DQ-Ladder：一种基于深度强化学习的自适应视频流码率阶梯

Reza Farahani、ZohaAzimi、Vignesh VMenon、Hermann Hellwagner、
Radu Prodan、Schahram Dustdar 和 Christian Timmerer

arXiv:2603.13597v1 [eess.IV] 13 Mar 2026

摘要—**通过 HTTP 对分段视频进行自适应流传输，通常依赖于一组预先定义的码率—分辨率对，即所谓的**码率阶梯。**然而，固定的码率阶梯往往忽略内容和解码复杂度的差异，导致编码时间、解码效率和视频质量之间的权衡并非最优。本文介绍了 DQ-Ladder，这是一种基于深度强化学习（DRL）的方案，用于为自适应视频流应用构建兼顾时间和质量的码率阶梯。DQ-Ladder 将每个分段的预测解码时间、质量分数和码率等级作为深度Q网络（DQN）智能体的输入，并由一个综合解码时间、视频质量和分辨率平滑度的加权奖励函数提供引导。我们利用机器学习模型预测解码时间、码率等级和客观质量指标（VMAF、XPSNR），从而无需进行穷举式编码或质量指标计算。我们使用**通用视频编码**（VVC）工具链（VVenC/VVdeC），在六种符合 Apple HLS 标准的分辨率和**750**个视频序列上，对**41**种量化参数进行了评估。与四种基线方法的实验结果表明，与 HLS 码率梯度相比，DQ-Ladder 的 XPSNR BD-rate降低幅度至少达到**10.3%**，同时解码时间减少**22%**。与其他方法相比，DQ-Ladder 对预测误差的敏感度显著更低，即使噪声高达**20%**，仍能保持稳健。**

索引词—**自适应视频流、深度强化学习、Q学习、码率阶梯、质量预测。**

I. 引言

视频流量主要由视频点播和直播产生，占据全球互联网流量的主导地位，因此对高质量、低延迟且可持续的内容交付的需求日益增长 [1], [2]。在各类流媒体技术中，*HTTP*自适应流（HAS）已成为主流的视频交付机制；它由 MPEG 基于 HTTP 的动态自适应流媒体（DASH） [3] 和 HTTP 实时流媒体（HLS） [4], 标准化。HAS 将视频内容拆分为多个片段，每个片段均以多种比特率和分辨率编码，形成称为表示的不同版本，并构成码率阶梯，存储在 HTTP服务器上。HAS客户端使用自适应比特率算法，从码率阶梯中选择并获取适当表示的各个片段，以优化播放质量 [5]。然而，对高分辨率内容日益增长的需求以及视频编解码器的进步，使码率阶梯中的表示数量不断增加，以适应不同的设备能力 [6]。

Reza Farahani、Zoha Azimi、Hermann Hellwagner 和 Christian Timmerer 任职于奥地利克拉根福大学 ITEC系（firstname.lastname@aau.at）。Vignesh V Menon 任职于德国科特布斯-塞弗特工业大学（电子邮件：Vignesh.Menon@b-tu.de）。Radu Prodan 任职于奥地利因斯布鲁克大学（radu.prodan@uibk.ac.at）。Schahram Dustdar 任职于奥地利维也纳工业大学分布式系统研究部（dustdar@dsg.tuwien.ac.at）。

传统上，码率阶梯构建依赖固定规则，而这些规则忽略了影响视觉质量和编码效率的内容特定特征 [7]。Netflix 的逐标题编码 [6] 标志着向内容感知编码转变：通过分析每个视频的率失真特性，利用视频多方法评估融合（VMAF）等感知质量指标，生成更优的码率阶梯 [8]。在这一范式的基础上，近期的多项研究 [9]–[11] 在目标码率范围内确定分辨率，以进一步提高压缩效率和感知质量。然而，这些方法通常忽略了解码时间，而解码时间对于资源受限的终端用户设备正变得日益关键。因此，如何平衡视觉质量与解码时间仍是码率阶梯构建的一项关键挑战。传统的基于启发式的方法 [12], [13] 会针对每个目标比特率，分别优化分辨率和量化参数（QPs），却忽略了早期决策如何影响后续权衡，例如质量一致性、解码复杂度增长，以及整个码率阶梯中的平滑分辨率转换 [14]。相比之下，深度强化学习（DRL）将码率阶梯构建建模为一个序贯决策问题，从而能够制定全局策略，预见长期的质量与复杂度权衡，并协调整个码率阶梯中的决策。

本文提出 DQ-Ladder，这是一种基于深度强化学习（DRL）的框架，旨在平衡质量、解码时间和分辨率平滑度。该框架利用视频复杂度特征，并在模块化架构中集成轻量级机器学习（ML）预测器，以预测未见视频序列的预期比特率、解码时间和客观质量（通过 VMAF 和 XPSNR 衡量）。这些预测用于引导深度 *Q* 学习（DQN）智能体探索多目标权衡空间。在包含 750 个内容复杂度各异的真实世界视频序列的数据集上进行的大量实验表明，DQ-Ladder 优于四种最先进的方法，BD-rate节省至少达到 10.9 %（PSNR）和 10.3 %（XPSNR），同时将解码时间和编码时间最多分别降低 22 % 和 31 %。DQ-Ladder 对预测误差具有稳健性，即使码率失真和解码时间估计受到高达 20 % 的噪声扰动，仍能保持稳定的质量—解码权衡。

DQ-Ladder 的贡献有四个方面：(*i*) 模块化架构：我们设计了一种基于深度强化学习的架构，利用深度Q网络构建码率阶梯，同时优化质量和解码复杂度。(*ii*) 指标预测：我们利用轻量级机器学习模型预测比特率、解码时间和质量指标，从而能够在线生成适应内容的码率阶梯。(*iii*) 问题建模：我们设计了一种加权奖励

函数，以高效探索比特率–质量–解码时间权衡空间，并纳入约束和惩罚来引导决策。*(iv)* 综合评估：我们使用 750 个真实视频序列和多种编码配置及 VVC 编解码器进行了大量实验，结果表明，在率失真性能、解码时间缩短和编码加速方面，DQ-Ladder 优于四种基线方法。

本文共分为八个章节。第二节回顾相关工作。第三节将码率阶梯构建问题表述为一个马尔可夫决策过程（MDP）。第四节介绍其架构，随后第五节阐述 DQ-Ladder 的执行工作流程。第六节介绍数据集、预测模型和实验设置。第七节分析评估结果。最后，第八节总结本文并讨论未来方向。

II. 相关工作

虽然平均意见分数（MOS）能够提供最准确的感知质量评估，但其成本和主观性使其难以应用于实时场景，因此推动了基于学习的 MOS 预测方法的发展 [15]–[17]。一些研究利用编码参数、网络状况（延迟和丢包）以及用户数据进行 MOS 预测，采用的方法包括决策树（DT） [18], 、多层感知机 [19], 或联邦学习 [20], 。除 MOS 外，PSNR [21],、结构相似性（SSIM） [22], 和 VMAF [8] 也已广泛用于内容感知视频流传输和编码优化。Amirpour 等人 [23] 提出了 VQM4HAS，这是一种回归模型，可利用低成本特征和编码器日志，为 HAS 表示近似计算 VMAF，从而避免大规模显式计算 VMAF。尽管 XPSNR 似乎比 VMAF 对伪影更敏感且计算效率更高 [24]–[26],，但对其进行预测建模仍有待探索。

Kranzler 等人 [27]利用多元线性回归设计了基于比特流的解码能耗预测模型，并将其集成到所提出的编码器中，以实现能效感知的率失真优化。Boujida 等人 [28] 提出了一种基于机器学习的 VVC 逐帧解码时间预测方法，使用极端随机树回归（ETR），并利用从比特流中提取的特征进行预测即编码单元大小、划分深度和运动矢量，从而实现自适应解码控制。Ghasempour 等人 [29]以视频分辨率和帧率为特征对解码能耗建模，并采用轻量级查找表方法，在直播中快速估算能耗。Boujida 等人 [30] 通过加入轻量级时间反馈（以往的解码时间）和特征选择，扩展了其逐帧 VVC 解码时间模型，从而实现准确的多分辨率预测，以支持基于动态电压频率调节（DVFS）的直播自适应。现有研究依赖依赖编解码器的语法或完整比特流解析，因此难以用于低延迟在线场景；其对未见内容及多样化编码设置的泛化能力仍鲜有研究。

