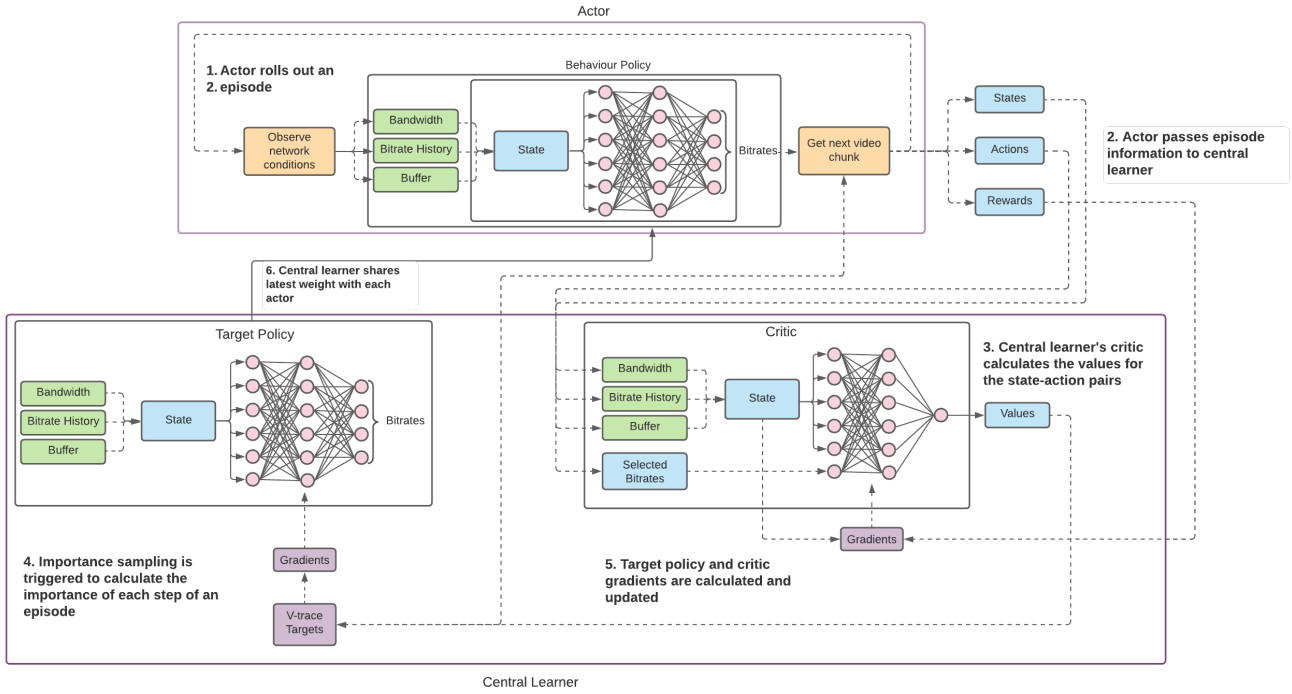

图 3. ALISA 的基于 RL 的自适应比特率控制器训练流程详解

其中，

- $\delta_t V \doteq \rho_t(r_t + \gamma V(s_{t+1}) - V(s_t))$ 是时间差。
- $\rho_t \doteq min\left(\bar{\rho}, \frac{\pi(a_t|s_t)}{\mu(a_t|s_t)}\right)$ 和 $c_i \doteq min\left(\bar{c}, \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}\right)$ 是重要性采样权重。重要性采样权重 ρ_t 和 c_i 用于赋予经验以重要性，这些经验与目标策略的相关性高于行为策略。此处，π 表示目标策略，μ表示行为策略。
- $\bar{\rho}$ 和 $\bar{c}$ 分别是对应重要性采样权重的下阈值，在我们的整个工作中均设为 1。
- $\overline{\rho_t}$ 表示在状态 x_t 下采取动作 a_t 遵循目标策略的可能性相较于行为策略高出多少。• $\prod_{i=j}^{t-1} c_i$ 表示预测的可能性高出多少。

状态 s_j 到 s_{t-1} 的路径与行为策略相比，是遵循目标策略的。

随后，使用 V-trace 目标代替 V 进行梯度计算。 n-步 V-trace 目标也可以递归定义为

$$v_j = V(s_j) + \delta_j V + \gamma c_j(v_{j+1} - V(s_{j+1})) \tag{9}$$

这是我们在整个工作实现过程中采用的方法。根据这些计算结果，按照目标策略更有可能采取的动作对 V-trace 目标的贡献更大。因此，重要性采样权重有助于强化学习模型聚焦于更相关的经验，从而带来更好的参数更新，并降低对次优经验的重视程度。根据上述 V-trace 目标的定义，算法 1 概述了 ALISA 使用重要性

采样权重的策略更新算法，其中 n 等于回合长度。ALISA 的 V-trace 目标计算以与一个回合相关的信息作为输入，包括状态序列（s）、按照行为策略得到的动作概率序列（a_b）、奖励序列（r），以及元参数 ρ 和 c，还有第 2 行至第 8 行中的演员模型和评论家模型。因此，在第 15 行，ρ 被更新为 ρ 与目标策略和行为策略之比中的较小值。在第 16 行，计算要添加到 s 的评论家价值（V）中的变化。随后计算 c，并在第 17 行将其设为 c 和ρ 中的较小值。通过将第 16 行计算出的值加到当前的 V-trace 目标上，更新 V-trace 目标。第 19 行至第 21 行说明了循环中每个 **i** 对应的 V-trace 目标如何更新。由于采用了重要性采样，当前目标策略更有可能采取的动作，相比于较早版本的目标策略（这些版本存在滞后，但并非当前策略）更有可能采取的动作，对梯度的贡献更大。该算法引导强化学习模型聚焦于更重要的经验，并降低对其他较不相关经验的重视程度。

算法 1 使用 ALISA 计算回合的 V-trace 目标
一个回合

1: **输入**:
2: $\boldsymbol{s}$: 状态序列
3: $\boldsymbol{a_b}$: 根据以下策略得到的动作概率序列
行为策略
4: $\boldsymbol{r}$: 奖励序列
5: $\bar{\boldsymbol{\rho}}$: ρ的下限
6: $\bar{\boldsymbol{c}}$: c的下限
7: **执行者**: 来自中央学习器的目标策略
8: **评论家**: 评论家模型
9: **输出**:
10: v: V-trace 目标
11: $V \leftarrow$ 评论家价值为 s
12: $p_b \leftarrow$ 最优行为策略概率
动作
13: $a_t \leftarrow$ 针对 s的目标策略概率
14: $p_t \leftarrow$ 目标策略概率对应于
使用 a_t 计算的行为策略的最优动作
以及 p_b
15: $\rho \leftarrow min\left(\rho, \frac{p_t}{p_b}\right)$
16: $\delta_t V \leftarrow \rho(r + \gamma\ V_{+1} - V)$
17: $c \leftarrow min\ (c, \rho)$
18: $v \leftarrow V + \delta_t V$
19: **对于** 每个位置**i** 从 v 的后部**执行**
20: $v_i \mathrel{+}= \gamma c_i * (v_{i+1} - V_{i+1})$
21: **结束循环**

V. 实验细节

本节介绍实验设置以及用于评估 ALISA 性能的性能指标。

A. 实验设置

我们使用 [7]，提出的基于 Python 的框架来生成和测试我们的自适应比特率算法。客户端向服务器请求数据块，并向其提供与观测到的网络状况相关的参数，例如带宽、缓冲区占用量和码率历史。如第 IV 节所述，我们集成了重要性采样，并为目标策略和行为策略分配了权重。为了模拟网络状况，从而有效测试训练好的强化学习模型，我们使用了 MahiMahi [12]框架；这是一个用于此任务的记录与回放 HTTP 框架。

我们使用四个数据集作为训练集的一部分：FCC 提供的宽带数据集 [13],在挪威收集的移动数据集 [14], OBOE 轨迹 [15], 以及实时视频流数据集 [16],，这些数据集已按照 MahiMahi格式进行预处理。随后，我们将 ALISA 与以下最先进的自适应比特率算法进行比较：

- Pensieve（原始 A3C） [7]：使用原始 A3C，不采用任何额外技术来训练智能体以提供自适应比特率。
- 基于速率（RB） [17]：RB 根据过去观测到的吞吐量的调和平均数预测最大支持比特率。
- 基于缓冲区（BB） [20]：BB 根据客户端的缓冲区占用量选择比特率。
- BOLA [18]：比特率选择仅基于缓冲区占用量，采用李雅普诺夫优化。
- RobustMPC [19]：模型预测控制使用与 RB 类似的缓冲区占用量观测值和吞吐量预测。此外，Robust MPC 通过对吞吐量估计值进行归一化，来考虑预测吞吐量与观测吞吐量之间的误差。

为量化自适应比特率算法的性能，我们采用以下体验质量公式：

$$QoE = \sum_{n=1}^{N} q(b_n) - \mu \sum_{n=1}^{N} T_n - \sum_{n=1}^{N-1} |q(b_{n+1}) - q(b_n)| \quad (10)$$

其中，b_i 和 q（b_i）分别表示分块 i的比特率和质量。比特率越高，意味着质量越高，体验质量也越高。然而，重新缓冲时间 T_i （由第二项表示）以及视频质量波动（由最后一项表示）也会带来惩罚，从而损害整体流畅度。本文采用三种体验质量变体 [7] 评估所提出方法的性能，这些变体取决于上述通用体验质量指标：

- QoE_{lin}: q（b_n）$= b_n$，其中再缓冲惩罚的取值为$\mu = 4.3$,
- QoE_{log}: q（b_n）$= log$（b/b_{min}），该指标考虑了随着比特率提高，质量的边际改善会逐渐降低这一事实，其中 $\mu = 2.66$，并且
- QoE_{HD}: 对高质量比特率赋予更高的值，对低质量比特率赋予更低的值，并使用 $\mu = 8$。