Katsenou 等人 [9] 使用基于回归的预测模型，为 HEVC 编码的视频序列构建了内容驱动的码率阶梯。Bhat 等人 [10] 采用随机森林模型预测每个场景的最佳分辨率，而 Mueller 等人 [31]结合卷积神经网络、XGBoost 和多层感知机进行内容复杂度分析。Menon 等人 [32]使用随机森林（RF）模型预测 VMAF，并在一种感知恰可察觉差异 [33] （JND）的编码方案中选择码率-分辨率。Azimi 等人 [12]整合 XGBoost 模型来估算编码时间和质量指标，从而实现解码感知码率阶梯构建。Lebreton 等人 [7][34] 提出了一种基于用户退出概率的阶梯生成方法，以使质量与用户参与度相匹配。Rajendran等人 [35]采用帕累托前沿分析，预测优化后的帧率，以构建节能阶梯。Zhao 等人 [36]提出了一种逐镜头码率阶梯构建方法：对少量分辨率-码率-预设点进行编码，并使用曲线拟合和机器学习模型（如随机森林、支持向量机和神经网络）预测其余率失真曲线。现有方法仅关注质量，忽视了解码复杂度和分辨率平滑度，且需要进行穷尽式预编码。

III. DQ−LA D D E R 模型

我们将码率阶梯构建建模为马尔可夫决策过程 （MDP）。

1）环境：我们将环境定义为视频提供商运营的视频编码系统，其中每个视频被拆分为片段 $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$，并分别以一组目标比特率 $\mathcal{TB} = \{tb_1, tb_2, \ldots, tb_n\}$进行编码。对于每个目标比特率 $tb_i \in \mathcal{TB}$，深度强化学习智能体从预定义集合中选择一个分辨率 $r_i \in \mathcal{R}$ 和量化参数 $qp_i \in \mathcal{QP}$。环境基于轻量级 ̂预测模型，为每个片段提供三个必要的反馈信号：解码时间 $\mathcal{T} = f_1$（E_Y、 h、 L_Y、 $\mathcal{R}$、 $\mathcal{QP}$）、 ̂质量分数 $\mathcal{Q} = f_2$（E_Y、 h、 L_Y、 $\mathcal{R}$、 $\mathcal{QP}$）以及码率等级 ⌢$\mathcal{B} = f_3$（E_Y、 h、 L_Y、 $\mathcal{R}$、 $\mathcal{QP}$），其中 f_1、 f_2 和 f_3 是相应的预测函数。这里， E_Y 表示亮度纹理复杂度， h表示时间复杂度，而 L_Y 表示从视频片段中提取的亮度。

2) 状态空间：我们定义状态空间 $\mathbb{S} = \{st_1, st_2, \ldots, st_k\}$，其总大小为 $|\mathcal{S}| \cdot |\mathcal{TB}|$，涵盖视频片段与目标码率等级的所有可能组合。我们将状态 $st_i \in \mathbb{S}$ 定义为 $st_i = \left[tb_i, \widehat{b}_{i-1}, \widehat{q}_{i-1}, \widehat{t}_{i-1}\right]$，其中包含每个片段当前的目标比特率 tb_i 以及上一个编码步骤 $i-1$的预测结果 ̂ ̂，即比特率 $b_{i-1} \in \mathcal{B}$、质量 $\widehat{q}_{i-1} \in \widehat{\mathcal{Q}}$和解码时间 $\widehat{t}_{i-1} \in \widehat{\mathcal{T}}$，从而使智能体能够推理编码行为，而无需承担实际编码和指标计算的高昂计算成本。

3) 动作空间： 我们定义动作集合$\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$，其中每个 $a_i \in \mathcal{A}$ 都指定了目标比特率 $tb_i \in \mathcal{TB}$ 对应的编码配置，其定义为 $a_i = [r_i, qp_i]$，其中 $r_i \in \mathcal{R}$ 是所选分辨率，$qp_i \in \mathcal{QP}$ 是所选量化参数。总体动作空间 $\mathbb{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_k\}$ 是视频片段 $\mathcal{S}$ 和目标比特率 $\mathcal{TB}$上所有可能动作的并集，代表码率阶梯构建过程中所有可行编码决策的完整集合。

4) 奖励函数：奖励函数通过最大化质量来引导智能体构建高效的码率阶梯，

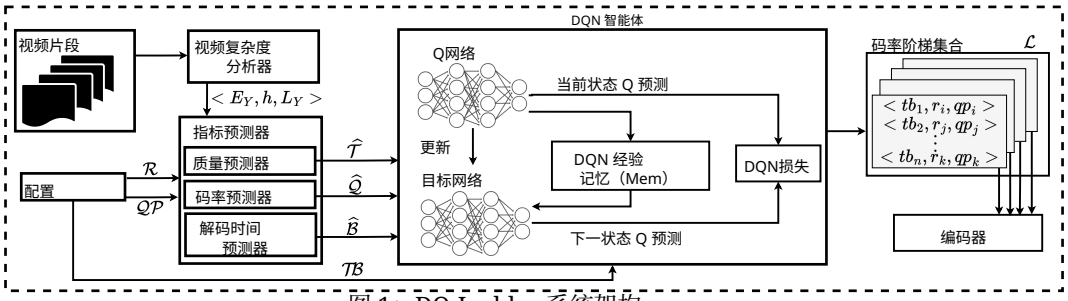


图 1：DQ-Ladder 系统架构。

同时尽量缩短解码时间，并保持不同分段编码之间的视觉稳定性。在为目标比特率为 $tb_i \in \mathcal{TB}$ 的分段 $s_j \in \mathcal{S}$ 选择动作 $a_i = [r_i, qp_i]$ 后，智能体会获得奖励 Rew_i，其公式如下：

$$Rew_i = \lambda_1 \cdot \widehat{\widehat{q}}_i - \lambda_2 \cdot \widehat{\widehat{t}}_i - \lambda_3 \cdot \delta_i. \tag{1}$$

其中，$\widehat{\widehat{q}}_i$ 和 $\widehat{\widehat{t}}_i$ 分别表示归一化预测质量和解码时间，使用最小-最大归一化计算：

$$\widehat{\widehat{q}}_i = \frac{\widehat{q}_i - \widehat{q}_{\min}}{\widehat{q}_{\max} - \widehat{q}_{\min}}, \quad \widehat{\widehat{t}}_i = \frac{\widehat{t}_i - \widehat{t}_{\min}}{\widehat{t}_{\max} - \widehat{t}_{\min}}. \tag{2}$$

其中，$\widehat{q}_{\min}$、$\widehat{q}_{\max}$、$\widehat{t}_{\min}$和 $\widehat{t}_{\max}$ 分别表示候选动作中预测质量和解码时间的最小值与最大值。为惩罚分段之间频繁的分辨率切换，并避免不必要的视觉波动，我们将分辨率变化惩罚 δ_i 定义为：

$$\delta_i = \begin{cases} 1, & \text{if } r_i \neq r_{i-1}; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

其中，$r_{i-1} \in \mathcal{R}$ 是前一个编码步骤 $i-1$中选定的分辨率。惩罚项可确保各片段之间的视觉一致性，避免不必要的分辨率波动。 λ 系数用于控制质量最大化、解码效率和分辨率稳定性之间的权衡，并根据系统偏好通过经验确定。

5）编码约束：我们施加两项基本约束，以确保所构建码率梯度的比特率和质量控制。首先，所选动作的预测比特率不得超过目标比特率阈值，从而确保每个分段的编码受比特率约束：

$$\widehat{b}_i \leq tb_{\max}. \tag{4}$$

其次，我们要求随着目标比特率增加，质量保持单调，以避免更高比特率下的质量下降：

$$tb_i > tb_{i-1} \Rightarrow \widehat{q}_i \geq \widehat{q}_{i-1}. \tag{5}$$

这确保了码率梯度有序，即更高的码率等级始终提供相同或更高的客观质量。

IV. DQ−LA D D E R 架构

我们为 DQ-Ladder 设计了一种模块化架构，以实现高效的视频片段级比特率阶梯构建，如图 1 所示。该架构集成了三个主要模块：*(1)* 视频复杂度分析器、*(2)* 轻量级机器学习指标预测器，以及*(3)* DQN 智能体。给定一组视频片段 $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$，视频复杂度分析会提取每个片段的空间、时间和亮度特征，并考虑基于亮度的特征$\langle E_Y$，h，$L_Y\rangle$。这些特征与候选编码参数（$\mathcal{R}$，$\mathcal{QP}$）一起输入三个独立的预测器 f_1，f_2 和 f_3，用于估算解码时间集合 $\widehat{\mathcal{T}}$、质量集合 $\widehat{\mathcal{Q}}$ 和比特率集合 $\widehat{\mathcal{B}}$，且无需承担实际编码开销。质量预测器与指标无关。尽管我们分别针对 XPSNR 和 VMAF 训练了预测器（第 VI-B 节），但推理时使用的模型会根据服务提供商的优化目标进行选择。例如，若优化目标是 VMAF，则会调用 VMAF 预测器，而 DQN 智能体会最大化与该指标相关的奖励。DQN 智能体与环境交互，通过以下三个主要步骤构建片段特定阶梯：