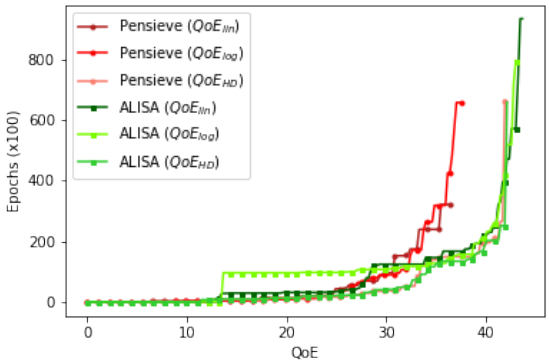

图4. 随已用训练轮数变化所获得的最大训练体验质量。对于体验质量指标的三种变体，ALISA 都能更快获得更高的体验质量。

B. 数据集详情

我们使用以下数据集进行训练、验证和测试。我们选择的训练、验证和测试数据集与先前的实验设置一致 [7], [15], [16]。训练时，我们使用了三个不同的数据集。第一个数据集包含 127 条轨迹，其中 59 条属于 FCC [13] 数据集，其余 68 条属于挪威 HSDPA [14] 数据集。第二个数据集包含 428 条 OBOE 轨迹 [15]，第三个数据集包含 100 条实时视频流轨迹 [16]。为展示 ALISA 在不同训练模型中的优势，我们针对上述三个不同的数据集分别生成了三个训练模型。对于这三个训练模型，我们使用相同的验证数据集，即 142 条挪威轨迹。最后，对于这三个训练模型，验证后使用 FCC 数据集中的 205 条轨迹和挪威 HSDPA 数据集中的 250 条轨迹进行测试。

C. 训练方法

我们在 Pensieve 和ALISA上针对每种配置分别训练三个模型，每个模型分别以 QoE_{lin}、QoE_{log} 和 QoE_{HD} 作为奖励指标。对于每个模型，我们始终使用一致的超参数组。折扣因子 γ 设置为 0.99。演员和评论家的学习率分别设置为0.0001 和 0.001。在 Pensieve 中，熵因子（H（.））通过使用熵正则化因子（β）进行控制。 β 在 100,000 个训练周期内使用从1到 0.1的熵衰减值。在ALISA中，我们将重要性采样阈值 $\bar{\rho}$ 和 $\bar{c}$ 均设置为1。我们针对不同的熵权重配置训练多个模型。首先，我们使用恒定的熵权重训练若干模型100,000 个训练周期。接下来，我们使用衰减的熵权重，使熵在 100,000 个训练周期内逐渐减小。

D. 测试方法

我们选择验证集体验质量最高的模型进行测试。我们使用 MahiMahi [12] 框架模拟无损和有损条件，并在丢包率为 0%、0.1%、0.5%、1% 和 2% 的条件下进行测试，随机丢弃视频流中的数据包。我们使用第 V-A 节讨论的三种不同体验质量指标评估所有模型。

VI. 结果

本节展示 ALISA 与其他最先进的自适应比特率算法的结果和比较。

A. 收敛速度

ALISA 在训练过程中采用重要性采样策略。因此，与 Pensieve（原始 A3C）相比，它通常能在更短的时间内达到更高的体验质量。图 4 展示了训练过程中已完成的轮数与截至当时达到的最高体验质量之间的关系。这些曲线是在使用第一组数据集进行训练时生成的，即 FCC 和挪威数据集中的 127 条轨迹。结果表明，当 ALISA 达到超过 40 的高体验质量时，Pensieve 达到的最高体验质量仅约为 35。此外，ALISA 达到 40 以上体验质量所需的时间不到 Pensieve 达到其最高体验质量所需时间的 $1/3^{rd}$ 。这说明 ALISA 能够更快地学习并适应新的条件，从而缩短训练时间。使用 OBOE 和实时视频流轨迹进行训练时，也观察到了类似结果。

B. 与最先进的自适应码率算法的比较

1) 使用训练集和验证集的结果：我们在三个数据集上进行了全面的训练，并报告了在 QoE_{lin}、QoE_{log} 和 QoE_{HD} 指标上的结果。表 I 展示了在 FCC 和挪威轨迹、OBOE 轨迹以及实时视频流轨迹上训练和验证后获得的奖励。我们还研究了采用不同熵正则化值 β 的效果。具体而言，我们考虑了以下取值：0、0.001、0.01、0.1、0.25、0.5、0.75 和 1。我们还探索了训练过程中 β 的变化。例如，在表 I 中，$\{2, 1.5, 1, 0.5, 0.1\}$ 指的是这样一种情形：前 20000 个训练轮次使用 $\beta = 2$，接下来的 20000 个训练轮次使用 $\beta = 1.5$，依此类推，直至最后 20000 个训练轮次使用 $\beta = 0.1$。同样，我们也采用了固定的熵正则化；在表 I 中，0.1×5 指的是在全部 100000 次迭代中均使用 $\beta = 0.1$ 的情形。