1）*DQN* 智能体运行：智能体通过神经网络对状态-动作价值函数进行建模，并利用观测到的经验对其进行迭代优化。在每个决策步中，对于给定分段和目标比特率，智能体观测当前状态，其中包含当前目标比特率以及上一个编码步骤的结果。该设计利用学习到的预测实现快速决策，无需进行实际编码计算，同时保留对过往编码行为的感知，以确保稳定性。基于此状态，智能体从动作空间中选择一个动作，指定用于编码该分段的分辨率和量化参数。随后，环境返回所选配置的预测指标，并根据式 1 中定义的多目标奖励函数计算奖励。为学习能够优化长期累积奖励的策略，智能体最小化*DQN*损失，该损失定义为当前状态-动作对的Q网络预测Q值与利用目标网络计算出的目标值之间的均方误差。Q网络参数根据贝尔曼最优性方程（式（6））进行更新：

$$Q(st_i, a_i) \leftarrow Rew_i + \gamma \cdot \max_{a'} Q\left(st_{i+1}, a'\right). \tag{6}$$

其中，a' 表示下一状态 st_{i+1}下的所有可能动作，γ 是控制未来奖励影响的折扣因子。

2）*DQN* 经验记忆：智能体维护一个经验回放池 Mem，以增强学习稳定性，并避免对近期决策过拟合。在状态 st_i 下选择动作$a_i = [r_i, qp_i]$ 并观察到奖励 Rew_i 和下一个

状态 st_{i+1}后，智能体将元组 $(st_i,\ a_i,\ Rew_i,\ st_{i+1})$ 存储在 Mem中。为了更新 Q网络，智能体从 Mem中均匀抽取小批量，从而能够基于 $\mathcal{S}$中不同片段的过往编码经验进行多样化学习。

*3) DQN*输出：对于所有 $tb_i \in \mathcal{TB}$，决策过程都会迭代进行，从而构建分段级码率阶梯 $\mathcal{L}_j = \{(tb_1,\ r_i,\ qp_i)$ $.....(tb_n,\ r_k,\ qp_k)\}$，适用于 s_j。最终输出是覆盖 $\mathcal{S}$ 中所有片段的完整码率阶梯集合 $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_m\}$，随后将其传递给编码器进行最终处理。

V. DQ−LA D D E R 执行工作流程

DQ-Ladder 的运行工作流程包括两个阶段：（*1*）*DQN*训练，在服务提供商侧利用真实编码反馈进行；以及（*2*）*DQN*推理，通过轻量级预测构建码率阶梯。

*1) DQN*训练：算法 1 展示了 DQN 智能体的训练流程，该流程在服务提供商侧的视频数据集上执行，以学习有效的码率阶梯构建策略。智能体与真实编码环境交互，其中每个分段 $s_j \in \mathcal{S}$ 都采用不同的配置（$r_i,\ qp_i$）进行编码，以收集实际的解码时间 t_i、质量 q_i和比特率 b_i值。在每个训练轮次 $e \in \mathcal{EP}$ （第 4 行）以及针对分段 s_j 和目标比特率 $tb_i \in \mathcal{TB}$ 的决策步（第 5–6 行），智能体利用先前的编码结果（第 7 行）观测状态 $st_i = [tb_i, b_{i-1}, q_{i-1}, t_{i-1}]$。智能体使用 ϵ-贪心策略（第 8–11 行）选择动作$a_i = [r_i, qp_i]$，指定分辨率 r_i 和量化参数 qp_i。以 ϵ 的概率选择随机动作来探索新配置；否则，以 $1-\epsilon$ 的概率选择当前 Q值最高的动作进行利用。

所选动作会触发实际编码（第 12 行），由此获得测得的比特率 b_i、质量 q_i 和解码时间t_i，并用于计算奖励 Rew_i （第 13–14 行）。随后观测下一个状态 st_{i+1} （第 15 行），并将经验元组（st_i、a_i、Rew_i、st_{i+1}）存储到经验回放存储器 Mem 中（第 16 行）。接下来，使用从 Mem 中抽取的小批量更新 Q网络参数 θ （第 17 行），以最小化预测Q值与目标Q值之间的时序差分损失（第 18–19 行）。为稳定学习，每隔 C 步（第 20 行），将目标网络参数 θ^- 与 θ 同步一次。接下来，将探索率 ϵ线性衰减至 ϵ_{end} （第 21 行），并将其限制在不低于 ϵ_{end} （第 22 行）。最后，训练会在后续训练轮数中继续进行（第 23 行），直至收敛。最终，输出为经过训练的 Q 函数，其参数为 θ，用于推理（第 24 行）。

我们注意到，DQN训练过程与编解码器无关，也不依赖于特定的奖励权重值。编解码器特有的影响会通过训练期间测得的比特率、质量和解码时间值隐式反映出来，而奖励权重则体现了服务提供商的优化偏好。因此，将 DQ-Ladder 适配到不同的编解码器或服务目标，无需更改学习形式或网络架构，只需使用相应的测量值和更新后的奖励权重重新训练即可。

算法1： DQN训练算法。

输入： 视频片段 $\mathcal{S}$、目标比特率 $\mathcal{TB}$、分辨率 $\mathcal{R}$、量化参数 $\mathcal{QP}$，折扣因子 γ、初始探索率 ϵ_{start}、最终探索率 ϵ_{end}，批量大小 BA、记忆库 Mem、轮次大小 $\mathcal{EP}$
输出： 训练后的 Q网络参数 θ

1 初始化经验回放池 Mem
2 初始化网络 $Q(st,\ a;\ \theta)$ 和目标网络 $Q(st,\ a;\ \theta^-) \leftarrow \theta$
3 初始化 $\epsilon \leftarrow \epsilon_{\text{start}}$
4 **对于** $e \le \mathcal{EP}$ **执行**
5 **对于** $s_j \in \mathcal{S}$ **执行**
6 **对于** $tb_i \in \mathcal{TB}$ **执行**
7 $st_i \leftarrow [tb_i, b_{i-1}, q_{i-1}, t_{i-1}]$
8 **如果** *Rand()* $< \epsilon$ **则**
9 $a_i \leftarrow$ 随机$([r_i, qp_i])$
10 **else**
11 $a_i \leftarrow \arg\max_a Q(st_i,\ a;\ \theta)$
12 执行(a_i)
13 测量$(b_i,\ q_i,\ t_i)$
14 $Rew_i \leftarrow \lambda_1 \cdot \bar{q}_i - \lambda_2 \cdot \bar{t}_i - \lambda_3 \cdot \delta_i$
15 观察(st_{i+1})
16 $Mem \leftarrow (st_i, a_i, Rew_i, st_{i+1})$
17 样本(Mem, BA)
18 $y_i \leftarrow Rew_i + \gamma$ 最大值$_{a'}\ \tilde{Q}(st_{i+1},\ a'; \theta^-)$
19 更新$(\theta, (y_i - Q(st_i, a_i; \theta)))^2$
20 每隔 C 步：$\theta^- \leftarrow \theta$
21 $\epsilon \leftarrow \epsilon_{\text{start}} - (\epsilon_{\text{start}} - \epsilon_{\text{end}}) \cdot \frac{e}{\mathcal{EP}}$
22 $\epsilon \leftarrow$ 最大值$(\epsilon_{\text{end}}, \epsilon)$
23 $e \leftarrow e + 1$
24 返回 $(Q(st, a;\ \theta))$

算法 2： DQN 推理算法。

输入：视频片段 $\mathcal{S}$、目标比特率 $\mathcal{TB}$、分辨率 $\mathcal{R}$、量化参数 $\mathcal{QP}$、已训练的 Q（st、a；θ）、预测器 f_1、f_2、f_3 输出：码率阶梯 $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_m\}$1 对于 $s_j \in \mathcal{S}$，执行以下操作 2 提取（E_Y、h、L_Y）3 对于 $tb_i \in \mathcal{TB}$，执行以下操作

4 $st_i \leftarrow [tb_i, \widehat{b}_{i-1}, \widehat{q}_{i-1}, \widehat{t}_{i-1}]$ 观察 选择 arg
5 $a_i \leftarrow {}_a\ Q\ st_i\ a\ \theta$
max（，；）6（$\widehat{t}_i$、$\widehat{q}_i$、$\widehat{b}_i$）←（f_1、f_2、f_3）7 $\mathcal{L}_j \leftarrow$（tb_i、r_i、qp_i）8 返回（$\mathcal{L}_j$）

2) DQN 推理：算法 2 展示了 DQN 在线推理过程。训练完成的 DQN 智能体无需进行实际编码，而是利用轻量级内容特征和基于机器学习的预测模型。对于每个分段 $s_j \in \mathcal{S}$（第 1 行），视频复杂度分析器会提取内容特征 $\langle E_Y, h, L_Y \rangle$（第 2 行）。对于每个目标比特率 $tb_i \in \mathcal{TB}$（第 3 行），智能体会使用目标比特率和上一步的预测指标来观测当前状态 $st_i = [tb_i, \widehat{b}_{i-1}, \widehat{q}_{i-1}, \widehat{t}_{i-1}]$（第 4 行）。随后，通过最大化已学习的 Q 函数来选择动作 $a_i = [r_i, qp_i]$（第 5 行）。环境利用预测器 f_1, f_2，以及返回预测结果（，$\widehat{q}_i, \widehat{t}_i$）（第 6 行），从而避免成本高昂的实际编码。此过程会遍历所有目标比特率，以构建针对该分段的码率阶梯 $\mathcal{L}_j$ （第 8 行）。最后返回相应的码率阶梯 $\mathcal{L}_j$ （第 8 行），供编码使用。

3）计算复杂度分析：离线 DQN 训练（算法 1）的时间复杂度为 $\mathcal{O}(\mathcal{EP} \cdot |\mathcal{S}| \cdot |\mathcal{TB}| \cdot (C_{enc} + BA \cdot C_{NN}))$，其中 $\mathcal{EP}$ 为训练轮数，$|\mathcal{S}|$ 为视频片段数，$|\mathcal{TB}|$ 为目标比特率数，C_{enc} 为每个动作的实际编码成本（离线执行），BA 为小批量大小，C_{NN} 为神经网络前向和反向传播的计算成本。在线推理（算法 2）的复杂度为 $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{TB}| \cdot (C_{NN} + C_{pred}))$，其中 C_{pred} 为轻量级预测模型 f_1、f_2、f_3的最坏运行时间。

表 I：编码与解码配置参数。

参数	配置
$\mathcal{R}$	$\{360, 540, 720, 1080, 1440, 2160\}$p
$\mathcal{QP}$	$\{10, 11, 12, \ldots, 50\}$
$\mathcal{TB}$	$\{145, 300, 600, 900, 1600, 2400, 3400, 4500, 5800, 8100, 11600, 16800\}$ 千比特每秒
编码器	VVenC ，4 个 CPU线程
解码器	VVdeC ，4 个 CPU线程

VI. 评估设置

本节详细介绍 DQ-Ladder 的评估设置。我们所有实验均在一台配备 128 核英特尔至强金牌处理器和两块 NVIDIA Quadro GV100 GPU 的服务器上进行。

A. 数据集说明与分析

我们使用了 Inter-4K 数据集中的 750 个超高清（UHD）视频序列 [37]，涵盖自然、城市、体育和合成场景等多种内容类型。每个序列均使用 VVenC v1.11 [38] 的 faster 预设以 60fps 编码。如表 I 所示，编码设置涵盖六种分辨率 $\mathcal{R} = \{360, 540, 720, 1080, 1440, 2160\}$p 和 41 个量化参数 $\mathcal{QP} = \{10, \ldots, 50\}$。对于每个编码版本，我们记录了实际比特率 b_i、编码时间以及使用 VVdeC v2.3.0 [39] 的解码时间 t_i。使用 XPSNR 和 VMAF 测量客观质量分数 q_i，为训练 DQN 智能体提供真实标签。使用 VCA v2.0 [40] 提取时空复杂度特征。我们分析了数据集特征，以突出内容的多样性。图 2 展示了视频序列在 (E_Y, h) 特征空间中的分布，体现了空间和时间复杂度的多样性，颜色表示亮度变化 L_Y。

B. 基于机器学习的指标预测模型

鉴于实际编码、解码和质量测量的开销较高，DQ-Ladder 采用轻量级的基于机器学习的预测器来预测解码时间、质量和比特率，从而能够在在线推理期间快速构建码率阶梯（第 V 节）。我们评估了七种知名机器学习模型，涵盖四个不同类别：*a)* 线性回归（LR）[41]，作为简单的线性模型；*b)* 随机森林（RF）[42]和极端随机树回归（ETR）[43]，作为基于树的集成模型；*c) AdaBoost* [44]，、*LightGBM* [45],和*XGBoost* [46]，作为基于梯度提升的集成模型；*d)* 多层感知机（MLP）[47]，作为具有全连接层的神经网络。

我们在视频级别将数据集划分为互不重叠的训练集（70%）和测试集（30%），确保给定视频的所有片段完全归属于其中一个集合，以避免数据泄漏。我们使用固定随机种子（种子 = 42）对视频标识符进行随机打乱，以确保三个预测任务中的结果可复现且划分一致。我们在训练集上使用 Scikit-learn [49] 中的 Grid-SearchCV [48] 进行五折交叉验证，以搜索基于集成模型的超参数。我们在预定义的参数空间内进行穷举网格搜索，并使用训练数据重新训练表现最佳的估计器。对于 MLP 模型，我们将训练集的 10% 用作验证集，并应用 *Optuna* [50]进行超参数优化。

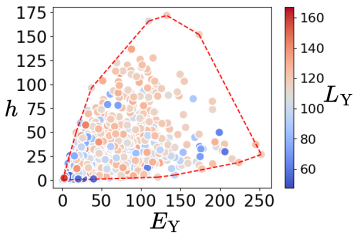


图 2：数据集中内容复杂度的特征分布。

我们根据每个预测器的设计灵活性，进行了针对模型的超参数调优。LR 模型没有可调参数，而基于集成的模型则需要优化估计器数量（n_{trees}）、最大树深度（$d_{\max}$）和学习率（η）。对于 MLP 模型，我们对隐藏单元数量（h_{units}）、学习率（η）和批量大小（bs）进行了微调。表 II 总结了各模型所探索的超参数搜索空间。在最终评估中，我们针对每个目标指标（解码时间、比特率、XPSNR、VMAF）采用表现最佳的模型，并在第 VII-A 节报告相应的超参数值。

C. DQN 智能体配置和超参数

为近似 Q 值函数，我们采用一个多层感知机（MLP），以四维状态向量 s_t^i 作为输入。所有状态特征均使用最小-最大缩放归一化到范围 $[0, 1]$，以提高稳定性和收敛性。网络由三个相连的隐藏层组成，分别包含 256、128 和 64 个神经元，并使用 ReLU 激活函数 [51]。输出层是大小为 $|\mathcal{A}|$ 的线性全连接层，其中每个神经元代表一个离散动作对（r_i, qp_i）的 Q 值。该模型使用 PyTorch 实现，采用 Kaiming（He）初始化 [52]，并使用 Adam [53]进行优化。DQN 智能体通过经验回放训练 $\mathcal{EP} = 2000$ 个回合，记忆库大小为$|Mem| = 10000$，小批量大小为 $BA = 128$。目标网络每隔 $C = 10$ 步同步一次。我们采用 ϵ-贪心探索策略，ϵ 在 2000 步内从 0.8 线性衰减至 $\epsilon_{\min} = 0.05$。超参数通过结合基于 Optuna 的优化 [50]和手动调整进行调优。学习率 $\eta \in \{0.05, 0.005, 0.0005\}$（取决于实验）和折扣因子 $\gamma = 0.7$。表 III 总结了最终的 DQN 智能体超参数。

D. 评估指标

我们使用不同的评估指标来评估 DQ-Ladder 的性能，并将其与基线方法进行比较。

1）预测模型指标：我们使用以下标准回归指标来评估预测模型的准确性和泛化性能：*a)* 决定系数 (R^2)；*b)* 平均绝对误差百分比 (MAE%)；*c)* 均方根误差 (RMSE)；*d)* 绝对误差标准差 (SDAE)；*e)* 推理时间。

2）*DQN* 智能体收敛性分析：我们使用累积奖励变化评估智能体的学习行为。该指标汇总训练步骤中每个回合的奖励，以反映整体策略表现；值越高，表示学习进展越好。

表 II：机器学习预测模型的超参数搜索空间。

预测模型	探索的超参数
LR	None
随机森林、ETR	$n_{trees} \in \{50, 100, 200, 300\}$, $d_{max} \in \{3, 5, 7, 10\}$
AdaBoost、LightGBM	$n_{trees} \in \{50, 100, 200, 300\}$ $\eta \in \{0.01, 0.1, 0.5\}$
XGBoost	$n_{trees} \in \{50, 100, 200, 300\}$ $d_{max} \in \{3, 5, 7, 10\}$ $\eta \in \{0.01, 0.1, 0.5\}$
MLP	$h_{units} \in \{64, 128, 256, 512\}$ $\eta \in [10^{-5}, 10^{-2}]$ $bs \in \{32, 64, 128, 256\}$