我们注意到，在熵非常低或非常高的情况下，模型在训练过程中都无法很好地收敛。我们发现，对于 QoE_{lin} 和 QoE_{log} 指标，将熵设为 0.1能够取得良好效果，而对于 QoE_{HD}，将其设为 0.75效果良好。我们还多次采用衰减熵正则化进行训练。我们从一个较高的值开始，每20,000个训练周期逐渐减小熵权重，最终逐步降至0.1。正如表 I 所示，我们发现，在几乎所有情况下，衰减熵正则化都更加有效，因为经过几个训练周期后，无需进行高强度探索即可获得最优策略。

2) 测试数据集上的结果：我们还将 ALISA 与其他几种最先进的自适应码率算法进行了比较，例如上一节中介绍的 RB、BB、BOLA 和 RobustMPC。我们还将ALISA 与基于强化学习的 Pensieve 进行了比较，后者是一种基于强化学习的

表 I 使用全部三个数据集和全部三种体验质量指标变体训练 ALISA 后的平均体验质量。

熵值	FCC 和挪威轨迹			OBOE 轨迹			实时视频流轨迹		
	QoE_{lin}	QoE_{log}	QoE_{HD}	QoE_{lin}	QoE_{log}	QoE_{HD}	QoE_{lin}	QoE_{log}	QoE_{HD}
0 (×5)	13.61	13.62	13.57	30.76	30.8	39.51	13.61	30.75	29.97
0.001 (×5)	13.61	13.62	13.61	42.47	30.78	39.52	13.61	30.87	13.57
0.01 (×5)	13.61	13.61	13.61	38.79	39.25	39.44	30.53	30.93	37.84
0.1, 2, 0.1, 2, 0.1	41.57	38.91	42.21	41.43	44.03	38.4	43.27	38.54	41.36
0.1 (×5)	43.88	43.29	37.88	42.89	38.37	38.55	44.1	30.76	27.96
0.25 (×5)	40.91	43.29	37.61	42.85	43.2	39.57	43.22	44.03	37.69
0.5 (×5)	38.11	37.14	**42.42**	36.79	37.99	39.17	37.86	38.35	41.71
0.75 (×5)	31.11	32.11	41.99	29.85	30.69	40.97	32.23	32.83	42.28
1, 0.75, 0.5, 0.25, 0.1	43.22	44.09	42.4	**44.34**	**44.94**	40.61	**44.76**	**45.96**	41.66
1 (×5)	27.46	25.8	40.98	25.07	25.15	41.11	26.41	26.07	41.17
2, 1.5, 1, 0.5, 0.1	43.86	44.19	41.27	43.16	43.73	**41.16**	44.45	45.54	41.76
3, 2, 1, 0.5, 0.1	**43.92**	44.31	42.11	43.95	43.36	40.56	44.25	44.54	42.32
4, 2, 1, 0.5, 0.1	43.15	43.4	41.8	42.99	44.93	39.33	43.92	45.93	**42.66**
5, 2, 1, 0.5, 0.1	41.94	**44.8**	42.06	43.23	42.33	40.99	43.52	45.12	41.64

表 II 在无数据包丢失的仿真环境下，使用体验质量指标的三种变体在所有数据集上取得的平均体验质量。

算法	FCC 和挪威			OBOE			实时视频流		
	线性	Log	HD	线性	Log	HD	线性	Log	HD
ALISA	**43.03**	**42.37**	**256.29**	**42.5**	**41.79**	**237.27**	**46.57**	**44.36**	228.63
Pensieve	39.62	35.26	239.08	37.52	37.01	194.29	39.12	41.68	**234.72**
BB	12.02	12.78	84.24	14.08	20.00	80.36	13.81	20.26	63.08
RB	35.62	36.45	139.82	36.15	37.97	138.02	37.44	37.35	120.52
BOLA	34.25	35.3	141.04	35.04	37.09	139.1	35.82	36.05	121.02
RobustMPC	39.93	40.44	195.52	40.21	38.03	188.65	40.59	38.99	177.58

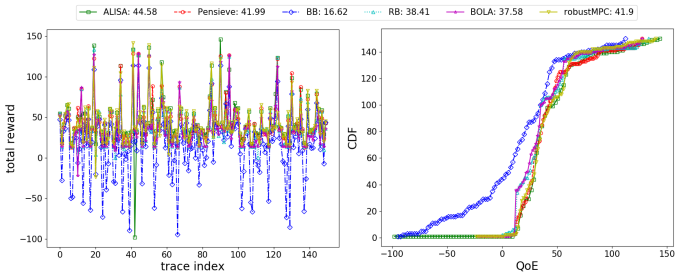

图5.使用QoE_{linear} 指标比较 ALISA 与其他自适应比特率算法：(左) 所有测试轨迹的平均奖励；(右) 在以0.1%的概率仿真随机丢包条件下的 CDF 与体验质量图。