表 III：DQN 智能体超参数。

超参数	值
学习率（η）	0.05、0.005、0.0005
折扣因子 (γ)	0.7
批量大小 (BA)	128
经验回放缓冲区大小 (Mem)	10,000
目标网络更新间隔 (C)	10
初始探索率 (ϵ_{start})	0.8
最终探索率（ϵ_{end}）	0.05
探索衰减计划	线性衰减
训练回合数（$\mathcal{EP}$）	2,000

表 IV：HLS 码率梯度。

比特率（千比特每秒）	分辨率
145	360
300	360
600	540
900	540
1600	540
2400	720
3400	720
4500	1080
5800	1080
8100	1440
11 600	2160
16 800	2160

3）码率梯度效率指标：我们使用以下指标评估所构建码率阶梯在质量与效率之间的权衡：*a*）*Bjøntegaard* 增量码率（BD-rate） [54] 用于衡量达到相同质量（XPSNR 或 VMAF）所需的比特率增幅；BD 码率越低（负值越大），编码效率越高。*b*）*Bjøntegaard* 增量质量（BD-PSNR、BD-XPSNR、BD-VMAF）表示在一定比特率范围内的平均质量差异。正值越高，表示在相同比特率下编码效率越高。*c*）*Bjøntegaard* 增量编码时间（BD-EnTime）量化达到与基准相同质量或比特率所需编码时间的平均变化；值越低，编码速度越快。*d*）*Bjøntegaard* 增量解码时间（BD-DeTime）衡量达到与基准相同质量或比特率所需解码时间的平均变化；值越低，解码速度越快。*e*）分辨率切换表示连续分辨率之间绝对差值的平均值；值越低，视觉过渡越平滑。

E. 用于比较的方法

我们将 DQ-Ladder 与四种基线方法进行比较：*a)HLS [55]*采用苹果规范（表 IV）定义的固定码率-分辨率阶梯。*b)CDBL [12]*选择满足给定解码时间约束且解码时间低于阈值 τ_L的编码配置中 XPSNR 最高的一种。*c)RQT-PF [13]*通过权衡解码时间和比特率来最小化复合成本函数，并由权衡参数 α 控制，但不直接优化 XPSNR 或 VMAF 等视频质量指标。*d) VEXUS [56]* 针对比特率-XPSNR 权衡优化码率阶梯，在不考虑解码复杂度的情况下，最大化质量并最小化比特率。

VII. 评估结果

本节评估预测模型、DQN 智能体的行为，以及与四种基线方法相比的码率阶梯效率。

A. DQ-Ladder 预测模型分析

我们评估了候选预测模型在四项目标指标上的表现：解码时间、比特率、XPSNR 和 VMAF。报告的结果代表模型在整个测试集上的平均表现；然而，推理时间仅适用于在线码率阶梯构建期间的每个分段，因此其开销可以忽略不计。每项目标指标都具有不同特征，会影响模型的适用性。例如，解码时间呈现出规律性模式，因此更适合 RF 等基于树的模型。与此同时，比特率和质量分数（即 VMAF、XPSNR）与编码参数之间的关系更为复杂且呈非线性，因此梯度提升和基于神经网络的模型（多层感知机）表现出更高的准确度。我们使用第 VI-D 节中的评估指标评估模型，并综合考虑预测准确度和推理效率，为每个目标选出表现最佳的模型。

1) 解码时间预测：表 VI 显示，随机森林取得了最高的 R^2 (0.96)和最低的均方根误差（3.9），其表现与多层感知机相近。不过，随机森林的平均绝对误差百分比更低（11.49），推理时间也更短（0.31 秒，而多层感知机为 0.7 秒），因此成为解码时间预测的首选模型。随机森林在使用 $n_{trees} = 100$ 和 $d_{\max} = 10$时表现最佳。

2) 比特率预测：表 V 显示，多层感知机取得了最高的 R^2 (0.91)，而 XGBoost 的绝对误差指标更优（均方根误差为 4651，而多层感知机为 6091；平均绝对误差百分比为 11.17，而多层感知机为 22.84），推理时间也更短（0.6 秒，而多层感知机为 0.8 秒）。因此，鉴于 XGBoost 在准确性和效率之间实现了更好的权衡，我们选择它进行比特率预测。XGBoost 在使用 $n_{trees} = 300$、 $d_{\max} = 7$和 $\eta = 0.5$时表现最佳。

3）*XPSNR* 预测：表 VII 显示，多层感知机再次取得了最低的均方根误差（1.16）和平均绝对误差百分比（2.39）。尽管 XGBoost 和 LightGBM 的性能与多层感知机相近，但 XGBoost 的推理时间更短（0.44 秒，相比之下分别为 0.56 秒和 0.92 秒），因此是更优的模型。XGBoost 模型在使用 $n_{trees} = 300$、$d_{\max} = 7$ 和 $\eta = 0.1$ 时性能最佳。

4）*VMAF* 预测：如表 VIII 所示，多层感知机在所有指标上均取得最佳结果， R^2 为 0.94，均方根误差为 5.99，平均绝对误差百分比为 6.92。与 XPSNR 的情况类似，XGBoost 和 LightGBM 的性能与多层感知机相近。由于推理时间较短（0.42 秒），我们选择 XGBoost 进行 VMAF 预测。XGBoost 在使用 $n_{trees} = 300$、 $d_{\max} = 5$ 和 $\eta = 0.5$ 时性能最佳。

5）特征重要性分析：我们采用 *SHapley* 加性解释 (SHAP) [57] 来展示各输入特征对预测目标的贡献。SHAP 值反映特征对模型输出的平均影响，而 SHAP 交互值则揭示特征对之间的依赖关系。图 3 展示了 SHAP 值绝对值的均值和交互热图。在所有预测指标中，量化参数 ($\mathcal{QP}$) 和分辨率 ($\mathcal{R}$) 是影响模型预测的主要特征。空间

表 V：比特率预测结果。

模型	R^2	*RMSE*	*SDAE*	*MAE%*	推理时间
XGBoost	**0.89**	**4651.06**	**4513.09**	**11.17**	**0.6**
LightGBM	0.89	6971.99	6489.9	24.8	0.53
LR	0.82	8816.12	8156.82	32.6	0.003
RF	0.87	7390.21	6862.91	26.71	1.84
极端随机树	0.86	7722.34	7185.67	27.56	0.32
AdaBoost	0.71	11295.59	10617.74	37.55	0.32
MLP	0.91	6091.83	5631.26	22.84	0.8

表 VI：解码时间预测结果。

模型	R^2	*RMSE*	*SDAE*	*MAE%*	推理时间
XGBoost	0.94	4.48	3.96	12.46	0.5
LightGBM	0.95	4.2	3.7	12.02	0.56
LR	0.75	9.43	7.14	36.8	0.002
RF	**0.96**	**3.9**	**3.41**	**11.49**	**0.31**
极端随机树	0.94	4.58	4.02	13.04	0.08
AdaBoost	0.85	7.37	4.08	36.64	0.33
MLP	0.96	3.87	3.3	12.05	0.7

表 VII：XPSNR 预测结果。

模型	R^2	*RMSE*	*SDAE*	*MAE%*	推理时间
XGBoost	**0.96**	**1.22**	**0.79**	**2.51**	**0.44**
LightGBM	0.96	1.23	0.79	2.53	0.56
LR	0.86	2.2	1.4	4.53	0.002
RF	0.94	1.42	0.9	2.97	1.55
极端随机树	0.94	1.45	0.95	2.94	0.34
AdaBoost	0.92	1.62	1.001	3.42	0.32
MLP	0.96	1.16	0.74	2.39	0.92

表 VIII：VMAF 预测结果。

模型	R^2	*RMSE*	*SDAE*	*MAE%*	推理时间
XGBoost	**0.93**	**6.38**	**4.44**	**6.74**	**0.42**
LightGBM	0.93	6.45	4.52	6.77	0.56
LR	0.75	11.74	6.75	14.13	0.002
RF	0.93	6.22	4.33	6.57	1.57
极端随机树	0.92	6.82	4.76	7.17	0.34
AdaBoost	0.85	9.22	4.6	11.75	0.32
MLP	0.94	5.99	3.69	6.92	0.84

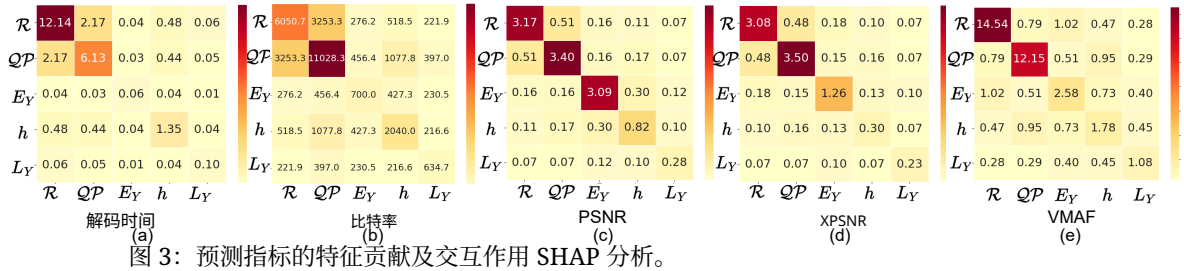


图 3：预测指标的特征贡献及交互作用 SHAP 分析。

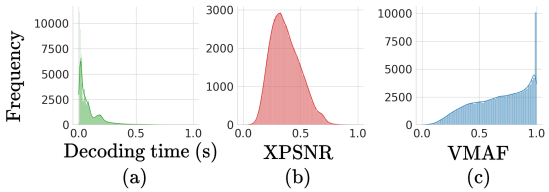


图 4：归一化解码时间和质量指标（XPSNR、VMAF）的分布。

复杂度 (E_Y) 对两项质量指标也都发挥着关键作用。相比之下，时间复杂度 (h) 与运动成本相关，因此对比特率和解码时间的影响更大。此外，最强的特征交互出现在 $\mathcal{QP}$ 和 $\mathcal{R}$之间，尤其是在解码时间和比特率预测方面，反映了它们对压缩和计算复杂度的影响。亮度变化 (L_Y) 对所有指标的贡献最小。

B. DQ-Ladder 智能体分析

我们通过本节介绍的四项实验来分析 DQN 智能体的行为。

1）奖励归一化与加权策略：我们分析了解码时间和质量指标的归一化分布，以精确指导奖励函数（公式 1）。如图 4 所示，即使经过归一化，$[0,1]$，各分布仍存在显著差异：解码时间呈明显右偏（偏度为=2.4），XPSNR 呈中度右偏（0.49），而 VMAF 呈轻微左偏（-0.42），这表明等权重 λ_1 和 λ_2 无法平衡奖励函数中的贡献。因此，我们放宽了约束$\lambda_1+\lambda_2+\lambda_3=1$，允许独立调节，从而在质量与解码效率之间实现更加灵活、数据驱动的权衡。

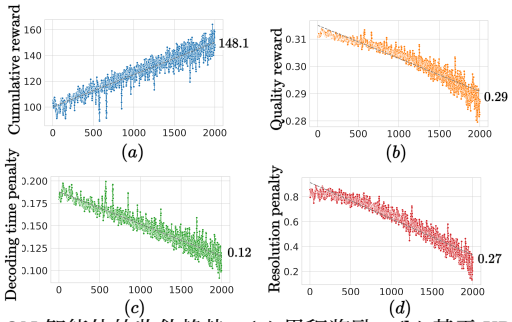


图 5：DQN 智能体的收敛趋势：(a) 累积奖励，(b) 基于 XPSNR 的质量奖励，(c) 解码时间惩罚，以及 (d) 分辨率平滑度惩罚。

2）累积奖励与收敛性分析：我们监测了 2000 个训练回合数，以评估 DQN 智能体的收敛行为。如图 5（a）所示，累积奖励稳步增加，证实了策略持续改进。XPSNR 奖励在图 5（b）中略微下降了 7.4%，表明智能体的策略转向降低时间复杂度。解码时间和分辨率平滑度惩罚分别下降了 36.1%（图 5（c））和 65.8%（图 5（d）），说明智能体学会了在质量与效率之间进行平衡。

3）惩罚影响：我们评估了两种体现特定提供商的示例策略的分辨率切换惩罚策略：（*i*）每当所选分辨率与前一次不同时施加固定惩罚（公式 3），以一致地抑制变化并优先保持平稳切换；（*ii*）当新分辨率偏离最近两次选择时触发历史感知惩罚，并根据归一化分辨率差异调整惩罚幅度，从而实现灵活且受控的自适应。为单独考察两者的影响，我们设定 $\lambda_1=0$，$\lambda_2=0$，$\lambda_3=1$，仅将分辨率切换最小化作为优化目标。图 6（c）和（f）显示，固定惩罚策略（蓝

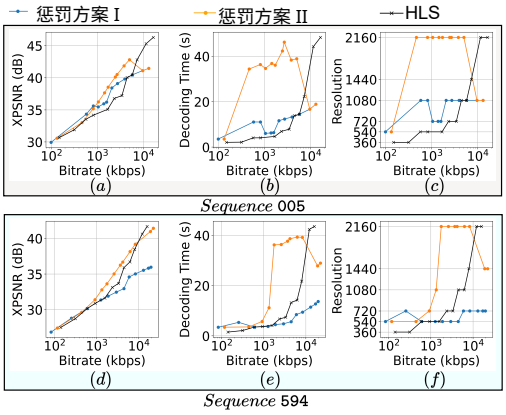


图 6：不同惩罚对 DQ-Ladder 的影响，在序列 005（a–c）和 594（d–e）上进行评估。

表 IX：DQ-Ladder 带来的平均延迟。

粒度	*VCA* *(s)*	指标预测 *(ms)*				*DQN* 智能体 *(ms)*	总计 *(s)*
		质量比特率			*Dec. time*		
		XPSNR	*VMAF*				
Rep.	2.71	0.12	0.23	0.16	0.16	0.3	2.71
Seg.		1.4	2.7	1.95	1.87	3.6	2.72

线）产生的码率阶梯较为保守，切换次数极少；而历史感知惩罚策略（橙线）则允许分辨率更平稳地提升。这些方案体现了稳定性与适应性之间固有的权衡，使服务提供商能够根据特定应用的需求调整切换行为。

4）系统开销：我们分析了 DQ-Ladder 引入的端到端延迟，以评估其在可扩展场景中的适用性。表 IX 报告了每种表示（rep.）和每个 4 秒分段（seg.）的平均延迟，其中包括三个主要组成部分：VCA、指标预测和 DQN 智能体决策。由于预测模型并行运行，因此可以将三个预测器中最长的推理时间作为参考。结果表明，DQ-Ladder 对每种表示造成的延迟可以忽略不计（低于毫秒级），而每个分段的完整码率阶梯构建延迟低于一秒，证实了其用于实时比特率选择的可行性。

C. DQ-Ladder 加权系数分析

我们通过以下实验分析不同加权系数对 DQ-Ladder 行为的影响。

1) 对码率阶梯性能的影响：我们调整了λ_1–λ_3 的值，以分析不同奖励权重的影响。如图 7 所示，设为 (1,0,0) 时质量最高（图 7 (a)），PSNR 和 XPSNR 分别比 HLS 提高 1.26 dB 和 0.96 dB（图 7 (b)），但编码时间增加 42.7 s，解码时间增加 14.4 s（图 7 (c)）。相比之下，设为 $(0,1,0)$ 时，编码和解码时间分别减少 12.9 s 和 2.1 s，但质量损失约 2 dB，且分辨率变化不稳定（图 7 (d)）。为平衡这些权衡，我们评估了其他配置。其中，$(0.8,0.6,0.1)$ 的权衡效果最佳：BD 码率降低 10.94 %（PSNR）和 10.3 %（XPSNR），编码和解码时间分别节省 7.96 s 和 0.72 s，分辨率平滑度也有所提升，同时质量没有下降。

2）对 *VMAF* 的影响：从针对 XPSNR 优化的初始权重配置出发，我们调整了 λ_1–λ_3 的数值，以分析不同奖励权重对 VMAF 提升的影响。在图 8（a）中，λ_1 和 λ_2较高的配置（例如 (0.7,0.8,0.1) 和 $(0.8,0.8,0.1)$）实现了最低的 BDR_V，表明其在 VMAF 方面具有更高的比特率效率。图 8（b）显示，采用这些设置时，BD-VMAF 提升约 5–6%，而 BD-PSNR 几乎保持不变，证实了质量有所提升。在复杂度方面（图 8（c）），与 $(0.8,0.8,0.1)$ 相比，$(0.7,0.8,0.1)$ 缩短了编码和解码时间，同时保持了效率。此外，分辨率平滑度（图 8（d））也有所提升，其中 $(0.7,0.8,0.1)$ 将分辨率切换造成的伪影降至最低。根据这些结果，我们选择 $(\lambda_1 = 0.7, \lambda_2 = 0.7, \lambda_3 = 0.1)$ 作为权重配置，取得了 5 个点的 BD 视频多方法评估融合增益、36% 的比特率降低和 -0.71 的 BD-DeTime，同时实现了平滑的分辨率转换，且与 HLS 相比没有牺牲质量。

D. DQ-Ladder 与基线方法分析

我们在选定的配置（$\lambda_1 = 0.8$，$\lambda_2 = 0.6$，$\lambda_3 = 0.1$）下，将 DQ-Ladder 的性能与四种最先进的码率阶梯构建方案进行了比较。