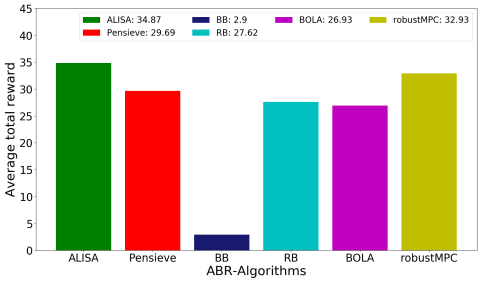

图 6. 在 1% 丢包率的仿真环境下，使用公式 10 的QoE_{linear} 指标比较 ALISA 与其他自适应比特率算法

表 III
在仿真中以 0.1% 的概率随机丢包时，使用体验质量指标的全部三种变体在所有数据集上达到的平均体验质量。

算法	FCC 和挪威			OBOE			实时视频流		
	线性	Log	HD	线性	Log	HD	线性	Log	HD
ALISA	**44.58**	**43.75**	**244.38**	**44.46**	**42.07**	**237.91**	**45.76**	**41.89**	233.96
Pensieve	41.99	35.06	235.12	39.34	36.69	194.37	36.62	39.08	**241.10**
BB	16.62	20.97	73.16	17.08	19.81	0.64	15.40	19.58	68.76
RB	38.41	38.73	132.98	39.05	37.64	35.58	37.15	37.65	130.14
BOLA	37.58	37.21	129.27	37.84	36.03	34.81	35.95	36.08	126.24
RobustMPC	41.90	37.30	189.53	42.80	37.97	187.43	40.97	37.56	182.75

表 IV 在仿真中以 0.5% 的概率使用体验质量指标的全部三种变体时，在所有数据集上取得的平均体验质量。

算法	FCC 和挪威轨迹			OBOE 轨迹			实时视频流 轨迹		
	线性	Log	HD	线性	Log	HD	线性	Log	HD
ALISA	37.34	**43.86**	**236.15**	39.2	**42.24**	**231.37**	**42.53**	**43.33**	227.66
Pensieve	35.43	34.1	214.59	36.44	36.22	201.32	34.98	40.80	**231.19**
BB	10.59	18.74	66.13	13.97	18.79	69.6	12.12	18.41	64.51
RB	32.91	35.89	108.86	34.19	35.35	117.44	33.99	34.92	111.23
BOLA	32.39	34.59	108.98	34.12	33.97	112.67	33.78	33.77	110.80
RobustMPC	**37.99**	38.2	177.87	**39.48**	38.46	184.96	38.62	38.09	179.55

表 V
在 1% 概率的仿真条件下，使用三种体验质量指标变体时，在所有数据集上达到的平均体验质量。

算法	FCC 和挪威轨迹			OBOE 轨迹			实时视频流轨迹		
	线性	对数	高清	线性	对数	HD	线性	对数	高清
ALISA	**34.87**	**38.18**	**198.34**	**35.35**	**44.86**	**181.71**	**36.97**	**38.2**	187.68
Pensieve	29.68	29.22	174.53	29.04	30.18	138.71	29.50	29.22	**188.78**
BB	2.90	10.11	19.07	2.91	10.57	12.67	6.43	10.11	-15.75
RB	27.62	28.58	78.17	27.38	28.72	77.99	28.00	28.58	25.38
BOLA	26.93	27.74	77.30	27.59	27.95	77.12	26.89	27.74	25.00
RobustMPC	32.92	33.13	149.21	32.62	33.47	142.51	33.74	33.13	145.87

表 VI
在 2% 概率下进行仿真时，使用三种体验质量指标变体在所有数据集上达到的平均体验质量。

算法	FCC 和挪威轨迹			OBOE 轨迹			实时视频流 轨迹		
	线性	Log	HD	线性	Log	HD	线性	Log	HD
ALISA	**27.60**	**28.98**	**128.53**	**25.62**	**26.67**	**122.05**	**28.73**	**30.83**	**122.47**
Pensieve	24.28	21.48	111.57	22.35	22.98	82.05	22.45	28.56	121.41
BB	-13.44	-6.74	-41.18	-15.48	-6.28	-37.73	-14.49	-4.15	-38.48
RB	16.09	17.35	13.44	17.27	17.75	14.88	17.21	18.22	14.21
BOLA	17.89	18.59	16.05	18.08	18.01	16.08	17.48	18.76	16.49
RobustMPC	24.43	20.72	99.98	25.28	21.66	99.14	24.98	21.69	97.09

不利用重要性采样权重的基本 A3C 方法。表 II 展示了网络中不存在丢包时的比较结果。我们的结果表明 that

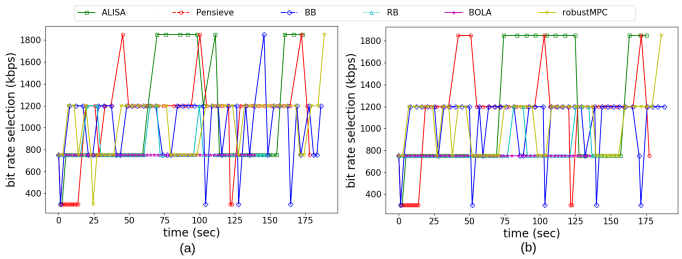

图 7. ALISA 与其他自适应比特率算法的比特率选择对比：(a) 样本轨迹 1 的比特率选择；(b) 使用 FCC 和挪威训练模型，在仿真中丢包率为 1% 时，样本轨迹 2 的比特率选择，体验质量指标为 QoE_{lin} （公式 10）。

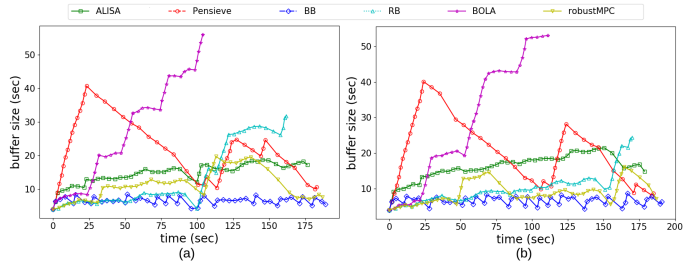

图 8. ALISA 与其他自适应比特率算法的缓冲区大小选择对比：(a) 样本轨迹 1 的缓冲区大小选择；(b) 使用 FCC 和挪威训练模型，在仿真中丢包率为 1 % 时，样本轨迹 2 的缓冲区大小选择，体验质量指标为 QoE_{lin} （公式 10）。

ALISA 在所有不同配置下的所有指标上都实现了更高的体验质量。与 RB 相比，ALISA 的体验质量最高提升 25%；与 BB 相比最高提升 230%；与 BOLA 相比最高提升 30%；与 RobustMPC 相比最高提升 25%；在无丢包条件下测试时，与 Pensieve 相比最高提升 20%。这一性能在有损条件下同样得以体现。我们注意到，在丢包率分别为 0.1%、0.5%、1% 和 2% 的情况下，与 Pensieve 相比，ALISA 的体验质量最高分别提升 25% 、 28% 、 48% 和 48%。对于随机丢包率，我们总结了其余的体验质量指标

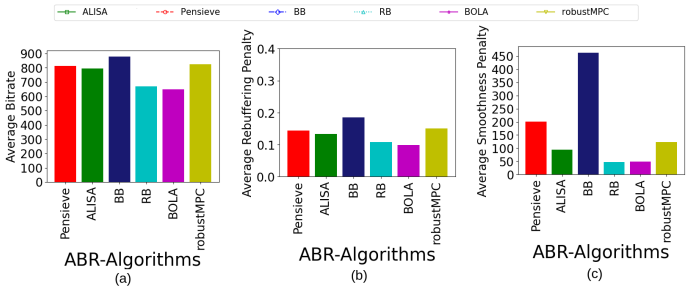

图 9. 通过分析各组成部分的表现，将 ALISA 与现有自适应比特率算法进行比较：(a) 平均比特率；(b) 平均重缓冲惩罚；(c) 体验质量指标为 QoE_{lin} 时的平均平滑度惩罚（仿真中丢包率为 1%，公式 10）。

丢包率分别为 0.1%、0.5%、1% 和 2%，对应表 III、表 IV、表 V 和表 VI。这些结果表明，ALISA 的表现显著优于许多其他固定规则型自适应码率算法以及 Pensieve。此外，我们还将公式（10）中的体验质量指标分解为不同组成部分并进行可视化，以了解 ALISA 优于其他自适应比特率算法的原因。图 5 展示了网络仿真丢包率为 0.1% 时，各自适应比特率算法在每条轨迹上使用 QoE_{lin} 体验质量指标所获得的总奖励。结果表明，ALISA 算法的平均体验质量达到 44.58，高于其他自适应比特率算法。图 6 展示了网络仿真丢包率为 1% 时，各自适应比特率算法在每条轨迹上使用 QoE_{lin} 体验质量指标所获得的平均总奖励。结果表明，ALISA 算法的平均体验质量达到 34.87，高于其他自适应比特率算法。

图 7(a) 和图 7(b) 展示了对于随机样本轨迹，ALISA 如何始终比其他方法实现更高的比特率。这提高了体验质量的第一项。从图 8(a) 和图 8(b) 可以看出，对于随机样本轨迹，与 Pensieve 相比，ALISA 保持了足够的缓冲区大小；由于比特率适中，这使得公式 (10) 中的第二项有所降低，同时缓冲区大小也保持适中，因此 ALISA 的再缓冲惩罚较低。另一方面，Pensieve 保持的缓冲区大小高于 ALISA，这使其能够实现较高的比特率。较高的比特率会导致较高的再缓冲惩罚。总体而言，与其他自适应比特率算法相比，ALISA 带来了更高的体验质量。