1）对码率阶梯性能的影响：图 9 比较了 DQ-Ladder 与 RQT-PF（使用 $\alpha = 0.75$）、CDBL（使用 $\tau_L = 16$）和 VEXUS 的平均性能。VEXUS 的 BD-PSNR（1.3 dB）和 BD-XPSNR（+1.05dB）最高，但代价是编码时间（32.2s）和解码时间（10.7 s）增加。DQ-Ladder 实现了均衡，其 BD 码率为 -10.94%（PSNR）和 -10.3%（XPSNR），同时使 BD-PSNR 提高了 0.19 dB，BD-XPSNR 提高了 0.18 dB。与 RQT-PF 相比，DQ-Ladder 在编码和解码时长几乎相同的情况下，降低了比特率开销并提升了质量。与 CDBL 相比，DQ-Ladder 仅付出很小的比特率代价，就节省了 6.1 s 的编码时间和 0.3 s 的解码时间。此外，DQ-Ladder 的分辨率平滑度得分最低（25.5），其分辨率切换次数少于所有基线方法。

2）对预测误差鲁棒性的影响：像 CDBL 这样的阈值优化方法会直接根据预测的解码时间和值做出码率阶梯决策，因此当预测值跨越决策边界时，容易受到估计噪声的影响。为量化这一影响，我们按照既有评估实践 [58], [59]，向预测的解码时间和 XPSNR 指标中注入最高达 20% 的加性高斯噪声。如图 10(a–c) 所示，CDBL 在预测值受噪声干扰时表现出明显的不稳定性，其 BD 码率从基础预测条件下的 -19.26% 恶化至噪声为 10% 时的 -2.04 %，并在噪声达到 20% 时转为正值（2.10 %）。此外，其 BD-PSNR 和 BD-XPSNR 同时降至零以下，表明压缩效率和质量增益均有所损失。相比之下，DQ-Ladder 在所有噪声水平下均保持稳健（图 10(d–f)）。即使噪声达到 20 %，DQ-Ladder 的 BD 码率和 BD 质量指标仍接近基线设置，编码和解码时间仅略有增加。

3）对视频序列的影响：图11展示了两个复杂度特征各异的代表性视频片段的码率阶梯。VEXUS 在这两个片段中均取得了最高的 XPSNR，但解码时间也更长。

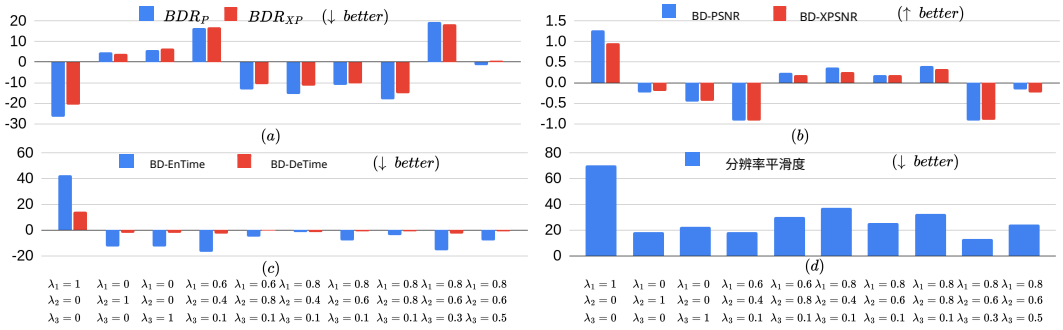
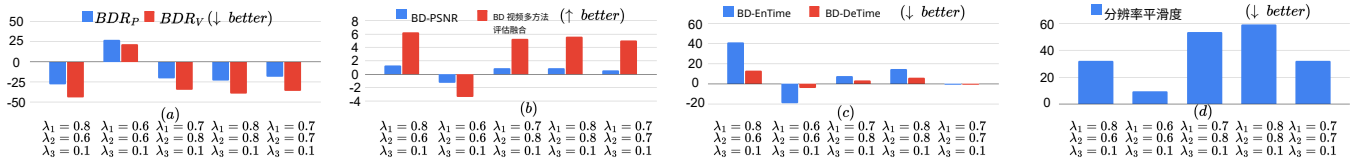


图 7：不同 λ_1、 λ_2 和 λ_3 值对 DQ-Ladder 性能结果的影响。



Fi图 8：改变 λ_1、 λ_2 和 λ_3 值对 DQ-Ladder 在 VMAF 指标下性能的影响。

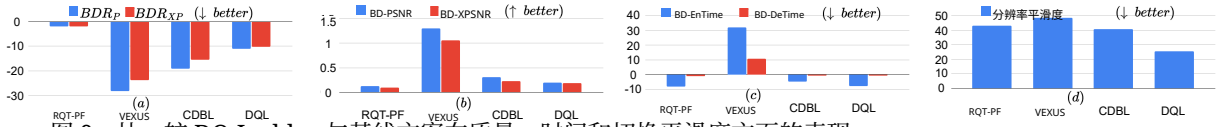


图 9：比 较 DQ-Ladder 与基线方案在质量、时间和切换平滑度方面的表现 s.

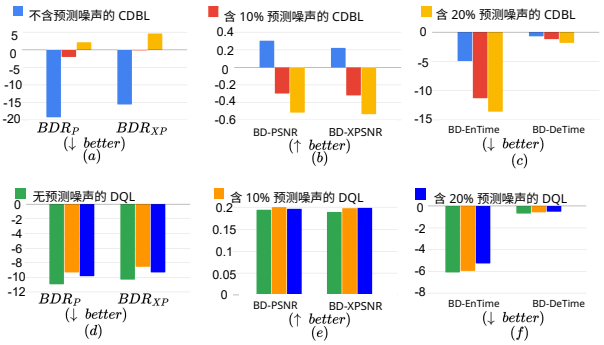


图 10：随着预测噪声增加，CDBL 和 DQ-Ladder 的鲁棒性。

DQ-Ladder 中的分辨率平滑度惩罚使 XPSNR 和解码时间沿码率阶梯呈现更稳定的变化趋势。相比之下，RQT-PF 和 CDBL 频繁切换分辨率，导致这两项指标出现明显波动。因此，基于 DQ-Ladder 的码率阶梯呈现出更连贯的变化，在保持较高 XPSNR 的同时，将解码时间维持在较低水平。

VIII. 结论

本文提出了 DQ-Ladder，一种基于深度 Q 学习（DQL）的框架，通过联合考虑视频质量、解码时间和分辨率稳定性来构建自适应码率阶梯。我们为 DQ-Ladder 设计了两阶段方案：1）训练阶段，使用真实编码反馈和兼顾视频质量、解码时间与分辨率稳定性的多目标奖励对智能体进行优化；2）推理阶段，利用轻量级内容特征和预测模型，高效地完成码率

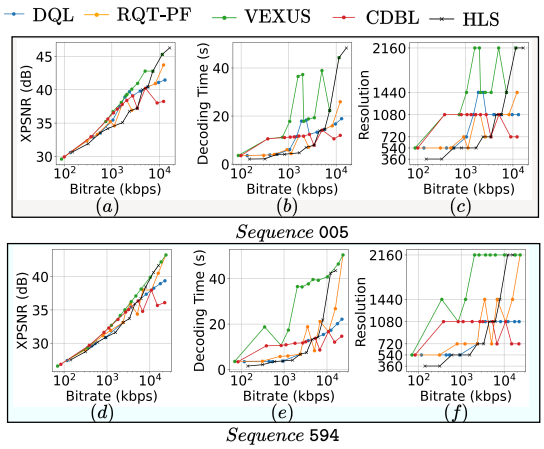


图 11：在视频序列 005（a-c）和 594（d-f）上，DQ-Ladder 与 RQT-PF、VEXUS 和 CDBL 的比较。

阶梯构建，而无需调用实际编码器。在 750 个多样化的真实世界视频序列上进行的大规模评估表明，DQ-Ladder 的性能优于现有最先进方法。未来的工作将重点拓展 DQ-Ladder，使其能够纳入网络动态和客户端反馈，以实现端到端自适应流媒体优化。

参考文献

[1] K. Rosenthal，《2025 年全球互联网现象报告》，2025年，获取日期：2026 年 2 月 6 日。 [在线]。可访问：https://www.a pplogicnetworks.com/blog/the-2025-global-internet-phenomena-report

[2] A. Bentaleb 等，《迈向一秒延迟：直播媒体流的发展》，*IEEE*《通信调查与教程》，2025年。[3] T. Stockhammer，“基于 HTTP 的动态自适应流媒体——标准与设计原则”，载于第二届 *ACM* 多媒体系统年度会议论文集，2011年。