为了解并展示 ALISA 的更优性能，我们分析体验质量指标的各个组成部分，并在图 9 中比较各种自适应比特率算法在仿真条件下、1% 丢包率时， QoE_{lin} 指标对应的平均播放比特率、平均再缓冲惩罚和平均平滑度惩罚。结果表明，除图 9(a) 中的 BOLA 和 RB 外，大多数自适应比特率算法都实现了较高的比特率。由于选择了较高的比特率，其中一些算法受到再缓冲惩罚的影响；图 9(b) 显示，BB 和 robustMPC 的再缓冲惩罚最高。同样，图 9(c) 显示，BB 也受到较高的平滑度惩罚。ALISA 实现了较高的平均比特率，同时再缓冲惩罚和平滑度惩罚相对较低。这些组成部分的综合影响使 ALISA 的平均体验质量高于其他自适应比特率算法。在 0.1%、0.5%、1% 和 2% 的数据包丢失率下进行仿真时，我们在 QoE_{lin} 、 QoE_{log} 和 QoE_{HD} 指标上也观察到了类似结果。

VII. 结论

我们展示了，在生成视频传输服务的自适应比特率算法时，结合重要性采样与结构化熵选择，可以显著提升原始 A3C 方法（使用 Pensieve 框架）的性能。通过将这些方法融入我们提出的系统 ALISA，我们能够持续实现

体验质量提升幅度最高可达 25-48 %，在某些情况下甚至更高。此外，我们还在更广泛的网络状况范围内，根据丢包情况评估了我们的方法，并观察到相当的收益。最后，我们将 ALISA 的比特率选择和缓冲区大小与其他自适应比特率算法（RB、BB、BOLA 和 RobustMPC）的相应指标进行了可视化比较，并表明 ALISA 在这两个方面都优于它们，从而带来了更高的体验质量。未来的工作将研究用于实现 ALISA 的先进云边混合架构。此外，我们计划在联邦环境中研究 ALISA，以利用跨多个去中心化边缘设备的分布式训练。

致谢

本研究得到印度TCS 基金会在 TCS 研究学者计划（2019—2023 年）下的支持。

参考文献

[1] “Sandvine 全球互联网现象报告，2020。” https://www.sandvine.com/phenomena. [2] “思科年度互联网报告，2018-2023。” https://www.cisco.com/c/
e
n/
u
s/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html. [3] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan 和 H. Zhang，“理解视频质量对用户参与度的影响”，载于 *ACM SIGCOMM 2011* 会议论文集，SIGCOMM ’11，（美国纽约州纽约市），第 362–373 页，计算机协会，2011。 [4] I. Sodagar，“用于互联网多媒体流传输的 MPEG-DASH 标准”，*IEEE* 多媒体，第 18 卷，第 4 期，第 62–67 页，2011。 [5] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer 和 R. Zimmermann，“HTTP 流媒体比特率自适应方案综述”，*IEEE* 通信调查与教程，第 21 卷，585，2019 1，562第。页 – 。 [6] R. S. Sutton 和 A. G. Barto，强化学习：导论。美国马萨诸塞州剑桥：A Bradford Book，2018。 [7] H. Mao, R. Netravali 和 M. Alizadeh，“使用 Pensieve 的神经自适应视频流媒体”，载于 *ACM* 数据通信特别兴趣小组会议论文集，SIGCOMM ’17，（美国纽约州纽约市），第 197–210 页，计算机协会，2017。 [8] P. Saxena, M. Naresh, M. Gupta, A. Achanta, S. Kota 和 S. Gupta，“Nancy：用于无线网络视频分发的神经自适应网络编码方法”，载于 *GLOBECOM 2020——2020* 年 *IEEE* 全球通信会议，第 1–6 页，2020。 [9] V. Mnih 等，“2016。深度强化学习的异步方法”，载于 国际机器学习会议，19281937。V. Mnih 等，深度强化学习的异步方法。载于国际机器学习会议。19281937，2016。 [10] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver 和 K. Kavukcuoglu，“深度强化学习的异步方法”，*CoRR*，第 abs/1602.01783 卷，2016。 [11] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning，等，“Impala：采用重要性加权行动者—学习者架构的可扩展分布式深度强化学习”，载于 国际机器学习会议（*ICML*）论文集，2018。 [12] R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens 和 H. Balakrishnan，“MahiMahi：针对 HTTP 的精准记录与重放”，USENIX ATC ’15，（美国），第 417–429 页，USENIX 协会，2015。 [13] 美国联邦通信委员会，“美国联邦通信委员会。2016。原始数据——测量美国宽带”，2016。 [14] H. Riiser, P. Vigmostad, C. Griwodz 和 P. Halvorsen，“3G 网络通勤路径带宽轨迹：分析与应用”，MMSys ’13，（美国纽约州纽约市），第 114–118 页，计算机协会，2013。 [15] Z. Akhtar，“OBOE：根据网络状况自动调优视频自适应比特率算法”，*OBOE*：根据网络状况自动调优视频自适应比特率算法，2018 年 8 月 20–25 日，匈牙利布达佩斯。

[16] G. Yi，“ACM 多媒体 2019 直播视频流媒体挑战赛”，ACM 多媒体 2019 直播视频流媒体挑战赛，2019 年 10 月 21–25 日，法国尼斯。 [17] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu 和 B. Sinopoli，“CS2P：利用数据驱动的吞吐量预测改进视频比特率选择与自适应”，2016 年 ACM SIGCOMM 会议论文集，2016。 [18] K. Spiteri, R. Urgaonkar 和 R. K. Sitaraman，“BOLA：在线高清视频的近最优比特率自适应”，载于 IEEE INFOCOM 2016——第 35 届 IEEE 国际计算机通信会议，第 1–9 页，2016。 [19] X. Yin, A. Jindal, V. Sekar 和 B. Sinopoli，“一种用于 HTTP 动态自适应视频流媒体的控制理论方法”，载于 2015 年 ACM 数据通信特别兴趣小组会议论文集，SIGCOMM ’15，（美国纽约州纽约市），第 325–338 页，计算机协会，2015。 [20] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell 和 M. Watson，“基于缓冲区的速率自适应方法：来自大型视频流媒体服务的证据”，载于 2014 年 ACM SIGCOMM 会议论文集，SIGCOMM ’14，（美国纽约州纽约市），第 187–198 页，计算机协会，2014。 [21] M. A. Pimentel-Ni˜no, P. Saxena 和 M. A. Vazquez Castro，“面向尽力而为卫星擦除链路的体验质量驱动型自适应视频与重叠网络编码”，载于第 31 届 AIAA 国际通信卫星系统会议，第 5668 页，2013。 [22] T. Huang 和 L. Sun，“DeepMPC：一种结合深度学习与模型预测控制的混合自适应比特率方法”，载于 2020 年 IEEE 国际图像处理会议（ICIP），第 1231–1235 页，2020。 [23] L. Amour, S. Souihi, A. Mellouk 和 M. Mushtaq，“Q2ABR：体验质量感知型自适应视频比特率解决方案”，《国际通信系统杂志》，第 33 卷，2020 年 07 月。 [24] A. Lekharu, K. Y. Moulii, A. Sur 和 A. Sarkar，“用于自适应视频流媒体的深度学习预测模型”，载于 2020 年国际通信系统与网络会议（COMSNETS），第 152–159 页，2020。 [25] G. Gao, H. Zhang, H. Hu, Y. Wen, J. Cai, C. Luo 和 W. Zeng，“通过观众兴趣推断优化自适应比特率流媒体的体验质量”，IEEE 多媒体汇刊，第 20 卷，第 12 期，第 3399–3413 页，2018。 [26] F. Chiariotti, S. D’Aronco, L. Toni 和 P. Frossard，“ 面向 DASH 客户端的在线学习自适应策略”，载于第七届国际多媒体系统会议论文集，MMSys ’16，（美国纽约州纽约市），计算机协会，2016。 [27] M. Claeys, S. Latr´e, J. Famaey, T. Wu, W. V. Leekwijck 和 F. D. Turck，“基于（模糊）Q学习的 HTTP 自适应流媒体客户端的设计与优化”，《连接科学》，第 26 卷，第 1 期，第 25–43 页，2014。 [28] H. Jin, Q. Wang, S. Li 和 J. Chen，“基于多智能体强化学习的视频流传输联合 QoS 控制与比特率选择”，载于 2020 年 IEEE 第 16 届控制自动化国际会议（ICCA），第 1360–1365 页，2020。 [29] M. Naresh, V. Das, P. Saxena 和 M. Gupta，“面向物联网环境中视频流传输的深度强化学习体验质量感知行动者—学习者架构”，《计算》，第 104 卷，2022 年 07 月。 [30] M. Naresh, N. Gireesh, P. Saxena 和 M. Gupta，“SAC-ABR：基于软行动者—评论家的自适应比特率流媒体深度强化学习”，载于 2022 年第 14 届通信系统与网络国际会议（COMSNETS），第 353–361 页，2022。 [31] S. Chen, X.-F. Zhang, J.-J. Wu 和 D. Liu，“用于异步深度强化学习的平均 A3C”，载于《神经信息处理》（L. Cheng, A. C. S. Leung 和 S. Ozawa 编），（瑞士沙姆），第 277–288 页，施普林格国际出版公司，2018。 [32] A. B. Labao, M. A. M. Martija 和 P. C. Naval，“A3C-GS：用于异步行动者—评论家智能体的带锁自适应动量梯度共享”，IEEE 神经网络与学习系统汇刊，第 32 卷，第 3 期，第 1162–1176 页，2021。 [33] T. Medini, X. Luan 和 A. Shrivastava，“A4C：预见式异步优势行动者—评论家”，国际学习表征会议，2018。 [34] J. Holliday 和 T. Le，“先跟随后觅食探索：改进异步优势行动者—评论家”，第 107–118 页，2020 年 07 月。 [35] J. Kone˘cn´y, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh 和 D. Bacon，“联邦学习：提升通信效率的策略”，载于 NIPS 隐私多方机器学习研讨会，2016。