[4] R. Pantos 和 W. May，“HTTP 实时流媒体”，RFC 8216，2017 年，获取日期：2026 年 2 月 6 日。 [在线]。可访问：https://rfc-editor.org/rfc/ rfc8216.txt[5] A. Bentaleb 等，“BoB：面向实时通信的启发式与强化学习带宽预测”，IEEE多媒体汇刊，2022年。[6] J. De Cock 等，“云端基于复杂度的一致质量编码”，载于 IEEE 国际图像处理会议，2016年。[7] P. Lebreton 和 K. Yamagishi，“基于退出率的自适应比特率视频流媒体码率阶梯选择机制”，IEEE多媒体汇刊，2023年。[8] Z. Li 等，“VMAF：旅程仍在继续”，载于 Netflix技术博客，2018年。[9] A. V. Katsenou 等，“面向自适应视频流的内容无关码率阶梯预测”，载于图像编码研讨会。IEEE，2019年。[10] M. Bhat 等，“结合视频质量指标，在宽质量范围内选择感知准确的分辨率：案例研究”，载于2021年 IEEE 国际图像处理会议。IEEE，2021年。[11] A. Premkumar 等，“面向自适应视频流的质量感知动态分辨率自适应框架”，载于第15届 ACM 多媒体系统会议论文集，2024年。[12] Z. Azimi 等，“面向自适应 VVC 流媒体的解码复杂度感知码率阶梯估计”，载于第32届欧洲信号处理会议，2024年。[13] A. Katsenou 等，“面向自适应 VVC 流媒体的解码复杂度—码率—质量帕累托前沿”，载于 IEEE 国际视觉通信与图像处理会议。IEEE，2024年。[14] A. Telili 等，“自适应视频流的码率阶梯预测方法：综述与基准测试”，arXiv预印本 arXiv:2310.15163，2023年。[15] Y. Li 等，“人脸视频压缩的感知质量评估：基准测试与有效方法”，IEEE多媒体汇刊，2024年。[16] W. Zou 等，“基于事件的感知质量评估：适用于存在播放中断的基于 HTTP 的视频流媒体”，IEEE多媒体汇刊，2017年。[17] D. S. Hands，“基础多媒体质量模型”，IEEE多媒体汇刊，2004年。[18] A. Hameed 等，“基于决策树的感知视频质量预测模型及其在无线多媒体通信前向纠错中的应用”，载于 IEEE 多媒体汇刊。IEEE，2016年。[19] T. Begluk 等，“LTE网络视频流的基于机器学习的 QoE预测”，载于第17届国际研讨会 INFOTEH-JAHORINA。IEEE，2018年。[20] S. Ickin 等，“利用分布式深度学习对拆分特征进行QoE建模”，Network，2021年。[21] 电信行业解决方案联盟，“使用峰值信噪比全参考技术进行客观视频质量测量”，载于 T1.TR.74-2001，2001年。[22] Z. Wang 等，“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，2004年。[23] H. Amirpour、J. Zhu、W. Zhou、P. Le Callet 和 C. Timmerer，“VQM4HAS：用于 HTTP自适应流中 HEVC 视频的实时质量指标”，IEEE多媒体汇刊，2025年。[24] M. Wien 和 V. Baroncini，“SDR UHD随机访问类别中的 VVC压缩性能验证测试报告”，载于 WG 05 MPEG 联合视频编码组与 ITU-T SG 16，文档 JVET-T0097，2020年。[25] C. R. Helmrich 等，“不同分辨率和位深下视频压缩扩展感知加权峰值信噪比的研究”，载于 ITU期刊：ICT Discoveries，2020年。[26] C. R. Helmrich 等，“SDR UHD验证测试中 VVC编码器的信息与分析”，载于 WG 05 MPEG 联合视频编码组与 ITU-T SG 16，文档 JVET-T0103，2020年。[27] M. Kr¨anzler 等，“通用视频编码的解码能耗建模”，载于 IEEE 国际图像处理会议。IEEE，2020年。[28] H. Boujida 等，“通用视频编码的解码时间预测”，载于 IEEE 第25届多媒体信号处理国际研讨会。IEEE，2023年。[29] M. Ghasempour 等，“面向实时视频流的实时质量与能耗感知码率阶梯构建”，IEEE电路与系统新兴及精选主题期刊，2025年。[30] P.-L. Cabarat 等，“低功耗 VVC 软件解码的图像解码时间估计”，载于第32届欧洲信号处理会议。IEEE

[31] C. Mueller 等，《将上下文感知视频编码作为基于网络的媒体处理工作流》，载于第 13 届 ACM多媒体系统会议论文集，2022年。[32] V. V. Menon 等，《面向自适应直播流媒体的 JND 感知逐标题两遍编码方案》，IEEE《视频技术电路与系统汇刊》，2023年。[33] D. Yuan 等，《视觉 JND：视频编码中的一种感知测量》，IEEE Access，2019年。[34] P. Lebreton 和 K. Yamagishi，《预测自适应比特率视频流媒体中的用户退出率》，IEEE 多媒体汇刊，2020年。[35] P. T. Rajendran 等，《面向自适应视频流的能质感知可变帧率帕累托前沿》，载于 2024 IEEE 国际视觉通信与图像处理会议，2024年。[36] Y. Zhao 等，《面向逐镜头自适应编码的高效码率阶梯构建》，载于 IEEE 国际视觉通信与图像处理会议。IEEE，2024年。[37] A. Stergiou 和 R. Poppe，《Adapool：用于信息保持下采样的指数自适应池化》，IEEE 图像处理汇刊，2022年。[38] A. Wieckowski 等，《VVenC：一种开放且经过优化的 VVC编码器实现》，载于 IEEE 国际多媒体与博览会研讨会。IEEE，2021年。[39] A. Wieckowski 等，《面向即将推出的通用视频编码编解码器的实时软件解码器实现》，载于 IEEE 国际图像处理会议。IEEE，2020年。[40] V. V. Menon 等，《面向自适应视频流高效编码的绿色视频复杂度分析》，载于首届国际绿色多媒体系统研讨会论文集，2023年。[41] D. C. Montgomery 等，《线性回归分析导论》。John Wiley & Sons，2021年。[42] L. Breiman，《随机森林》，《机器学习》，2001年。[43] P. Geurts、D. Ernst 和 L. Wehenkel，《极端随机树》，《机器学习》，2006年。[44] D. P. Solomatine 和 D. L. Shrestha，《AdaBoost.RT：一种用于回归问题的提升算法》，载于 IEEE 国际联合神经网络会议。IEEE，2004年。[45] G. Ke、Q. Meng、T. Finley、T. Wang、W. Chen、W. Ma、Q. Ye 和 T.-Y. Liu，《LightGBM：一种高效的梯度提升决策树》，《神经信息处理系统进展》，2017年。[46] T. Chen 和 C. Guestrin，《XGBoost：一种可扩展的树提升系统》，载于第 22 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集，2016年。[47] M.-C. Popescu 等，《多层感知机与神经网络》，WSEAS 电路与系统汇刊，2009年。[48] “GridSearchCV”，检索日期：2026 年 2 月 6 日。[在线]。可访问：https://scikit-learn.org/stable/modules/generated/sklearn.model selection.GridSearchCV.html[49] F. Pedregosa 等，《Scikit-learn：Python 机器学习》，《机器学习研究杂志》，2011年。[50] T. Akiba 等，《Optuna：新一代超参数优化框架》，载于第 25 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集，2019年。[51] V. Nair 和 G. E. Hinton，《修正线性单元提升受限玻尔兹曼机的性能》，载于第 27 届国际机器学习会议论文集，2010年。[52] K. He 等，《深入研究修正线性单元：在 ImageNet分类中超越人类水平的性能》，载于 IEEE 国际计算机视觉会议论文集，2015年。[53] D. P. Kingma 和 J. Ba，《Adam：一种随机优化方法》，arXiv 预印本 arXiv:1412.6980，2014年。[54] N. Barman 等，《Bj {\o} ntegaard Delta (BD)：该指标的教程概述、演进、挑战与建议》，arXiv 预印本 arXiv:2401.04039，2024年。[55] 苹果公司，《Apple设备HLS创作规范》。[在线]。可访问：https://developer.apple.com/documentation/http live streaming/hls authoring specification for apple devices[56] V. V. Menon 等，《使用 XPSNR 进行通用视频编码的凸包估计》，载于 IEEE 国际图像处理会议。IEEE，2024年。[57] S. M. Lundberg 和 S.-I. Lee，《解释模型预测的统一方法》，《神经信息处理系统进展》，2017年。–[58] H. Zhang 等，《双侧敏感性分析：–更好地理解神经网络》，《国际机器学习与控制论杂志》，2022年。[59] M. Wisniewski 等，《传感器受限环境中导航任务的深度强化学习基准测试》，《智能与机器人系统杂志》，2025年。

, 2024年。